RITA Intelligent Transportation Systems Joint Program Office

Core System System Architecture Document (SAD)

www.its.dot.gov/index.htm
October 14, 2011

U.S. Department of Transportation
Research and Innovative Technology Administration

Produced by Lockheed Martin
ITS Joint Program Office
Research and Innovative Technology Administration
U.S. Department of Transportation

Notice

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD MM YYYY)	2. REPORT TYPE	3. DATES COVERED
14 10 2011	(Architecture Description)	N/A

4. TITLE AND SUBTITLE
Core System: System Architecture Document (SAD)

5a. CONTRACT NUMBER
GS-23F-0150S

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
N/A

5d. PROJECT NUMBER
DTFH61-10-F-00045

5e. TASK NUMBER
4

5f. WORK UNIT NUMBER
N/A

6. AUTHOR(S)
Core System Engineering Team

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Lockheed Martin
9500 Godwin Drive
Manassas, VA 20110

8. PERFORMING ORGANIZATION REPORT NUMBER
11-USDOTSE-LMDM-00044

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US Department of Transportation
Research and Innovative Technology Administration
ITS Joint Program Office
1200 New Jersey Ave., S.E.
Washington D.C. 20590

10. SPONSORING/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITOR'S REPORT NUMBER(S)

12a. DISTRIBUTION/AVAILABILITY STATEMENT
This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

12b. DISTRIBUTION CODE

13. SUPPLEMENTARY NOTES

System Architecture Document (SAD) for the Core System portion of the *connected vehicle* program

14. ABSTRACT (Maximum 200 words)
This document describes the System Architecture of the Core System for the United States Department of Transportation's (USDOT) next generation integrated transportation system. It describes the architecture of the system from five viewpoints: enterprise, functional, connectivity, communications, and information. It is consistent with the Core System Concept of Operations (ConOps) and Core System System Requirements Specification (SyRS).

15. SUBJECT TERMS
connected vehicle, core system, architecture, stakeholders, functional, enterprise, viewpoint, views, connectivity, communications, information, security, credentials, data distribution

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	None	305	Walt Fehr
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER: (202) 366-0278

CHANGE LOG

Revision	*Change Summary*	*Author*	*Date*
-	Initial Release	Lockheed Martin	6/13/2011
A	Walkthrough Draft	Lockheed Martin	7/13/2011
B	Draft	Lockheed Martin	9/06/2011
C	Incorporated feedback from requirements and architecture walkthroughs.	Lockheed Martin	10/14/2011

READER'S GUIDE

Those readers familiar with previous editions of this document should understand the changes that the document has undergone to make this version:

- In every case where an alternative was presented a selection was made and included in chapter 4. All alternatives not selected have been moved to chapter 6. Rationale for the rejection of alternatives accompanies those alternatives in chapter 6. Most significantly, the functionality of IEEE 1609.2 Certificate Authority has been moved outside the Core System.
- The organization of Functional Views has been changed. Misbehavior Management, System Configuration and User Configuration are now in their own views and not integrated within other views. The removal of IEEE 1609.2 Certificate Authority functions has affected many Functional Views.
- Enterprise relationships facilitating the development of business models are included in an Enterprise View – Business Model Facilitation.
- Additional Communications and Information Views have been added.
- Software Engineering Objects have been added to Connectivity Views.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table ***Title*** ***Page***

1.0 INTRODUCTION

1.1 Identification

This document is the System Architecture Document (SAD) for the Core System for the United States Department of Transportation's (USDOT) *connected vehicle* program.

1.2 Document Overview

The USDOT initiated this Systems Engineering (SE) project to define the Concept of Operations (ConOps), requirements, and architecture for the Core System that will enable safety, mobility, and environmental applications in an environment where vehicles and personal mobile devices interact wirelessly, hereafter referred to as the *connected vehicle* environment.

The intended purpose of this SAD is to describe the system architecture of the Core System by following Standard 1471-2000 of the Institute of Electrical and Electronics Engineers (IEEE), the IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.

The IEEE 1471-2000 standard suggests the use of views for documenting different aspects of a software intensive system, without recommending particular viewpoints. The standard does discuss several example viewpoints:

- Structural and Behavioral Viewpoints, which are used for software-intensive systems,
- Physical Interconnect and Link Bit Error Rate Viewpoints, which focus on communications and one particular performance metric (bit error rate),
- "Decomposition and Allocation" Viewpoint, which incorporates requirements development and traceability to components of the system architecture,
- Enterprise, Information, Computational, Engineering and Technology Viewpoints of the Reference Model of Open Distributed Processing (RM-ODP), defined by International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC) 10746-3: 1996. These viewpoints provide an architectural framework suited to distributed processing systems, in particular those "in which discrete components may be located in different places, or where communications between components may suffer delay or may fail."

Functionality within the *connected vehicle* environment is distributed between mobile and non-mobile entities and the Core, and interactions take place over a variety of wired and wireless links with varying performance characteristics subject to environmental degradation and failure. Consequently, the RM-ODP model is most appropriate to the Core System and how it fits within the *connected vehicle* environment. The Viewpoints defined in RM-ODP include concepts from the other sets of Viewpoints but address a greater variety of concerns while maintaining a high degree of coordination between Viewpoints.

The Consultative Committee for Space Data Systems (CCSDS) extended the RM-ODP framework and refined the five viewpoints in the Reference Architecture for Space Data Systems (RASDS). These Viewpoints include most concepts of the RM-ODP Viewpoints but are better suited to describing layered communications interfaces.

Some of the elements that the Core System interacts with are constantly in motion and may only occasionally be in contact with other elements and the Core. The *connected vehicle* environment is dependent on wireless communications that requires functionality at all communications layers to address security concerns within an intermittently connected system. The RASDS identifies Viewpoints that are

more applicable to address these dependencies and concerns. Consequently this architecture description uses the RASDS framework.

Within RASDS, the Enterprise and Information Viewpoints are similar to viewpoints of the same name in RM-ODP. RM-ODP's Computational viewpoint is replaced with the Functional Viewpoint, which addresses the same concerns but uses slightly different language. The Engineering and Technology viewpoints are replaced with the Connectivity and Communications Viewpoints. The replacements include most of the information from the RM-ODP counterparts but have a greater focus on the disconnected nature of interacting elements, which makes these views preferable for the Core System.

The System Architecture document consists of the following sections:

- Section 1.0 provides an introduction to this document.
- Section 2.0 identifies the stakeholders and their concerns that were considered when developing the system architecture.
- Section 3.0 introduces the concepts of architecture Viewpoints and Views, and further details the Viewpoints that have been selected to address stakeholder concerns. The Viewpoints are based on those defined in RM-ODP, but have been customized to define the Core System.
- Section 4.0 presents detailed architecture views for each of the viewpoints introduced in Section 3.0.
- Section 5.0 describes the known inconsistencies among the architecture views as well as a discussion of the consistencies across all views.
- Section 6.0 describes trade-offs considered, alternatives not chosen, and other analyses that led to choosing the architecture described in this System Architecture Document.
- Section 7.0 contains the appendix which has a table summarizing the contents of the architecture viewpoints as well as supporting analysis used during the development of the Views in section 4.
- Section 8.0 contains the glossary and acronym list

1.3 Scope

This document presents the System Architecture of the Core System. The Core provides two basic services to participants in the *connected vehicle* environment:

1. It facilitates the trusted and secure exchange of information between participants.
2. It provides mechanisms for participants to distribute information to each other without entering into relationships.

The needs defined in the ConOps and the requirements in the System Requirements Specification (SyRS) define exactly what must be done to provide these two services. This document describes *how* it is to be done.

In systems engineering parlance, this SAD provides the high level design. It does not specify particular technologies unless requirements demand or constraints restrict the system to a particular technology. The next phase in the systems engineering process, low level design, thus has the freedom to choose particular technologies for implementation.

This document describes the Core in five ways:

1. It describes the context of the *connected vehicle* environment in which the Core operates by defining the enterprise interactions that affect the Core System's ability to deliver services.

2. It defines the functions and interactions that are necessary to implement the Core System.
3. It defines the physical implementation of the Core in terms of processing nodes and communications links.
4. It defines the communications environment that *connected vehicle* participants use to interact with the Core.
5. It defines the information that the Core requires of *connected vehicle* participants, and the information the Core provides to them.

Together, these definitions, or Viewpoints, satisfy the requirements specified in the SyRS. Traceability between the SyRS and the SAD is included in the SyRS.

1.4 Context

Figure 1-1 shows the context in which the Core System resides as documented in the ConOps.

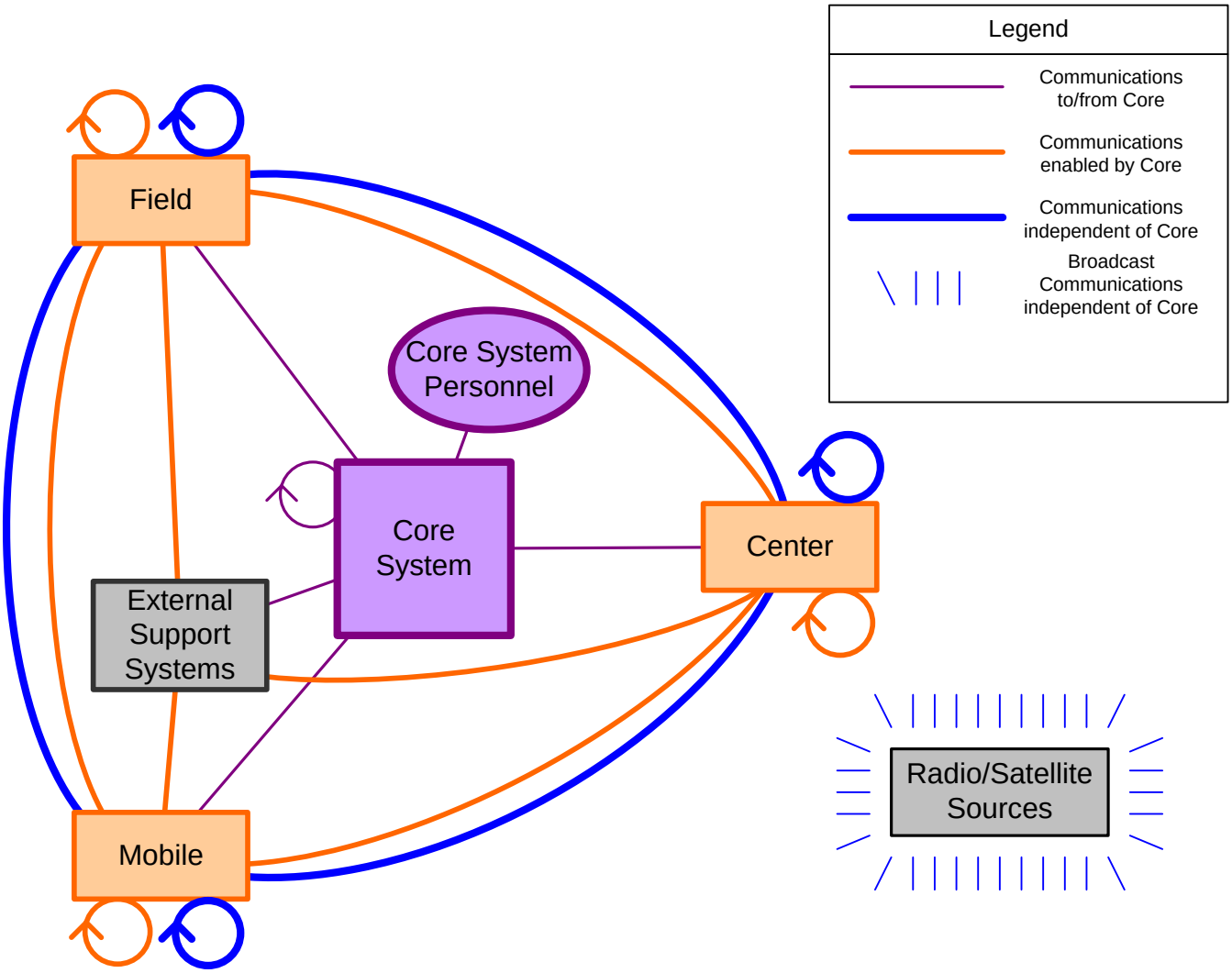


Figure 1-1: Core System Context Diagram

The Core System interacts with four types of entities:

- **Mobile** includes all vehicle types (private/personal, trucks, transit, emergency, commercial, maintenance, and construction vehicles) as well as non-vehicle-based platforms including portable personal devices (smart phones, tablets, etc.) used by travelers (vehicle operators, passengers, cyclists, pedestrians, etc.) to provide and receive transportation information. Mobile entities interact with other Mobile and Field entities (e.g. Dynamic Message Signs (DMS), Roadside Equipment (RSE)) in the Mobile entity's vicinity, and Center entities from any location.
- **Field** represents the intelligent infrastructure distributed near or along the transportation network which perform surveillance (e.g. traffic detectors, cameras), traffic control (e.g. signal controllers), information provision (e.g. DMS) and local transaction (e.g., tolling, parking) functions. Typically, their operation is governed by transportation management functions running in back offices. Field entities also include RSE supporting Dedicated Short Range Communications (DSRC), and other non-DSRC wireless communications infrastructure that provides communications between Mobile entities and fixed infrastructure.
- **Center** represents back office systems including public and commercial transportation and non-transportation systems that provide management, administrative, information dissemination, and support functions. These systems may exchange information relevant to the *connected vehicle* environment with other center systems. All of these systems take advantage of the Core System to provide or also make use of application data.
- **Core System Personnel** represents the people that operate and maintain the Core System. In addition to network managers and operations personnel, Core System Personnel also includes those that are deploying and provisioning Core elements. Other personnel interacting with the Core include developers of software services that, maintain, fix and expand Core services or extend the system as required.

The **Core System** also interacts with other instantiations of Core Systems. More than one Core may exist, each providing services over given geographic or topical areas. Some may provide backup or standby services for others; some may provide more or less services than others.

Radio/Satellite Sources refer to terrestrial radio and satellite broadcast, including Global Positioning System (GPS) broadcasts, and position correction broadcasts.

External Support Systems (ESS) provide services on behalf of and/or in support of the Core System. These services are provided by the ESS because it makes more sense to manage, maintain and share the service between multiple Cores due to overriding institutional, performance or functional constraints.

When referring to multiple Core Systems, this document will often drop the word System. In most cases Core System is used for the first reference in a paragraph, and Core used thereafter in that paragraph, though in those sections describing interactions between Cores or talking about Core subsystems the word System is usually left off entirely. This is strictly for readability. The title Core is a shortened version of Core System.

Throughout the remainder of this document, the term **System Users** refers to the combination of **Mobile, Field,** and **Center**. The term **End User** refers to the human user of the System User device. End Users do not interact directly with the Core System, but are referred to as the ultimate beneficiaries or participants in the *connected vehicle* environment.

1.5 References

Documents listed below were used in the preparation of this SAD.

- Core System Concept of Operations, 7 October 2011
- Core System System Requirements Specification (SyRS), 14 October 2011
- Crash Avoidance Metrics Partnership (CAMP) Security Final Report, Draft, 8/31/2010
- CCSDS 311.0-M-1 – Reference Architecture for Space Data Systems, Recommended Practice, September 2008
- IEEE Std. 1471 – IEEE Recommended Practice for Architectural Description of Software Intensive Systems, 21 September 2000
- IEEE 1609.2 Standard for Wireless Access in Vehicular Environments (WAVE) - Security Services for Applications and Management Messages, Jun 2006
- IEEE 1609.3 Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services, Apr 2007
- Object Management Group (OMG) Systems Modeling Language (OMG SysML™), version 1.1, November 2008
- Society of Automotive Engineers (SAE) J2735 - Dedicated Short Range Communications (DSRC) Message Set Dictionary, v2, Nov 2009
- Telecommunications Industry Association (TIA) Standard 942 – Telecommunications Infrastructure Standard for Data Centers, April 2005
- Vehicle Infrastructure Integration (VII) Privacy Policies Framework, Version 1.0.2, February 16, 2007

2.0 STAKEHOLDERS AND CONCERNS

A system stakeholder is an individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system. Concerns are those interests which pertain to the system's development, its operation or any other aspects that are critical or otherwise important to one or more stakeholders (see IEEE 1741-2000). All needs and goals and most constraints identified in the ConOps are reflected by one or more concerns, as documented in the description of those concerns.

2.1 Stakeholders

In the following sections the different stakeholder roles relevant to the Core System are described. IEEE 1471-2000 requires that at least the following stakeholders be considered: Users, Acquirers, Developers, and Maintainers.

The Concept of Operations defines three types of System Users: Mobile, Field and Center. These are applications that interact with the Core System. Each of these devices or systems is associated with one or more stakeholders, including those that create, maintain and use them. The ConOps also defines a fourth group that operates the Core: Operators.

2.1.1 Users

The User stakeholder is the operator of the Mobile, Field, or Center device or system. Owing to the significant logistical differences between these three types of devices and systems, their associated stakeholders are identified individually.

Operators are a fourth user type—not a System User, but a user nonetheless in that Operators are people that interact with the Core.

2.1.1.1 Mobile User

The Mobile User operates a personal information device (e.g. smart phone) or operates or rides in a vehicle that uses Core Services. Mobile Users include automobile, bus, truck, construction and emergency vehicle drivers and passengers, as well as cyclists and pedestrians.

2.1.1.2 Field User

The Field User owns, operates and/or maintains the intelligent infrastructure distributed near or along the transportation network which performs surveillance, information provision, local transactions, and control functions.

2.1.1.3 Center User

The Center User owns, operates, and/or maintains back office systems that use Core Services. Center User stakeholders include public and commercial transportation, transit and fleet managers, vehicle and device manufacturers, information service providers, emissions, commercial vehicle and other regulatory managers, toll administrators, maintenance and construction operators, public safety, insurance and transportation data aggregators.

2.1.1.4 Operator

The Operator is the day-to-day administrator of Core System services. The Operator is the user with authority to control delivery of services and manage interactions between Cores. The Operator has responsibility for system configuration and any manual processes that are part of day-to-day Core operations.

The Operator role will usually be funded by the same entity that serves as the Manager. The Operator may be part of the same agency as the Manager, or it may be contracted to a separate entity under control of the Manager.

2.1.2 Acquirer/Deployer

The acquirer is the entity that procures a Core System.

Examples of public entities that may assume the Acquirer role include state Departments of Transportation (DOTs), transit agencies, commercial vehicle administration agencies, or regional coordination entities that include elements of multiple such agencies. Examples of private entities that may assume the Acquirer role include commercial fleet management, vehicle manufacturer or information service provider (e.g., traveler information) entities.

2.1.3 Maintainer/Administrator

The Maintainer is the role that ensures continuous system operation by ensuring the availability of resources including power, environmental control, hardware, software patches and upgrades required for the Core System to operate properly.

The Maintainer role will usually be funded by the same entity that serves as the Manager. The Maintainer may be part of the same agency as the Manager, or it may be contracted to a separate entity under control of the Manager.

2.1.4 Developer

The Developer is responsible for transforming the system architecture into a functioning system that meets the requirements set forth in the System Requirements Specification.

The Developer role will likely be assumed by a combination commercial software development and hosting companies and standards bodies.

2.1.5 Manager

The Manager is responsible for planning the deployment and managing the operations of the Core System. The Manager does not interact directly with the Core. Changes to the Core that require operational modification (to enforce policies imposed by the Policy Setter for example) are determined by the Manager, but depending on scope and required functionality are implemented by the Developer, Maintainer or Operator.

The Manager role will be part of an agency or it may be contracted to a separate entity under control of an agency.

2.1.6 Tester

The Tester is responsible for verifying that changes made to Core System functionality are properly implemented. This includes both verification and validation steps.

2.1.7 Policy-Setter

The Policy-Setter determines policies that affect Core System deployment, implementation or operations. This could be a local entity such as a DoT, a regional body or a federal entity such as the Federal Communications Commission (FCC).

2.1.8 Application Developer

The Application Developer creates applications that will leverage the Core System's services on behalf of the users of those applications.

2.1.9 Device Developer

The Device Developer creates devices and systems (Mobile, Field and Center) that will interact with the Core System.

2.1.10 Service Provider

The Service Provider provides a service that is used by the Core System or by System Users to access the Core. Examples of Service Providers are RSE owner/operators, cellular network providers, and security providers such as the operator of a certificate authority.

2.2 Concerns of Stakeholders

Concerns are concepts that are of interest or importance to one or more stakeholders. Concerns may apply to any phase of the system life cycle. However, some concerns may apply more during system design, implementation, operations, or maintenance and not during other phases of the system's life. IEEE 1471 requires the purpose or mission to be included as a concern. This is documented in the Concept of Operations. The needs from the ConOps are derived from this mission, and are identified as concerns under the Appropriateness concern below.

2.2.1 Performance

This concern deals with characteristics such as speed (responsiveness of Core System services), availability, reliability, capacity, and other quantitative measures.

2.2.2 Interfaces

This concern deals with how interfaces are defined and how users access and operate with the Core System.

2.2.3 Functionality

This concern deals with how the Core System functions internally; how its components work together and how they transition between operational modes.

2.2.4 Security

This concern deals with the security of Core System services, maintaining sole control (both physical and non-physical) of the Core and ensuring the system's integrity. It also deals with maintaining the privacy and integrity of information passing to and through the Core.

2.2.5 Organization/Resources

This concern is about organization of system development and the resources required to develop, deploy, operate and maintain the system.

2.2.6 Appropriateness

This concern asks whether the delivered Core System fulfills the needs set forth in the Concept of Operations and meets the overall goals defined therein.

2.2.7 Feasibility

This concern deals with feasibility of deploying the Core System with regard to the current state of technology and available resources.

2.2.8 Risks

This concern deals with the risks of system development and operation. This concern clarifies which level of risk is acceptable. There may be various risks concerning the timely, cost effective deployment of a correctly functioning Core System.

2.2.9 Evolvability/Flexibility

This concern deals with the ability to extend and expand the system post-deployment to deal with new and potential unforeseen conditions or changes in its mission or scope.

2.2.10 Deployability

This concern deals with the issues surrounding deployment, including capital, human and other resource requirements, business models as well as transitions between development, deployment and operations.

2.2.11 Maintainability

This concern addresses issues of maintenance, in terms of resources required and impact on operations.

2.3 Stakeholder / Concern Matrix

Table 2-1 summarizes the relations between stakeholders and concerns as a matrix, where rows correspond to concerns and columns to stakeholder roles. A shaded cell indicates that the concern is one that is relevant to the stakeholder's role in the Core System's lifecycle.

Table 2-1: Stakeholder/Concern Matrix

Concern \ Stakeholder	Mobile User	Field User	Center User	Operator	Acquirer	Maintainer	Developer	Manager	Tester	Policy Setter	Application Developer	Device Developer	Service Provider
Performance	■	■	■	■	■	■	■		■		■	■	■
Interfaces		■	■	■		■	■		■		■	■	■
Functionality	■	■	■	■	■	■	■		■		■	■	
Security	■	■	■	■	■	■	■	■	■	■	■	■	■
Organization/Resources				■	■	■		■		■			■
Appropriateness	■	■	■	■	■	■			■	■			
Feasibility					■		■	■			■	■	■
Risks	■	■	■	■	■	■	■	■	■	■	■	■	■
Evolvability					■		■	■			■	■	
Deployability					■			■		■	■	■	■
Maintainability				■	■	■	■	■	■	■	■		■

Across the entire Core System there are the following relationships between Stakeholders and Concerns, as shown in the Table above:

- Mobile Users – concerns include Performance, Functionality, Security, Appropriateness, and Risk
- Field Users – concerns include Performance, Interfaces, Functionality, Security, Appropriateness, and Risk
- Center Users – concerns include Performance, Interfaces, Functionality, Security, Appropriateness, and Risk

- Operators – concerns include Performance, Interfaces, Functionality, Security, Organization/Resources, Appropriateness, and Risk
- Acquirers – concerns include Performance, Functionality, Security, Organization/Resources, Appropriateness, Feasibility, Risk, Evolvability, Deployability, and Maintainability
- Maintainers – concerns include Performance, Interfaces, Functionality, Security, Organization/Resources, Appropriateness, Risk, and Maintainability
- Developers – concerns include Performance, Interfaces, Functionality, Security, Feasibility, Risk, Evolvability, and Maintainability
- Managers – concerns include Security, Organization/Resources, Feasibility, Risk, Evolvability, Deployability, and Maintainability
- Testers – concerns include Performance, Interfaces, Functionality, Security, Appropriateness, Risk, and Maintainability
- Policy Setters – concerns include Security, Organization/Resources, Appropriateness, Risk, Deployability, and Maintainability
- Application Developers – concerns include Performance, Interfaces, Functionality, Security, Feasibility, Risk, Evolvability, Deployability, and Maintainability
- Device Developers – concerns include Performance, Interfaces, Functionality, Security, Feasibility, Risk, Evolvability, and Deployability
- Service Providers – concerns include Performance, Interfaces, Security, Organization/Resources, Feasibility, Risk, Deployability, and Maintainability

3.0 ARCHITECTURAL VIEWPOINTS

The architecture of the Core System is described from multiple Viewpoints, each focusing on different concerns associated with the system.

A Viewpoint Specification is a specification of the rules and structure required to focus on particular concerns within a system. Each Viewpoint Specification includes a template from which to develop individual Views. A View is a representation of the system from the perspective of a related set of concerns.

Each Viewpoint Specification is intended to describe the system from a different perspective than the other viewpoints, but some specific areas of overlap exist to allow the elements in different Viewpoints to be related. Each Viewpoint exposes a different set of design concerns and issues, and each provides the means for reasoning about that aspect of the system.

Each Viewpoint Specification describes the Core System as a set of Objects and interactions among them. An Object is an abstract model of an entity in the system. Objects have behaviors and states and are distinct from any other object. An object is characterized by whatever attributes distinguish it from other

Encapsulation is the property that the information contained in an object is accessible only through interactions at the interfaces supported by the object. Because objects are encapsulated, there are no hidden side effects of interactions. That is, an interaction with one object cannot affect the state of another object without some secondary interaction taking place with that object. Thus, any change in the state of an object can only occur as a result of an external request of an object, an internal action of the object, or an interaction of the object with its environment. For example, the Modify C2C Operational State function in Functional View – Credentials Distribution controls the operational state of Core2Core subsystem objects. It does nothing else.

Abstraction is the selective examination of certain aspects of a problem. The goal of abstraction is to isolate those aspects that are important for some purpose and suppress those aspects that are unimportant. For example, the Core Certification Authority defined in Enterprise View – Governance is concerned with certification of devices and systems in the *connected vehicle* environment. It is not concerned with anything else.

The **behavior** of an abstract data object is fully defined by a set of abstract operations defined on the object; the user of an object does not need to understand how these operations are implemented or how the object is represented. For example, the Local Encryptor defined in Connectivity View – Core System Functional Allocation accepts messages and provides encrypted messages. Those are its only behaviors.

Consider the analogy of a house to the Core System. A house has organized collections of structural, electrical, plumbing, and mechanical components that work together to allow the house to function. Each Viewpoint looks at exactly one of these sets of components. In other words, there is a structural viewpoint, electrical viewpoint, and so forth.

Imagine that one could see one type of system by wearing the appropriate pair of glasses; e.g. electrical glasses, mechanical glasses. Looking in a window of the house with electrical glasses shows one View of the Electrical Viewpoint. In order to understand the entire house's electrical systems, one must look in every window with the electrical glasses. In order to understand the entire house, one must look in every window, and with every set of glasses. The electrician might only care about the Electrical Viewpoint, and the plumber the Plumbing Viewpoint.

Similarly, in order to understand the entirety of the Core System Architecture, one must understand every View of the Enterprise, Functional, Connectivity, Communications and Information Viewpoints. Different stakeholders may care about only certain Viewpoints, depending on how they interact with the Core.

objects and by encapsulation, abstraction, and behavior. An example of an Enterprise Object is an abstract model of an organization. Some attributes of this object include resources, policies etc. Objects defined in their primary Viewpoint often have corresponding objects in other Viewpoints.

A Viewpoint Specification defines the rules for constructing Views of the system.

A View is a representation of the Core System or a portion of the Core that address a set of concerns about the Core. Architecture views are representations of the overall architecture that are meaningful to one or more stakeholders in the system. They enable the architecture to be communicated to and understood by the stakeholders, so they can verify that the system will address their concerns. Views are themselves modular and well-formed; each View is intended to correspond to exactly one Viewpoint and is constructed using the rules defined by that Viewpoint Specification.

The conceptual model of architectural description is captured in the figure below. In the figure, boxes represent classes of things. Lines connecting boxes represent associations between things. An association has two roles (one in each direction). A role can optionally be named with a label. The role from A to B is closest to B, and vice versa (e.g., A System has an Architecture). A role can have a multiplicity, e.g., a role marked with "1.*" is used to denote *many*, as in a one-to-many or many-to-many association. A diamond (at the end of an association line) denotes a *part-of* relationship. The relationships illustrated in Figure 3-1 can be read as:

- A System has an Architecture
- A System has one or more Stakeholders
- An Architecture is described by one Architectural Description
- A Stakeholder has one or more Concerns
- An Architectural Description identifies one or more Stakeholders
- An Architectural Description selects one or more Viewpoints
- An Architectural Description is organized by one or more Views
- A Concern is important to one or more Stakeholders
- A Viewpoint is addressed to one or more Stakeholders
- A Viewpoint is used to cover one or more Concerns
- A View conforms to a Viewpoint
- A View is part of an Architectural Description

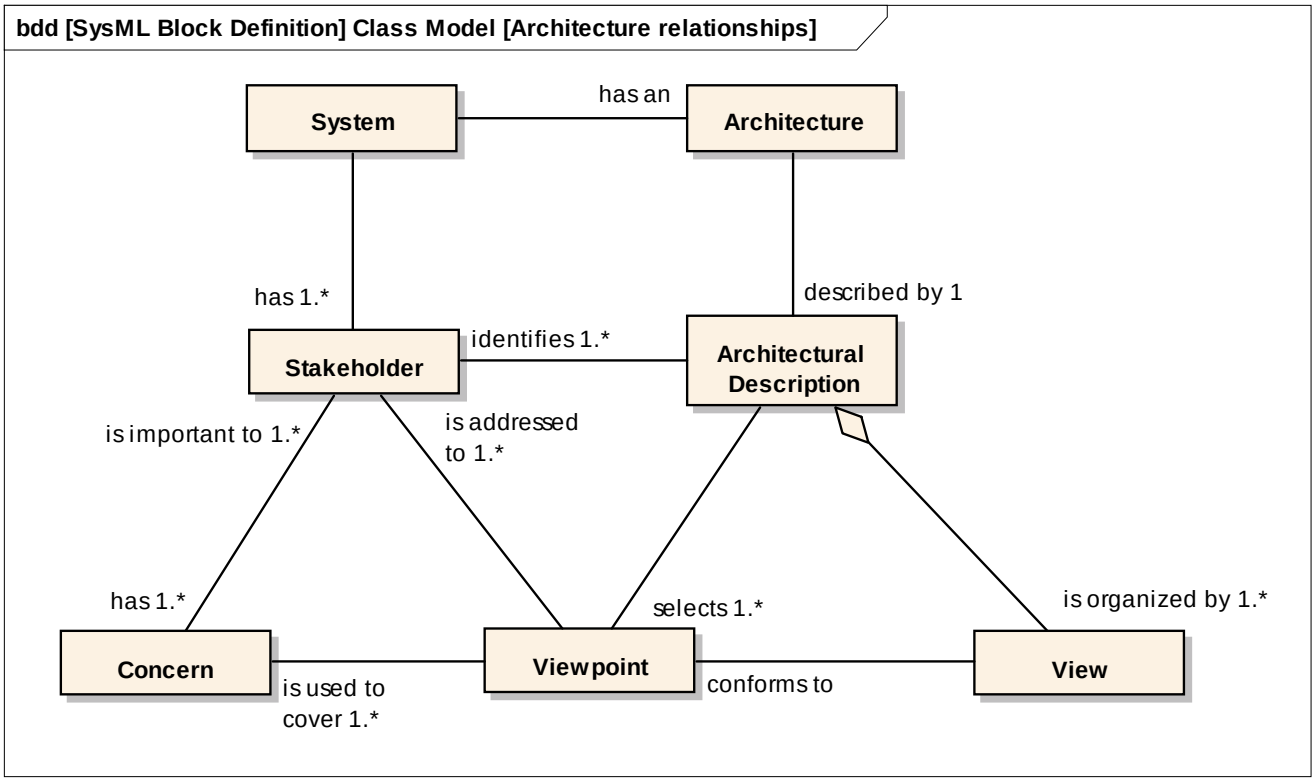


Figure 3-1: Conceptual Model of Architectural Description

Another way to picture the relationship between the Architecture, Viewpoints and Views is shown below. This figure identifies the five Viewpoints used to describe the Core System: Enterprise, Functional, Connectivity, Communications, and Information. There are multiple Views per Viewpoint. Only by considering all Views of each Viewpoint can one gain a complete picture the architecture.

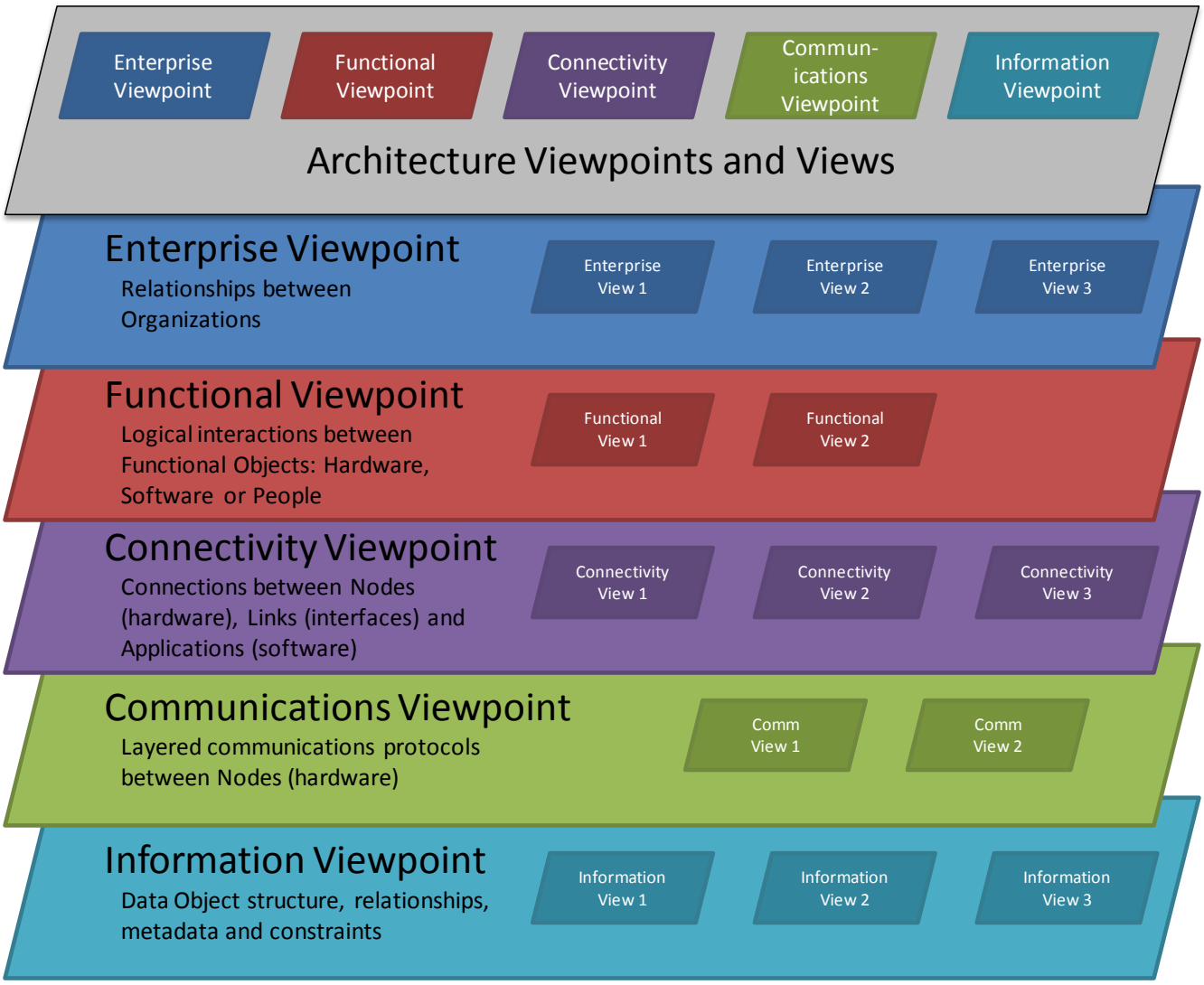


Figure 3-2: Core System Architecture Viewpoints

The ISO Open Systems Interconnection (OSI) Model, an industry standard communications protocol model has parallels to some of these viewpoints. The seven layers of the OSI Model are more rigid, in that each layer of the OSI Model provides services to the layers above and receives services from the layers below. While Viewpoints are not so aligned from top to bottom, Viewpoints do include objects that mirror the relationships between OSI Model layers. Viewpoints tend to include objects that could be assigned to particular layers of the OSI Model:

- The Communications Viewpoint includes the OSI model layers 1-5 (Physical, Data Link, Network, Transport and Session)
- The Information Viewpoint covers layer 6 (Presentation)
- The Functional and Connectivity Viewpoints include functional components that align with Layer 7 (Application)

The architecture of the Core System is described by five Viewpoint Specifications - one per Viewpoint - which are explained in the following subsections. Each Viewpoint is documented as follows, where the letter *i* stands for the number of the viewpoint (1, 2, etc.).

Table 3-1: Viewpoint Description Template

Section #	Section Name	Description
3.i	**Name of Viewpoint**	Name of the Viewpoint (e.g., Enterprise, Functional).
3.i.1	**Overview**	A brief overview of the viewpoint.
3.i.2	**Definition of Terms**	Defines terms used by the Viewpoint.
3.i.3	**Stakeholders Addressed**	Lists the stakeholders from Section 2.1 whose concerns are addressed by this Viewpoint.
3.i.4	**Concerns Addressed**	Describes the concerns from Section 2.1.8 that this viewpoint is intended to address. Includes lists of questions that can be answered by consulting views that conform to this viewpoint.
3.i.5	**Objects and Relationships**	Identifies and describes the types of objects (e.g. Enterprise Object) and their respective attributes that appear in the Viewpoint, and the rules for object interaction.
3.i.6	**Method(s) to Model/Represent Conforming Views**	Lists the methods and techniques used to model or represent views conforming to this Viewpoint (e.g., Systems Modeling Language, or a newly defined set of graphical constructs). Includes a legend for objects that may appear in Views. Not all methods are mandatory; in some cases Viewpoints may define methods that are possible for as-yet undefined views, or that were used as part of architectural analysis and have since been rejected.
3.i.7	**Viewpoint Source**	Provides a citation for the source of this Viewpoint definition, if any.
3.i.8	**Security Issues**	Describes the security issues that this Viewpoint is intended to address. These are called out separately because security concerns must be addressed by every Viewpoint, and because a failure in security from one perspective is effectively a failure overall.
3.i.9	**Views Modeled**	Describes the Views that are modeled using the Viewpoint specification.
3.i.10	**Example View**	Illustrates an example View and describes the information that can be obtained from the view image.

3.1 Global Definitions

These terms apply to more than one Viewpoint (these definitions based on those in CCSDS 311.0-M-1):

An **element** is (1) a constituent part of something; (2) any thing that is one of the individual parts of which a composite entity is made up; (3) an identifiable component, process or entity of a system.

An **entity** is any physical or abstract thing of interest. For example, an entity may be a computer, an organization, a piece of software or a set of functions performed by a system.

A **policy** is a set of guidelines and constraints on the behaviors and states exhibited by the objects in the Core System.

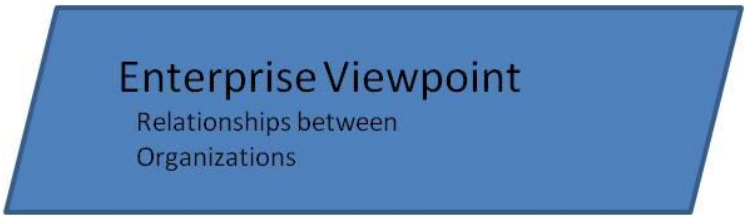
A **role** describes the way in which an object participates in a relationship; an object's set of behaviors and actions associated with the relationship of that object with other objects.

An **object** is an abstract model of an entity in the real world. It contains information, has behavior and may offer services. An object is characterized by that which makes it distinct from other Objects. Specific types of objects are defined in specific viewpoints: e.g. Enterprise Object is defined in the Enterprise Viewpoint.

A **service** is a provision of an interface of an object to support actions of another object.

A **standard** is a formal specification that defines and governs functions and protocols at interfaces of a data system. It describes in detail the capabilities and establishes the requirements to be met by interfacing systems to achieve compatibility.

3.2 Enterprise Viewpoint



3.2.1 Overview

The Enterprise Viewpoint addresses the relationships between organizations and the roles those organizations play that involve various resources, including digital certificates, roadside equipment, Core System infrastructure, and communications network equipment. It also addresses the personnel (including operators, users and support staff) that are part of those organizations.

In the Enterprise Viewpoint, the Core System is depicted as a set of Enterprise Objects that interact with Enterprise Objects outside the Core. It focuses on the relationships between Core and external Enterprise Objects, but may include interactions between external Enterprise Objects if an alternative view shows that relationship having an impact on the Core. Enterprise Objects representing system elements that have significant resources may appear in an Enterprise View as Facilities.

The relationships between Enterprise Objects are largely determined by roles, responsibilities, policies and goals of the Enterprises, not by Core System policies or goals. The relationships between Enterprise Objects will therefore depend on the responsibilities of the Enterprises involved and the roles they choose to play in implementing a Core.

3.2.2 Definition of Terms

An **Enterprise Object** represents an entity that is governed by a single authority that has its own objectives and policies for operating the object. An Enterprise Object may be a component of another larger Enterprise Object, which may in turn be a component of a third, even larger, Enterprise Object (e.g., a

Traffic Management Center is a component of State DOT is a component of State Government). Enterprise Objects may participate wholly or in part in other Enterprise Objects (e.g., a Device Developer is a component of Auto Manufacturer but also participates in Standards Body). External Support Systems (ESS) are Enterprise Objects.

A **Facility** is a physical infrastructure element that supports the use of services and other resources.

A **Resource** in general terms is anything available to a system that can support the achievement of objectives; i.e., any physical or virtual element that may be of limited availability within a system. In this context a Resource is an Enterprise Object that has some role, offers services, and performs some action within a system. A resource may serve more than one activity.

A **domain** is an Enterprise Object that is under single organizational, administrative or technical control.

A **federation** is a group of domains that coordinate to share resources while each domain retains its authority over its own resources. Federations are governed by negotiated agreements.

An organizational Enterprise Object may own a facility or resource Enterprise Object. **Ownership** means having administrative and fiscal responsibility for the owned element and the right to exclusively control and use it for one's own purposes. It is the state or fact of having exclusive possession or control of some object, facility, intellectual property or some other kind of property. Not every organizational object owns facilities or resources. Some resources are owned by one organization and used by others. The term **cross support** is used to describe an agreement between two or more organizations to exploit the technical capability of interoperability for mutual advantage, such as one organization offering support services to another in order to enhance or enable some aspect of a mission.

3.2.3 Stakeholders Addressed

Mobile User, Field User, Center User, Operator, Acquirer, Maintainer, Developer, Manager, Policy-Setter, Application Developer, Device Developer and Service Provider.

3.2.4 Concerns

Security	What entities are involved in the distribution of digital certificates, and what roles do those entities have? What entities are involved in the detection of misbehavior by System Users, and what roles do those entities have?
Organization/Resources	Who needs to contribute resources to Core System development, testing, transition, and operations? What Core System resources are required to support external application development? How do the entities responsible for a Core System need to interact with entities responsible for other Core Systems?

Risks	What relationships between Core Enterprise Objects and external Enterprise Objects provide risks to Core System development, deployment, operations, and maintenance? What steps can be taken to lessen risks that are a function of Enterprise relationships?
Evolvability	How do the relationships between Enterprise Objects need to change to support the integration of new Enterprises? Specifically, what is the decision mechanism for integrating new Enterprises and modifying roles of existing Enterprises?
Deployability	Who needs to be involved with the transition from development to operations? What resources are required to support the transition from development to operations?
Maintainability	Who needs to contribute resources to Core System maintenance activities? What resources are required to support Core System maintenance? What interactions between Enterprise Objects are required to support maintenance activities while maintaining Core operations?

3.2.5 Objects and Relationships

The following elements may appear in the Enterprise Viewpoint:

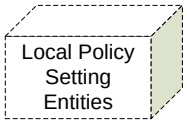
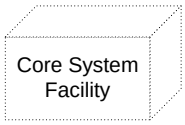
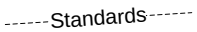
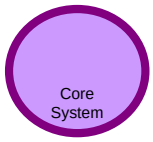
- Enterprise Objects:
  - Types: Organizations and Resources
  - Attributes: Name, type, role, objectives, location (relative or absolute), members, resources, interaction modes, requirements and constraints
  - Inputs: Requirements, agreements, contracts, funding, policies, rules, purpose
  - Outputs: Requirements, policies, rules, agreements
- Domains: boundaries of responsibility or ownership
- Information: instances of documents, agreements, contracts, policies, requirements, objectives, goals, interface specifications. Formal definition of data to be exchanged may be found in the Information Viewpoint

Relationships between Enterprise Objects: ownership, membership, participation, contractual, exchange of information, agreement

3.2.6 Method(s) to Model/Represent Conforming Views

Enterprise View diagrams use the graphical objects defined in Table 3-2.

Table 3-2: Enterprise View Graphical Object Definitions

	Enterprise Objects are represented by 3-dimensional shaded boxes drawn with dashed lines, with the name of the Enterprise on the front face. Objects are named with regard to the role they play in the View. An Enterprise Object may appear multiple times in a drawing, but only to improve drawing clarity. Duplicates will be shaded.
	Facilities are represented by 3-dimensional shaded boxes drawn with dotted lines, with the name of the Facility on the front face. Objects are named with regard to the role they play in the View. A Facility may appear multiple times in a drawing, but only to improve drawing clarity. Duplicates will be shaded.
	Logical relationships between Enterprise Objects are represented with dashed lines and indicate the exchange of information. The example here indicates that standards documents are being exchanged.
	Domains are modeled with colored circles.

Each Enterprise Object is defined textually. The relationships between Enterprise Objects are defined in the text describing those objects. The type of relationship and/or information exchanged between Objects is depicted in the relationship line on the diagram. Domains are drawn to graphically include the Enterprise Objects that are included within them.

3.2.7 Viewpoint Source

This Viewpoint is based on the Enterprise Viewpoint documented in CCSDS 311.0-M-1.

3.2.8 Security Issues

In the Enterprise Viewpoint the security issues which must be addressed include organizational roles, policies, trust relationships, domain boundaries and cross-support security agreements. The implementation mechanisms to enforce these rules and agreements are detailed in other viewpoints.

Organizational roles define the behaviors and actions that an entity may undertake in its relationships with other entities. This implies some limits on the entity's behavior; if behaviors are not followed as the architecture describes, a security or functional problem may result.

Similarly, policies established by one entity may constrain or place requirements on the roles of other entities. Enforcement of adherence to these policies and thus fulfilling roles requires system or even multi-system level decisions. Is policy enforcement going to be undertaken by an over-arching entity, or will entities simply dissolve relationships if the relevant partner does not fulfill his role? In either case, if the role in question is one that is critical to system operation, who will step in to fill the void? The answers to these questions impact security because if an entity fails to fulfill his promised duty with respect to the Core System, the maintenance of trust between System Users may fail. These are questions that cannot be answered by the architecture, but they can be illustrated.

Trust relationships are at the heart of the Core System; the Core's main job is facilitating trust between System Users. Trust between entities that implement, operate and develop Cores is a different thing entirely. The trust that the Core ensures is between devices for which protocols can be defined that provide a high confidence in the trustworthiness of communicating partners. Trust between entities must be established by humans, established by agreement and possibly documented in a contractual relationship. Such trust relationships can be addressed and illustrated in the architecture, but must be implemented by leaders and validated by the stakeholders of the respective entities involved.

Domain boundaries and cross-support security agreements follow in the wake of trust relationships, roles and policies governing operational scope and security. A federation of Core Systems, where a Core is a singular domain, may be established by establishing trust between Cores, agreeing on and implementing policies that are consistent between Cores, including the conventions for defining and modifying Core boundaries. This federated environment will enable security agreements between Cores, and ensure a consistent approach to the handling of security issues such as certificate distribution and revocation across all Cores.

Details concerning these security issues are addressed individually in the text associated with each view.

3.2.9 Views Modeled

The following Enterprise Views are modeled:

1. Enterprise View – Security Credentials Distribution: addresses the distribution of digital certificates required to manage security of System Users.
2. Enterprise View – Operations: addresses the relationships between Core Systems and between Cores and System Users necessary to implement the Core's services.
3. Enterprise View – Core System and Application Development and Deployment: addresses the development of Core System services and applications.
4. Enterprise View – Configuration and Maintenance: addresses the maintenance of the Core System.
5. Enterprise View – Governance: addresses management and leadership decision-making related to defining Core System use and management.
6. Enterprise View – Business Model Facilitation: addresses relationships relevant to data exchange from data provider to data subscriber using the Core System.

3.2.10 Example View

Figure 3-3 illustrates a sample Enterprise View. There are five Enterprise Objects, six relationships and one domain shown in this view. Read an Enterprise View by understanding relationships. Most can be read in the form <object 1> <action verb> <relationship> with <object 2>. For example, *Funding Agency* has a *load and deployment agreement* with *System Acquirer*. The domain is an Enterprise Object that includes other Enterprise Objects under the same organizational, technical or administrative control, e.g., *System Acquirer* is part of the *System* domain.

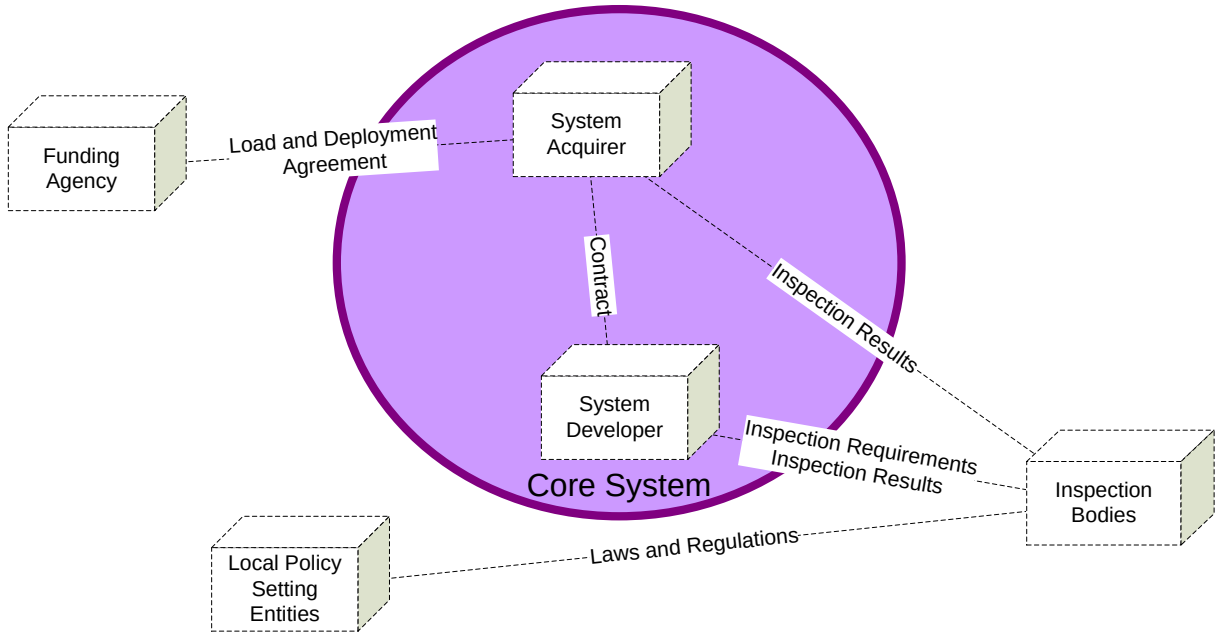


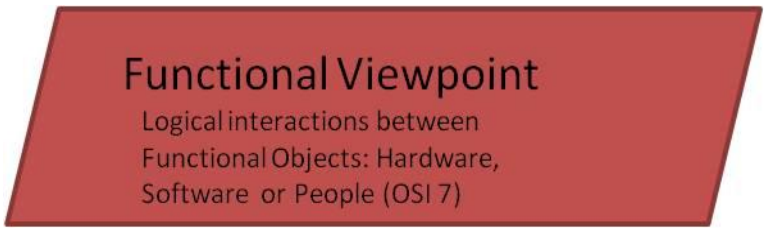
Figure 3-3: Enterprise View Example

Complete listing of information that can be gleaned from this example:

1. *Funding Agency* has a *load and deployment agreement* with *System Acquirer*.
2. *Local Policy Setting Entities* provides *laws and regulations* to *Inspection Bodies.*
3. *System Acquirer* has a *contract* with *System Developer.*
4. *Inspection Bodies* provide *inspection requirements* to *System Developer.*
5. *Inspection Bodies* provide *inspection results* to *System Developer.*
6. *Inspection Bodies* provide *inspection results* to *System Acquirer.*
7. *System Developer* and *System Acquirer* are part of *System Domain.*

Understanding the exact role and makeup of an Enterprise Object is not always intuitive; while they are named as descriptively as possible, an object's role in a particular view may not be so easy to define in two or three words. This is why the object roles and descriptions are provided in text form. Similarly, inferring the verb that defines the relationship between two objects is not always obvious from the drawing; this is why text describing relationships is provided as part of the view.

3.3 Functional Viewpoint



3.3.1 Overview

The Functional Viewpoint addresses the analysis of abstract functional elements and their logical interactions rather than engineering concerns of how functions are implemented, where they are allocated, how they transfer information, which protocols are used, and what method is used to implement them.

The Functional Viewpoint of the Core System focuses on the behavior, structure, and interaction of the functions performed by the system. This Viewpoint addresses Functional Objects, their behavior, the logical connections between them, and the information they exchange.

The behavior of a Function is the set of actions performed by this element to achieve an objective. A **Functional Object** performs actions to achieve an objective of the Core System or to support actions of another Functional Object. This may involve data collection, data transformation, data generation, data generation or processing in performing those actions. Functional Views define Functional Objects to control and manage system behavior, such as monitoring, and other active control elements that are part of describing the functional behavior of the system. They also describe processing functions and the logical flows of information among these Objects.

3.3.2 Definition of Terms

A **Functional Object** is an abstract model of a functional entity that receives requests, performs actions, and generates or processes data. Functional objects that only transport data are called Protocol Entities and are treated explicitly in the Communications Viewpoint. A Functional Object may be a component of another larger Functional Object, which may in turn be a component of a third, even larger, Functional Object. A Functional Object may be implemented by people but most are implemented as software and/or hardware. Implementations of Functional Objects are basic Engineering Objects which are described in the Connectivity Viewpoint.

3.3.3 Stakeholders Addressed

Mobile User, Field User, Center User, Operator, Acquirer, Maintainer, Developer, Manager, Tester, Application Developer, Device Developer, and Service Provider.

3.3.4 Concerns

Interfaces	How difficult is it to develop applications that use Core System interfaces? How flexible are Core System data distribution interfaces? How does the Core System enable control of the services it provides?

Functionality	How does the Core System monitor the services it provides? How does the Core System support the coordination of resources between different Cores? How does the Core System function internally? How do the Core System's components work together? How does the Core System transition between operational modes?
Security	What functional elements are involved in the distribution and revocation of digital certificates, and what roles do those entities have? What functional elements are involved in the detection of misbehavior by System Users, and what roles do those entities have? How does the Core System maintain the integrity of information provided to it by System Users? How does the Core System maintain the privacy of communications between System Users? How does the Core System secure System Users' personal information?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?
Evolvability	How easily can the Core's functionality be expanded to cover new needs if they arise? Does the functionality of the Core scale to support foreseeable demands from System Users?

3.3.5 Objects and Relationships

The following elements may appear in the Functional Viewpoint:

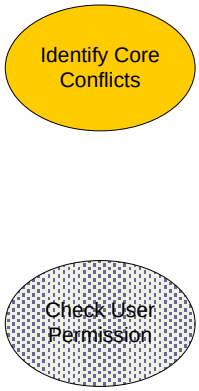
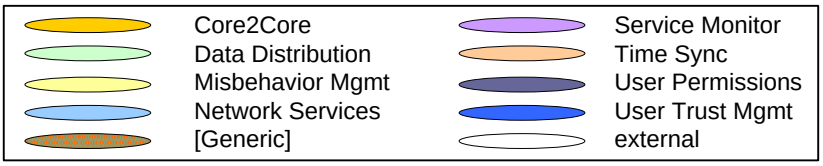
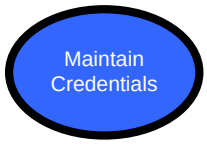
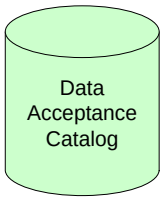
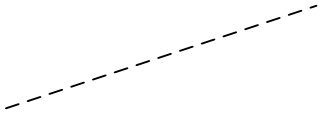
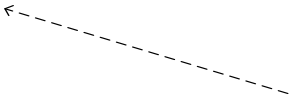
- Functional Objects:
  - Types: Data source, data sink, data transformer, control, planning, monitoring, analysis
  - Attributes: Name, role, behavior, interface definition, data types handled, interaction mode, allocated requirements and constraints
  - Inputs: Control request, service request, data, configuration
  - Outputs: Data, service request, control request, configuration
- Logical Links: connections between Functional Objects, associated with behavior and object properties
- Domains: boundaries of responsibility or ownership
- Information: representations of data that are exchanged between Functional Objects. Formal definition of data to be exchanged may be found in the Information Viewpoint

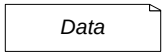
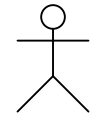
Relationships between Functional Objects: configuration, control flows, data flows, management flows

3.3.6 Method(s) to Model/Represent Conforming Views

Functional View diagrams use the graphical objects defined in Table 3-3.

Table 3-3: Functional View Graphical Object Definitions

	Functional Objects are represented by ovals, with the name of the object in the oval. Objects are named with regard to the role they play in the View. Objects are colored according to the subsystem (see section 5 of the Con-Ops) of which they are a part:  A Functional Object may appear multiple times in a drawing, but only to improve drawing clarity. Duplicates will be shaded as shown.
	Functional Objects that represent optional functions are shaded with horizontal stripes.
	Functional Objects that are secured in some way are depicted with bold outlines.
	Functional Objects with the specialized role of data store are depicted as colored cylinders.
	Logical relationships between Functional Objects are represented with dashed lines. Relationships may be tagged with content type (control, configuration). Relationships may be associated with an Information Object.
	Lines with arrows represent a directional external data flow between a Functional Object and an external Object. Directional external flows may be unidirectional or bidirectional.
	Rectangles with the top right corner turned down represent Information Objects that are exchanged as part of a relationship between Functional Objects. A brief description of the Information Object's contents, source and destination may appear in the Functional View. The Information Object is colored the same as the Functional Object from which it originates. For Information Objects that move within the same subsystem, the labels are placed as close to their destination object as possible. Formal definition of the Information

	Object may be found in the appropriate Information View.
	An Information Object whose sender expects an acknowledgement is named with *italicized text.* This acknowledgement message is not shown on the diagram.
	An Information Object that is secure is drawn with bold outlines. A secure object whose sender expects an acknowledgement has italicized text with bold outlines.
	An external actor is identified as a stick figure with the identity of the actor specified in text below the figure.

Each Functional Object is defined textually. The relationships between Functional Objects are defined in the text describing those objects. The type of relationship may also be depicted in the relationship line on the diagram. Data flow relationships are not labeled, but the Information Objects that are exchanged as part of the relationship are shown adjacent to the relationship line. The Information Object color matches that of the source.

State machines have their own diagram, and are modeled as SysML state machine diagrams.

3.3.7 Viewpoint Source

This Viewpoint is based on the Functional Viewpoint documented in CCSDS 311.0-M-1.

3.3.8 Security Issues

The Functional Objects and services that are used to implement security policies and approaches are defined in the Functional Viewpoint; these include access control interfaces on functions and specific functional elements such as authentication, encryption, certificate distribution, certificate revocation and key management. Some of these may be shown as Functional Objects in their own right (e.g., Public Key Infrastructure (PKI) management function), or just as attributes of other Functional Objects (e.g., access control on a management or control function).

While application of software patches relates directly to objects shown in the Connectivity Viewpoint, the Functional Viewpoint may document functionality that assists in security patch application.

Details concerning these security issues are addressed individually in the text associated with each view.

3.3.9 Views Modeled

The following Functional Views are modeled:

1. Functional View – Top Level: provides a high level Functional View considering the subsystems defined in the Concept of Operations.
2. Functional View – Data Distribution: addresses the Core System's implementation of publish and subscribe.

3. Functional View – System Configuration: addresses the configuration of Core System services and the exchange of configuration information between Cores.
4. Functional View – User Configuration: addresses the creation and modification of System User preferences and Core System interactions.
5. Functional View – System Monitor and Control: addresses the Core System day-to-day maintenance and monitoring operations.
6. Functional View – Credentials Distribution: addresses the distribution of digital certificates for System Users.
7. Functional View – Misbehavior Management: addresses the detection of misbehaving or malfunctioning System Users and Operators, and actions taken to mitigate the effects of their actions.
8. Functional View – Core Decryption: addresses decryption of data sent by a System User in encrypted form and intended for Core System use.
9. Functional View – Networking: addresses the Core's networking functionality.
10. Functional View – Core Backup: addresses the Cores' backing up one another, including data store backup and service backup.

3.3.10 Example View

Figure 3-4 illustrates a sample Functional View. There are three Functional Objects (one of which is a data store), four Information Objects and three relationships and shown in this view. Read a Functional View as a series of input/output relationships between Functional Objects. Inputs are Information Objects positioned on an attached relationship line close to the Functional Object. Outputs are Information Objects positioned far from a Functional Object on an attached relationship line. For example, *Interface to Outside World* accepts *Subscriber ID, function* from an external source and sends *Scanned Subscriber ID, function* to *Transform or Logic Function.*

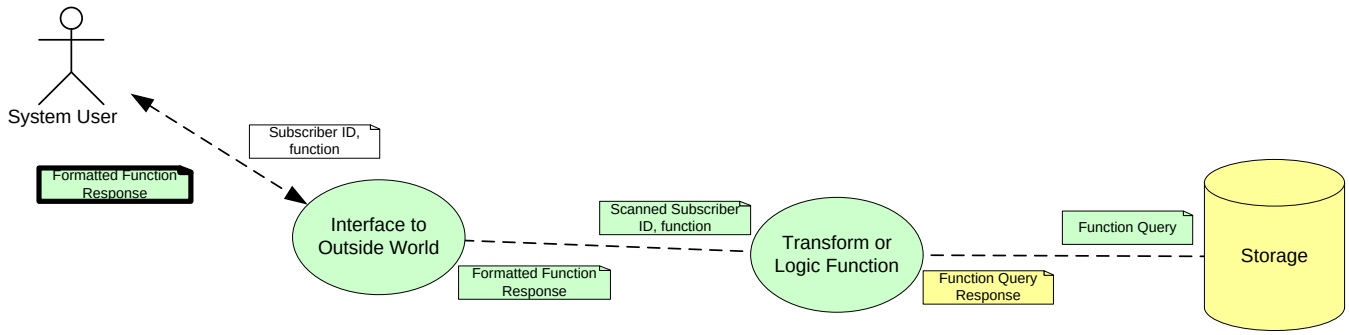


Figure 3-4: Functional View Example

The complete listing of information that can be gleaned from this example is as follows:

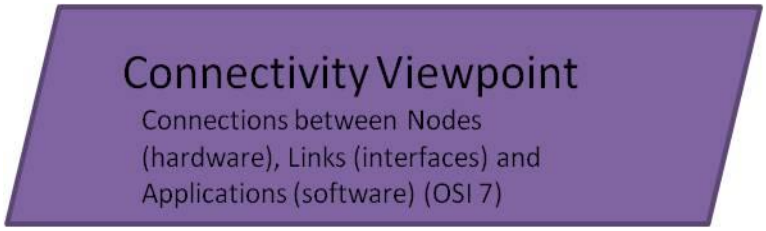
1. *Interface to Outside World* receives *Subscriber ID, function* from System User.
2. *Interface to Outside World* provides *Formatted Function Response* to System User in secure form.
3. *Interface to Outside World* provides *Scanned Subscriber ID, function* to *Transform or Logic Function.*
4. *Interface to Outside World* receives *Formatted Function Response* from *Transform or Logic Function.*
5. *Transform or Logic Function* receives *Scanned Subscriber ID, function* from *Interface to Outside World.*

6. *Transform or Logic Function* provides *Formatted Function Response* to *Interface to Outside World*.
7. *Transform or Logic Function* provides *Function Query* to *Storage*.
8. *Transform or Logic Function* receives *Function Query Response* from *Storage.*
9. *Storage* receives *Function Query* from *Transform or Logic Function.*
10. *Storage* provides *Function Query Response* to *Transform or Logic Function.*

The exact role of a Functional Object cannot be gleaned from the Functional View diagram. While the inputs and outputs are illustrated and the Functional Object is named according to its role, a complete definition does not fit inside the object drawing. The associated object definition and view text defines the function(s) that a given Functional Object performs.

Functional Objects receive data, send data, transform data, and/or provide control (of other objects or interfaces), planning (chiefly a human function), monitoring or analysis. Functional Objects may perform one or more such functions. An object that performs more than one function could be decomposed into multiple Functional Objects. The Functional Views in the SAD include objects that are fully decomposed and attributable to one requirement, and other objects that satisfy multiple requirements and could be more fully decomposed. The level of decomposition chosen for a given function or set of functions was chosen to characterize the behavior of the Core System and describe alternatives while maintain readability. Prior to system development, functions that can be decomposed will be, but on separate view diagrams.

3.4 Connectivity Viewpoint



3.4.1 Overview

The Connectivity Viewpoint represents physical elements that operate in the field and the back office, where connections between elements and interactions with the external environment are considered. The Connectivity Viewpoint deals with the composition of these physical elements (including application servers, data stores, network components, wired and wireless links), their physical connections and interactions, and the allocation of functionality to those elements. For analysis of the Core System and its relationship with other elements of the *connected vehicle* environment, the focus is on nodes, links, computational and data transport functions.

Mobile Users are in motion and consequently connectivity issues exist associated with the wireless communications media used to interact with the Core System or other users.

The **Connectivity Viewpoint** is an engineering view that shows Engineering Objects, which may be hardware or software. The Connectivity Viewpoint is focused on a **Node** and **Link** view of a system, the composition of the Nodes, the physical connections among Nodes, their physical and environmental constraints, and their physical dynamics. The Connectivity Viewpoint also describes how the abstract functional design described in the Functional Viewpoint is to be implemented as software Engineering Objects, i.e., applications or software components or hardware Engineering Objects, and how these are allocated on the major hardware Engineering Objects (Nodes) of the system.

3.4.2 Definition of Terms

An **Engineering Object** is an implementation or realization of some abstract function. It may be implemented as hardware (Node) or as software (application or software component).

A **Node** is a physical hardware Engineering Object that is a run-time computational resource and generally has at least memory and processing capability. Run-time software Engineering Objects reside on nodes. A Node has some well-understood, possibly rapidly moving, location. A Node may be composed of two or more **(sub) Nodes**.

A **Link** is the locus of relations among Nodes. It provides interconnections between Nodes for communication and coordination. It may be implemented by a wired connection or with some radio frequency (RF) or optical communications media. Links implement the primary function of transporting data. Links connect to Nodes at a Port.

A **Port** is the physical element of a Node where a Link is connected. Nodes may have one or more Ports.

3.4.3 Stakeholders Addressed

Mobile User, Field User, Center User, Operator, Acquirer, Maintainer, Developer, Tester, Application Developer, Device Developer, Service Provider.

3.4.4 Concerns

Performance	Can the Core System provide services with sufficient responsiveness to ena-

	ble System User applications? Can the Core meet all of the performance requirements defined in the SyRS (e.g., availability, reliability, capacity, and other quantitative measures)?
Interfaces	Can the Core System meet the interface requirements defined in the SyRS?
Security	What physical elements are involved in the distribution and revocation of digital certificates, and what are their roles? How are the Core System components secured from network attack? How are the Core System components physically secured?
Feasibility	Are Core System services feasible to develop given current technology and resources?
Risks	Is the Core System's hardware and software architecture susceptible to failure, and if so under what circumstances? What are the characteristics of this failure?
Evolvability	Is the structure of the Core System sufficiently flexible and scalable to deploy and to enable changes to cover new needs if they arise?
Deployability	Are the Engineering Objects that make up the Core System practical to deploy in the projected deployment environment and in a timely manner?
Maintainability	Is the structure of the Core System maintainable with a reasonable allocation of resources for the entities that are likely to consider deployment? Can the Core's functionality be sustained with acceptable levels of downtime as per the SyRS?

3.4.5 Objects and Relationships

The following elements may appear in the Connectivity Viewpoint:

- Engineering Objects:
    - Nodes (Hardware Engineering Objects (HEO))
        - Types: hardware objects, ports
        - Attributes: Name, type, location, available resources, physical interfaces, capabilities (e.g., processing, memory, bandwidth, throughput, etc.), allocated functions and constraints
        - Inputs/Outputs: Links to/from other Nodes
    - Links (connections between Nodes)
        - Types: Wireless, physical link
        - Attributes: Name, type, end-points (ports), physical interfaces, performance, access, ownership
    - Applications (Software Engineering Objects (SEO))
        - Types: software engineering objects

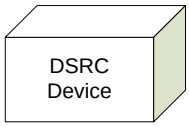
  - Attributes: name, type, algorithms, implemented functions, allocation to nodes, required resources, implemented interfaces, implementation constraints (e.g., language, operating system, framework, etc.)
- Environment: physical environments, physical interactions and effects
- Information: defined representation of data that are exchanged among Engineering Objects. Formal definition of data to be exchanged may be found in the Information Viewpoint

Relationships between Engineering Objects: composition, interfaces, constraints, configuration

3.4.6 Method(s) to Model/Represent Conforming Views

Connectivity View diagrams use the graphical objects defined in Table 3-4.

Table 3-4: Connectivity View Graphical Object Definitions

	Nodes are represented by 3-dimensional shaded boxes drawn with solid lines, with the name of the Node on the front face. Nodes are named with regard to the role they play in the View. Objects are colored according to the subsystem (see section 5 of the ConOps) of which they are a part: 
	A link between Nodes is shown as a straight solid line. The link may be tagged with content type (control, service, or data). This link is one that could be implemented using a wireless or wired connection.
	A link between Nodes may also be shown with a jagged line. The link may be tagged with content type (control, service, or data). This link with a jagged line is one that is typically implemented using a wireless connection.
	Software Engineering Objects are represented by boxes with rounded edges.
	Rectangles with the top right corner turned down represent Information Objects that are exchanged as part of a relationship between Engineering Objects. The Information Object definition may be found in the appropriate Information View.
	Ports are modeled as small boxes.

Each Node is defined textually. The links between Nodes are defined in the text describing those Objects.

State machines have their own diagram, and are modeled as SysML state machine diagrams.

3.4.7 Viewpoint Source

This Viewpoint is based on the Connectivity Viewpoint documented in CCSDS 311.0-M-1.

3.4.8 Security Issues

In the Connectivity Viewpoint security issues are dealt with by the physical elements that are used to implement security policies and barriers. These include routers and firewalls, hardware encryption devices, and physical boundaries such as access-controlled rooms.

Installation of security patches to Software Engineering Objects contributes to maintaining the security of the Core System. Patching software implies that the relevant component is not operational, so mechanisms must be provided to maintain system availability. This could include backup functionality (see the Functional Viewpoint). The Connectivity Viewpoint may illustrate mechanisms to maintain availability, including the distribution of duplicate Engineering Objects to allow operations of some software while other software is being patched.

The protocol entities that may implement elements of security functionality such as security protocols will be addressed in the Communications Viewpoint.

3.4.9 Views Modeled

Three Connectivity Views are modeled:

1. Connectivity View – High Level: addresses the basic component structure of the Core System.
2. Connectivity View – Core System Functional Allocation: addresses functional allocation to Core components.
3. Connectivity View – State and Mode Transitions: addresses Engineering Object state and mode transitions.

3.4.10 Example View

Figure 3-5 illustrates a sample Connectivity View. There are two Nodes, two links, 8 SEOs and four Information Objects on this drawing. SEOs are placed within the Node that they operate on. All inputs from and outputs from sources and sinks external to the Node must come through a link.

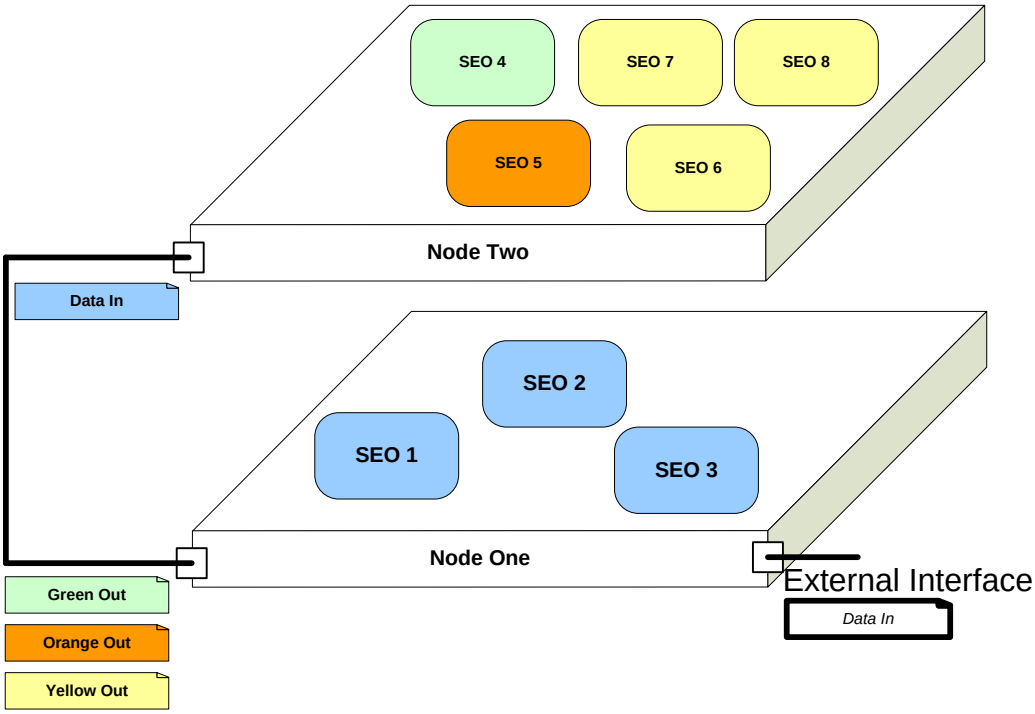


Figure 3-5: Connectivity View Example

In this example, *SEO 1*, *SEO 2* and *SEO 3* all send *Data In* to *Node Two*. *SEO 4* sends *Green Out*, *SEO 5* sends *Orange Out*, and *SEO 6*, *SEO 7* and *SEO 8* send *Yellow Out* to *Node One*. *Data In* is received on *Node One's* external interface. Exactly what each SEO accomplishes would be specified in text associated with the view.

3.5 Communications Viewpoint

3.5.1 Overview

The Communications Viewpoint defines the layered sets of communications protocols that are required to support communications among the Engineering Objects that compose the Core System and between the Core and System Users. These protocols need to meet the requirements on performance and the constraints imposed by physical connectivity, environmental and operational challenges, and relevant policies (such as the assurance of anonymity for mandatory data provision).

The Communications Viewpoint focuses on the design and implementation of protocols and communications standards, including implementation choices, and specification and allocation of communications functionality to engineered components of the system.

In the Communications Viewpoint the Core System is depicted with Communications Objects that are called Protocol Entities. For context, these are often shown along with representations of the Nodes, Links, and software Engineering Objects that are defined more fully in the Connectivity Viewpoint. The Communications Viewpoint describes the protocols that are required for the software Engineering Objects actually to communicate with one another and supports descriptions of the *connected vehicle* environment.

3.5.2 Definition of Terms

A **Protocol Entity** performs actions to exchange or transfer data (as distinguished from a Functional Object that generates or processes data). Protocol Entities are used to support interactions between two Engineering Objects or among groups of Engineering Objects that are contained in separate Nodes (see the Connectivity Viewpoint for discussion of Engineering Objects and Nodes). Protocol Entities are often shown as two peer entities communicating with each other over a Link between connected Nodes. The Engineering Objects that use services provided by Protocol Entities may be implemented in hardware or software, and the Protocol Entities themselves may be implemented in hardware or software.

3.5.3 Stakeholders Addressed

Mobile User, Center User, Field User, Operator, Acquirer, Manager, Maintainer, Developer, Tester, Policy Setter, Application Developer, Device Developer, Service Provider.

3.5.4 Concerns

Performance	Do the communications protocols allow the Core to meet the performance requirements defined in the SyRS?
Interfaces	Do communications protocols allow the Core to meet the interface requirements defined in the SyRS?
Functionality	What functionality exists in the communications protocols used by the Core? What reliability features are included in the communication protocols?

Security	What provisions for ensuring the privacy of communications by System Users are included in the communications protocols? How do the communications protocols provide non-repudiation of messages sent to and from the Core System? How do the communications protocols protect the integrity of messages sent to and from the Core System.
Organization/Resources	What resources are required to develop, deploy, operate and maintain the communications protocols that the Core System needs?
Appropriateness	Do the communications protocols fulfill the needs set forth in the ConOps and support applications that meet the overall *connected vehicle* goals?
Feasibility	Are the communications protocols required by the Core System feasible to specify, develop, and deploy?
Risks	If communications protocols chosen are developed by independent bodies are there any risks associated with changes to those standards that may impact the applicability of those standards? If communications protocols chosen are developed by independent bodies are there any risks associated with the timely publication of those standards?
Evolvability	Are the communications protocols chosen scalable to support foreseeable demands from System Users?
Deployability	Are the communications protocols required by the Core practical to deploy from a capital and human resource perspective?

3.5.5 Objects and Relationships

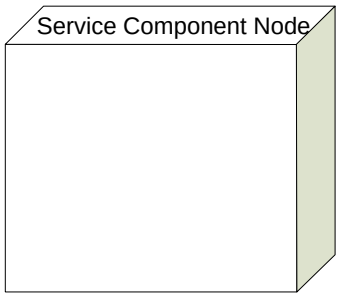
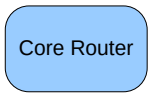
The following elements may appear in the Communications Viewpoint:

- Protocol Objects:
  - Types: protocol purpose (e.g., coding, link, network, transport, etc.)
  - Attributes: Name, type, capabilities (e.g., in order, once only, bandwidth efficient, error correcting, delay tolerant), constraints, services (offered, required), interface signature (requests, indications, responses, confirmations), standard reference identifier, standards organization, management interface
  - Inputs and Outputs defined above under attributes
- Nodes and Links: representations of the physical elements from the Connectivity Viewpoint, for context
- Software Engineering objects: representations of implemented functions from the Connectivity Viewpoint for context

3.5.6 Method(s) to Model/Represent Conforming Views

Communications View diagrams use the graphical objects defined in Table 3-5.

Table 3-5: Communications View Graphical Object Definitions

	Protocol entities are represented by rectangles with the name of the entity inside.
	Nodes are represented by 3-dimensional shaded boxes drawn with solid lines, with the name of the Node printed above or on the top face of the Node. The Node definition may be found in the appropriate Connectivity View.
	The Link between nodes that connects to the lowest layer protocol entity is shown as a solid line. Extensions of this link go between protocol entities, and between protocol entities and Software Engineering Objects.
	The logical Link between Protocol Entities operating at the same communications layer, or between Software Engineering Objects, is shown with a dashed line.
	Software Engineering Objects are represented by boxes with rounded corners. Software Engineering Objects in the Communications View implement one or more Functional Objects from a Functional View. Software Engineering Objects may be in a Connectivity View.

3.5.7 Viewpoint Source

This Viewpoint is based on the Communications Viewpoint documented in CCSDS 311.0-M-1.

3.5.8 Security Issues

Certain functions for implementing data system security may be allocated to the Communications Viewpoint. These will typically include network layer, transport layer and session layer security protocols. These functions need to be traced to the SyRS. Different physical communications media may require different upper-layer protocols.

Implementation of security protocols between Mobile Users and the Core System, and Mobile Users and other System Users, is a primary concern tracing back to many of the needs documented in the ConOps.

3.5.9 Views Modeled

Four Communications Views are modeled:

1. Communications View – Mobile DSRC Device and Core: addresses Mobile Users communicating with the Core System using DSRC communications.

2. Communications View – Mobile Wide-Area Wireless User and Core: addresses Mobile Users communicating with the Core using cellular or Wi-Fi communications.
3. Communications View – Fixed Point Center/Field User and Core, Core2Core: addresses Cores communicating with fixed-point Field and Center Users, and Cores communicating with one another.
4. Communications View – Core Routing: addresses the Core's routing function, where the Core facilitates communications between private networks.

3.5.10 Example View

Figure 3-6 illustrates a sample Communications View. This view includes two Nodes, four SEOs (two Application layer and two operating at more than one protocol layer), six protocol entities, two physical links between Nodes that extend through protocol entities two SEOs, and two logical links between Protocol Entities.

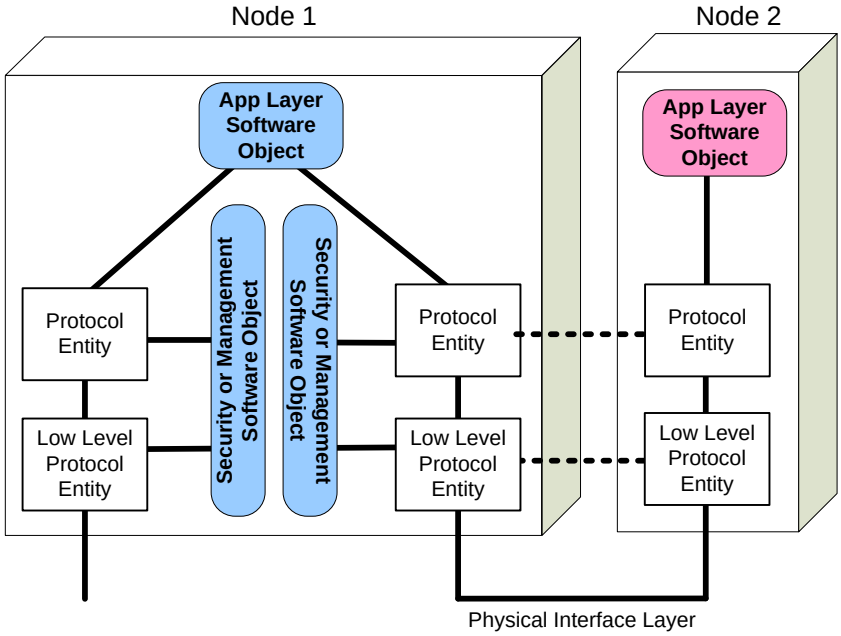


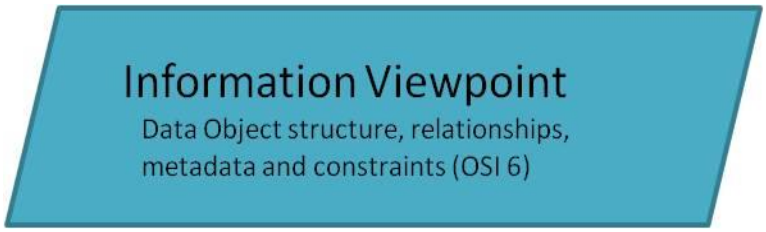
Figure 3-6: Communications View Example

Communications Views should be looked at starting from the top of a Node *down* to the bottom, then on to the next Node along the physical interface link, and then back *up* to the top element in the stack on that Node. For the example given, consider the journey of a packet (which could be anything including data, control, service request etc.) that starts at Node2's Application level Software Object

1. Generated at *Node 2's Application level Software Object*
2. Passes through *Node 2's Protocol Entity*
3. Passes through *Node 2's Low Level Protocol Entity*
4. Passes across the *Physical Interface Layer* between *Node 2* and *Node 1*
5. Goes to *Node 1's Low Level Protocol Entity* which has a logical relationship with *Node 2's Low Level Protocol Entity*. From here the diagram implies that the packet could head two directions:
    a. Goes to *Node 1's Security or Management Security Object*
    b. Goes to *Node 1's Protocol Entity* which has a logical relationship with *Node 2's Protocol Entity*. From here the diagram implies that the packet could head two directions:
        i. Goes to *Node 1's Security or Management Security Object*
        ii. Goes to *Node 1's Application level Software Object*

The associated text description may clarify what happens in steps 5 above. For example, the packet may be held at *Node 1's Low Level Protocol Entity* until *Security or Management Security Object* provides a control response to pass it on. Where possible, Protocol Entities physical interface layers will be specified, e.g. Internet Protocol (IP) version 6 (IPv6), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), 802.11a, etc.

3.6 Information Viewpoint



3.6.1 Overview

The Information Viewpoint describes the data objects that are passed among the elements in the Core System, and between the Core and System Users. These data objects may have different elements, structures, semantics, relationships, and policies. The Information Viewpoint is used to address the data architecture and definition aspects of the Core. Representations of the Information Objects that are fully defined in this Viewpoint appear in other Viewpoints. They are managed (i.e., stored, located, accessed and distributed) by information infrastructure elements and also shown as being passed among Functional Objects.

The Information Viewpoint specification of the Core System focuses on the information used by the system. This is limited to views of information content and the relationships among information elements.

3.6.2 Definition of Terms

Information Objects are descriptions of data along with the necessary structure and syntax to allow interpretation and use of these Objects. An Information Object may also have associated metadata, and Information Views may define the relationships among Information Objects, rules for their use and transformation, and policies on access.

Metadata is 'data about data', the information that describes content. It is information about the meaning of data, as well as the relationships among Information Objects, rules for their use and transformation, and policies on access.

An **Information Package** consists of a primary Information Object, with associated Metadata necessary to use the Information Object.

3.6.3 Stakeholders Addressed

Mobile User, Field User, Center User, Operator, Acquirer, Maintainer, Developer, Tester, Application Developer, Device Developer.

3.6.4 Concerns

Security	Do any Information Objects potentially impact the privacy of System Users?
Interfaces	Are there any standards that define applicable message sets for Core System Information Objects? Which interfaces are candidates for standardization?
Appropriateness	Do the Information Objects convey information sufficient to realize Core System functionality as defined in the SyRS?

3.6.5 Objects and Relationships

The following elements may appear in the Information Viewpoint:

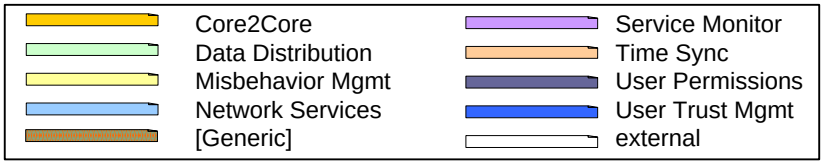
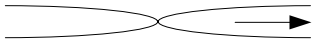
- Information Objects:
  - Types: data, metadata, schema, model
  - Attributes: Name, type, length, semantics, rules, policies
  - Inputs and Outputs defined above under attributes
- Constraints: permanence, policies

Relationships between Information Objects: aggregation, transformation

3.6.6 Method(s) to Model/Represent Conforming Views

Information View diagrams use the following graphical objects:

Table 3-6: Information View Graphical Object Definitions

	Rectangles with the top right corner turned down represent Information Objects. Objects are colored according to the subsystem (see section 5 of the ConOps) they originate from: 
	Lines between objects indicate a "part-of" relationship. The Parent Information Object is at the top and the child object is to the right and below.
	Vertically angled solid lines between objects indicate an "aggregation" relationship, where multiple instantiations of one Object are combined to create a new Object that is dependent on the contents of the original Object instantiations. The new aggregated Object is on top and the Object it is dependent on is on the bottom.
	Curving lines that cross in the middle between objects indicate a "Transformation" relationship, where one Information Object is modified through the application of some other information or process into another Information Object. The resulting Information Object is indicated by the arrowhead.

3.6.7 Viewpoint Source

This Viewpoint is based on the Information Viewpoint documented in CCSDS 311.0-M-1.

3.6.8 Security Issues

Information Objects may contain information that needs to be protected from authorized access. These objects must be identified so that appropriate security mechanisms can be architected in the Communications and Functional Viewpoints.

3.6.9 Views Modeled

Two Information Views are modeled:

1. Information View – Top Level External Objects: addresses the Information Objects that the Core sends to and receives from System Users.
2. Information View – Top Level Internal Objects: addresses the Information Objects passed between Core subsystems.

3.6.10 Example View

Figure 3-7 illustrates a sample Information View. There are a total of five object types (of varying levels) and ten objects shown on this view. Each Information Object inherits the characteristics of the object types above it.

For example:

- All objects and object types inherit all characteristics of *Subsystem 1 Objects*
- *Sub-object Type 1* inherits all characteristics of *Object Type 1* and *Subsystem 1 Objects*
- *ST1 Sub2-object 1* is an Information Object that inherits the characteristics of *Sub-object Type 1, Object Type 1* and *Subsystem 1 Objects*

Object 4 is an aggregation of *Object 3*. The exact nature of that aggregation (how many *Object 3* are used to create *Object 4* and the algorithm by which this is done) is not clear; it may be specified in the view text. For instance, *Object 4* could be the average of the values included in 10 *Object 3*, or it could be the median of all *Object 3* received in a two minute period, or any other similar sort of aggregation.

Object 5 and *Object 6* have a transformation relationship. The exact nature of that relationship is not clear; it may be specified in the view text. It could be a formatting difference, or one object could be used to create another according to some algorithm.

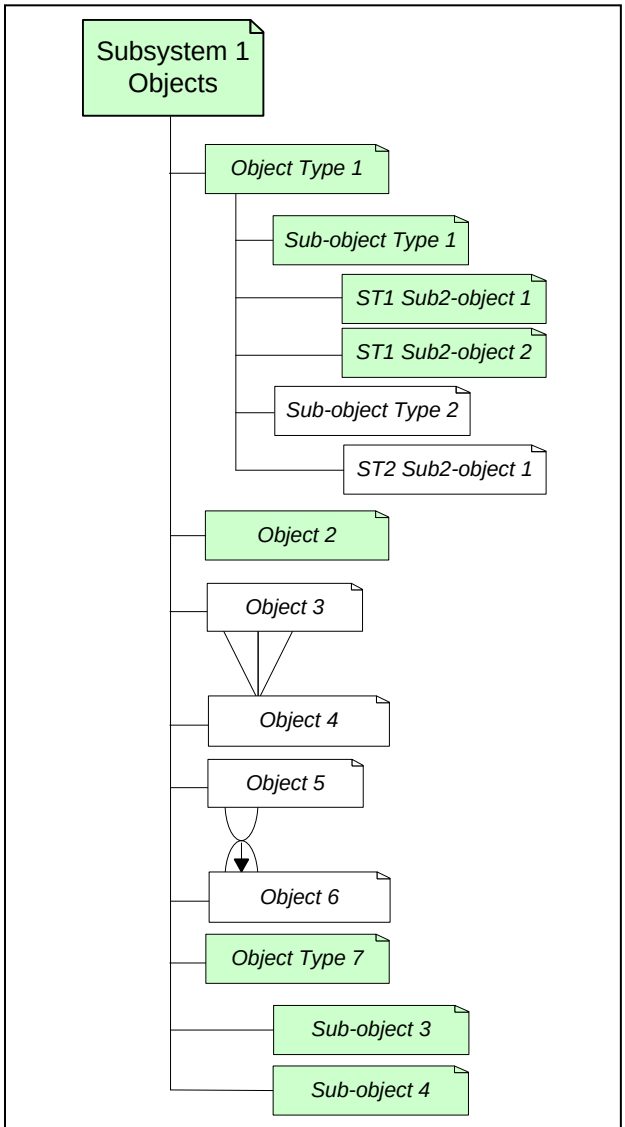


Figure 3-7: Information View Example

4.0 ARCHITECTURAL VIEWS

This section of the System Architecture Document contains views for each viewpoint listed in Section 3. Each view is documented as follows:

Section #	Section Name	Description
4.x	**Name of Viewpoint**	Name of the Viewpoint x
4.x.y	**Name of View**	Name of the View y of Viewpoint x
4.x.y.1	**Introduction**	A brief introduction to the view is provided
4.x.y.2	**Concerns Addressed by this View**	Lists the concerns from chapter 2 that are addressed by this view.
4.x.y.3	**Object Definitions and Roles**	Defines all of the objects that appear in the view, their function and or purpose, and relationship to other objects.
4.x.y.4	**View Description**	A graphical representation of the system is constructed with the methods of the associated viewpoint, accompanied by a textual discussion.
4.x.y.5	**Configuration Information**	Identifies which other views need to be considered when managing updates to this view. Changes to this view may necessitate changes to these other views.

4.1 Enterprise Viewpoint

The Enterprise Views illustrate relationships that document institutional issues and relationships. These relationships affect the other Viewpoints because of the interfaces the relationships imply. Enterprise Views in the SAD were selected to illustrate the number and variety of entities and the relationships that could be involved in operations, deployment and management of Core Systems. These views illustrate all of the Core's external interfaces, and the support relationships between external enterprises that affect the *connected vehicle environment.*

Six Enterprise Views are presented:

- Distribution of security credentials to System Users
- Core System operations
- Development and Deployment of the Core System and System User applications
- Core System Configuration and Maintenance
- Governance
- Business Model Facilitation

Table 4-1 shows which Enterprise Views are relevant to each stakeholder.

Table 4-1: Enterprise View Stakeholder Matrix

Information View \ Stakeholder	Mobile User	Field User	Center User	Operator	Acquirer	Maintainer	Developer	Manager	Tester	Policy Setter	Application Developer	Device Developer	Service Provider
Security Credentials Configuration	■	■	■	■	■		■	■		■	■	■	
Operations	■	■	■	■	■	■	■	■		■	■	■	■
Core System and Application Development and Deployment					■	■	■	■		■	■		
Configuration and Maintenance				■	■	■	■	■		■	■		■
Governance	■	■	■	■	■	■	■	■		■	■	■	■
Business Model Facilitation	■	■	■	■	■		■	■		■	■		

The following Enterprise Views are relevant to each Stakeholder as shown in the Table above:

- Mobile Users: Security Credentials Configuration, Operations, Governance, and Business Model Facilitation
- Field Users: Security Credentials Configuration, Operations, Governance, and Business Model Facilitation
- Center Users: Security Credentials Configuration, Operations, Governance, and Business Model Facilitation
- Operators: Security Credentials Configuration, Operations, Configuration and Maintenance, Governance, and Business Model Facilitation

- Acquirers: Security Credentials Configuration, Operations, Core System and Application Development and Deployment, Configuration and Maintenance, Governance, and Business Model Facilitation
- Maintainers: Operations, Core System and Application Development and Deployment, Configuration and Maintenance, and Governance
- Developers: Security Credentials Configuration, Operations, Core System and Application Development and Deployment, Configuration and Maintenance, Governance, and Business Model Facilitation
- Managers: Security Credentials Configuration, Operations, Core System and Application Development and Deployment, Configuration and Maintenance, Governance, and Business Model Facilitation
- Testers: none
- Policy Setters: Security Credentials Configuration, Operations, Core System and Application Development and Deployment, Configuration and Maintenance, Governance, and Business Model Facilitation
- Application Developers: Security Credentials Configuration, Operations, Core System and Application Development and Deployment, Configuration and Maintenance, Governance, and Business Model Facilitation
- Device Developers: Security Credentials Configuration, Operations, and Governance
- Service Providers: Operations, Configuration and Maintenance, and Governance

4.1.1 Enterprise View – Security Credentials Distribution

4.1.1.1 Introduction

The safety-critical nature of many DSRC/WAVE applications makes it vital to protect messages from attacks such as spoofing, alteration, and replay. To accomplish this, the DSRC devices use IEEE 1609.2 digital certificates to digitally sign their messages, which the Core System, Field and Mobile Users may use to verify the authenticity of DSRC messages.

The Core System does not distribute certificates directly, but relies on External Support Systems functioning as Registration Authority (RA) and Certificate Authority (CA). Current research conducted by CAMP indicates that a small number of RAs (approximately 5), one CA and an undetermined number of Linkage Authorities (LA) will be required to ensure the anonymity of Mobile Users. The LA is not documented as part of the architecture because it has no relationship with the Core, and its relationships with the RA and CA do not impact the Core.

Standards bodies such as the Internet Engineering Task Force (IETF) and IEEE have drafted standards for configuration and distribution of security credentials in the Internet environment. IEEE is developing security standards suitable for implementing certificate-based trust schemes using DSRC-based communications in IEEE 1609.2.

Companies like VeriSign, Entrust, and Microsoft have developed security products based on IETF standards. Core System Deployers will need to create policies related to security credentials for devices operating in their domain. Deployers can procure Commercial-off-the-Shelf (COTS) products for managing security credentials or outsource this activity (e.g., to VeriSign or Entrust).

This view depicts the Enterprise Objects involved in the mission of configuring Security Credentials for Center, Mobile and Field Users.

4.1.1.2 Concerns Addressed by this View

Security	What entities are involved in the distribution of digital certificates, and what roles do those entities have? What entities are involved in the detection of misbehavior by System Users, and what roles do those entities have?
Organization/Resources	How do the entities responsible for a Core System need to interact with entities responsible for other Core Systems?
Risks	What relationships between Core Enterprise Objects and external Enterprise Objects provide risks to Core System development, deployment, operations, and maintenance? What steps can be taken to lessen risks that are a function of Enterprise relationships?

4.1.1.3 Object Definitions and Roles

4.1.1.3.1 Center Owner/Operator

The Center Owner/Operator is the owner and operator of the Center that interfaces with the Core System. For example, this could be an information service provider or traffic management center. This object also includes the Center facility which is not drawn separately to help keep the diagram readable.

Relationships:

- The Center Owner/Operator registers his device with the Core System Facility; the Core System Facility provides credentials that the Center can use when interacting with the Core.
- The Center Owner/Operator establishes and negotiates special permissions for use of the Core System with the Core System Manager. These permissions are based on needs of the Center, local laws, regulations and policies.
- The Center Owner/Operator provides misbehavior reports, describing activity of other System Users that it considers misbehavior, to the Core System.
- The Center Owner/Operator receives Certificate Revocation Lists (CRLs) from the Core System Facility.

4.1.1.3.2 Core Certification Authority

The Core Certification Authority verifies that a Core System and other connected vehicle devices meet specification set forth in the USDOT's documents and associated standards. For additional information on the Core Certification Authority, see Enterprise View – Governance.

Relationships:

- The Core Certification Authority receives the Core System requirements and architecture from the USDOT Core System Documents.
- The Core Certification Authority receives applicable standards from Standards Bodies.
- The Core Certification Authority creates the ESS DSRC Registration Authority Certification Plan and shares this with the ESS DSRC Registration Authority. Once it certifies the ESS DSRC Registration Authority, it provides the certification results to the ESS DSRC Registration Authority, ESS DSRC Certificate Authority and Core System Manager. It also provides periodic re-certification results.
- The Core Certification Authority creates the ESS DSRC Certificate Authority Certification Plan and shares this with the ESS DSRC Certificate Authority. Once it certifies the ESS DSRC Certificate Authority, it provides the certification results to the ESS DSRC Certificate Authority and Core System Manager. It also provides periodic re-certification results.

4.1.1.3.3 Core System Deployer

The Core System Deployer deploys the Core System.

Relationships:

- The Core System Deployer receives recommended best practices from National Policy Setting Entities
- The Core System Deployer receives policies related to user authorization from Local Policy Setting Entities
- The Core System Deployer provides operational policies and policies related to user authorization to the Core System Manager.

- The Core System Deployer receives applicable standards from Standards Bodies.
- The Core System Deployer enters into agreements with other Core System Deployers, describing the scope of each Core's credentials distribution, the hierarchy of CAs, and information to be shared to manage CRLs.

4.1.1.3.4 Core System Facility

The Core System Facility is the hardware and software that provides Core services.

Relationships:

- The Core System Facility is controlled by the Core System Manager.
- The Core System Facility receives the IEEE 1609.2 CRL from the ESS DSRC Certificate Authority. It sends requests for additions to the CRL based on misbehavior reporting to the ESS DSRC Certificate Authority.
- The Core System Facility receives the X.509 CRL from the ESS X.509 Certificate Authority.
- The Core System Facility receives registration information from the Center Owner/Operator; the Core System Facility provides credentials that the Center can use when interacting with the Core.
- The Core System Facility receives misbehavior reports from the Mobile Device.
- The Core System Facility receives misbehavior reports from the Field Node Owner/Operator.
- The Core System Facility receives misbehavior reports from the Center Owner/Operator.
- The Core System Facility exchanges misbehavior reports with other Core System Facilities.
- The Core System Facility exchanges Certificate Revocation Lists with other Core System Facilities.
- The Core System Facility provides CRL Change requests based on misbehavior identification to the ESS DSRC Certificate Authority.
- The Core System Facility provides CRLs to the Center Owner/Operator, the Field Node Owner/Operator, and the Mobile Device.

4.1.1.3.5 Core System Manager

The Core System Manager manages the operations of the Core System. This includes daily operations activities such as configuration changes, backups and other system administration tasks, and maintenance activities.

Relationships:

- The Core System Manager receives operational policies and authorization criteria for users from the Core System Deployer.
- The Core System Manager negotiates special user permission information with the Mobile Device Operator. These permissions are based on needs of the Mobile Device Operator, local laws, regulations and policies.
- The Core System Manager negotiates special user permission information with the Field Node Owner/Operator. These permissions are based on needs of the Field Node, local laws, regulations and policies.

- The Core System Manager negotiates special user permission information with the Center Owner/Operator. These permissions are based on needs of the Center, local laws, regulations and policies.
- The Core System Manager provides special user permission information to the ESS DSRC Registration Authority.
- The Core System Manager receives local laws and regulations from Local Policy Setting Entities.
- The Core System Manager receives scope and contact information from the ESS DSRC Certificate Authority so that it may maintain an understanding of what entities are responsible for granting and revoking certificates and under what conditions they grant and revoke certificates.
- The Core System Manager negotiates an agreement with the ESS DSRC Certificate Authority whereupon the Core System Manager will provide CRL change requests based on misbehavior analysis, and the ESS DSRC Certificate Authority will add entries to the CRL based on those requests.

4.1.1.3.6 ESS DSRC Certificate Authority

The ESS DSRC Certificate Authority takes the role of CA in the public-key infrastructure architecture used to support IEEE 1609.2 certificate distribution.

Relationships:

- The ESS DSRC Certificate Authority receives the DSRC Certificate Authority Certification Plan, certification results and periodic re-certification results from the Core Certification Authority.
- The ESS DSRC Certificate Authority receives standards from Standards Bodies.
- The ESS DSRC Certificate Authority provides scope and contact information to the Core System Manager, so that the Core System Manager may maintain an understanding of the ESS DSRC Certificate Authority's area of responsibility, and conditions for granting and revoking certificates.
- The ESS DSRC Certificate Authority receives recommended practice information from National Policy Setting Entities.
- The ESS DSRC Certificate Authority enters into an operating agreement with the ESS DSRC Registration Authority so the ESS DSRC Registration Authority can obtain certificates for DSRC devices from the ESS DSRC CA. The ESS DSRC Certificate Authority receives credential requests to the RA, and provides IEEE 1609.2 certificates.
- The ESS DSRC Certificate Authority receives CRL Change Requests from the Core System Facility based on its analysis of misbehavior reports.

4.1.1.3.7 ESS DSRC Registration Authority

The ESS DSRC Registration Authority takes the role of RA in the public-key infrastructure architecture used to support IEEE 1609.2 certificate distribution.

Relationships:

- The ESS DSRC Registration Authority receives the DSRC Registration Authority Certification Plan, certification results, and periodic re-certification results from the Core Certification Authority.

- The ESS DSRC Registration Authority receives special user permission information from the Core System Manager.
- The ESS DSRC Registration Authority enters into an operating agreement with the ESS DSRC Certificate Authority so the ESS DSRC Registration Authority can obtain certificates for DSRC devices from the ESS DSRC CA. It provides credential requests to the CA and receives IEEE 1609.2 certificates from the CA.
- The ESS DSRC Registration Authority enters into an agreement with the Mobile Device Owner so the ESS DSRC Registration Authority can obtain and configure IEEE 1609.2 certificates for the Mobile Device.
- The ESS DSRC Registration Authority enters into an agreement with the Field Node Owner/Operator so the ESS DSRC Registration Authority can obtain and configure IEEE 1609.2 certificates for the devices operated by the Field Node Owner/Operator.
- The ESS DSRC Registration Authority receives security credential requests from the Mobile Device. It provides certificates and CRLs to the Mobile Device.

4.1.1.3.8 ESS X.509 Certificate Authority

The ESS X.509 Certificate Authority provides X.509 certificates to Mobile Devices and the CRL to the Core System Facility.

Relationships:

- The ESS X.509 Certificate Authority enters into an agreement with the Mobile Device Owner for the ESS X.509 Certificate Authority to provide certificates to the Mobile Device.
- The ESS X.509 Certificate Authority receives certificate requests from Mobile Devices and provides them X.509 certificates.
- The ESS X.509 Certificate Authority provides the X.509 CRL to the Core System Facility.
- The ESS X.509 Certificate Authority receives recommended best practices from the National Policy Setting Entities.

4.1.1.3.9 Field Node Owner/Operator

The Field Node Owner/Operator is the owner and operator of the field infrastructure used by the Mobile Device Operator to interface with the Core System. This could be a cellular provider or DSRC field infrastructure manager for example.

Note that while this view is concerned with the provision of certificates, particularly those that are to be used over 5.9GHz DSRC, those certificates could be provided using other forms of communications infrastructure than 5.9 GHz DSRC. Cellular 3G/4G networks are one option, but so are municipal, commercial or home wireless networks, particularly for stationary devices.

This object also includes the Field Node facility which is not drawn separately to help keep the diagram readable.

Relationships:

- The Field Node Owner/Operator registers his device with the Core System Facility; the Core System Facility provides credentials that the device can use when interacting with the Core.

- The Field Node Owner/Operator establishes negotiates special permissions for use of the Core System with the Core System Manager. These permissions are based on needs of the Field Node, local laws, regulations and policies.
- The Field Node Owner/Operator enters into an agreement with the ESS DSRC Registration Authority so the ESS DSRC Registration Authority can obtain and configure IEEE 1609.2 certificates for the devices operated by the Field Node Owner/Operator.
- The Field Node Owner/Operator provides misbehavior reports, describing activity of other System Users that it considers misbehavior, to the Core System.
- The Field Node Owner/Operator receives CRLs from the Core System Facility.
- The Field Node Owner/Operator provides CRLs to the Mobile Device.

4.1.1.3.10 Local Policy Setting Entities

Local Policy Setting Entities establish policies, procedures, and regulations that may impact the configuration, operation and/or maintenance of the Core System. Entities filling this role could include local DoTs, local, county and state government, and regional transportation authorities.

Relationships:

- Local Policy Setting Entities provide IT policies, best practices, and policies related to user authorization to National Policy-Setting Entities.
- Local policies related to user authorization (i.e., who can do what) to the Core System Deployer
- Local laws and regulations to the Core System Manager.

4.1.1.3.11 Mobile Device

The Mobile Device is the device equipped with a wireless radio that enables it to communicate in the *connected vehicle* environment (e.g. 5.9GHz DSRC radio, cellular or other wide-area wireless radio). The Mobile Device requires security credentials to establish trust with other devices. The type of credentials vary with the type of device (e.g., 5.9 GHz devices require IEEE 1609.2 credentials), but the relationships required are the same.

Relationships:

- The Mobile Device receives an initial certification from the Mobile Device Manufacturer.
- The Mobile Device may request certificates from the ESS DSRC Registration Authority, which responds with IEEE 1609.2 certificates and CRLs.
- The Mobile Device may request certificates from the ESS X.509 Certificate Authority, which responds with an X.509 certificate.
- The Mobile Device provides misbehavior reports, describing activity of other System Users that it considers misbehavior, to the Core System.

4.1.1.3.12 Mobile Device Manufacturer

This is the manufacturer of a Mobile device. The Mobile Device Manufacturer may want his device certified so that prospective users of his device know that the device is independently certified to be compatible with the *connected vehicle* environment.

Relationships:

- The Mobile Device Manufacturer provides proof of device certification to the Mobile Device.

4.1.1.3.13 Mobile Device Operator

This is the operator of the Mobile device. If the device is mounted in a vehicle, the operator is the driver. If the device is handheld or carried, the operator is the user of the device.

Relationships:

- The Mobile Device Operator may enter into a usage agreement with the Mobile Device Owner, allowing the Operator to use the Mobile Device. This may require the Operator to provide some form of identification to the Owner.
- The Mobile Device Operator negotiates special user permission information with the Core System Manager. These permissions are based on needs of the Mobile Device Operator, local laws, regulations and policies.
- The Mobile Device receives CRLs from the Field Node Owner/Operator.

4.1.1.3.14 Mobile Device Owner

The Mobile Device Owner is the owner of the Mobile Device that participates in the *connected vehicle* environment.

Relationships:

- The owner of a DSRC device may enter into an agreement with the ESS DSRC Registration Authority for that ESS to obtain IEEE 1609.2 certificates for the device. The ESS DSRC Registration Authority provides certificates for the device.
- The owner may enter into an agreement with the ESS X.509 Certificate Authority for that ESS to provide X.509 certificates for the Mobile Device.
- The Mobile Device Owner may enter into a usage agreement with the Mobile Device Operator, allowing the Operator to use the Mobile Device. This may require the Operator to provide some form of identification to the Owner.

4.1.1.3.15 National Policy Setting Entities

National Policy Setting Entities establish policies, procedures, and regulations that may impact the configuration, operation, and/or maintenance of the Core System. Entities filling this role could include government agencies (e.g. the FCC), and trade groups (e.g. American Association of State Highway and Transportation Officials (AASHTO), IEEE).

Relationships:

- National Policy-Setting Entities receive IT policies, best practices, and policies related to user authorization from Local Policy Setting Entities.
- The National Policy Setting Entities provide recommended best practices to the Core System Deployer.
- The National Policy Setting Entities provide recommended best practices to the ESS DSRC Certificate Authority.
- The National Policy Setting Entities provide recommended best practices to the ESS X.509 Certificate Authority.

4.1.1.3.16 Standards Bodies

Standards bodies develop standards that are used to implement Core System interfaces.

Relationships:

- Standards Bodies make their standards available to the Core System Deployer, ESS DSRC Certificate Authority, and Core Certification Authority

4.1.1.3.17 USDOT Core System Documents

The library of documents that make up the USDOT's specification for the Core System: The ConOps, SyRS and SAD are included, but this library should also contain documents relevant to the certification of End User Applications.

Relationships:

- USDOT Core System Documents are provided to and used by the Core Certification Authority.

4.1.1.4 View Description

The Core System relies on external entities to provide the credentials necessary for Mobile Users to interact in the *connected vehicle* environment. For Mobile devices that wish to communicate using 5.9GHz DSRC, this means relying on two new entities, the ESS DSRC Certificate Authority, and the ESS DSRC Registration Authority for certificates. For Mobile Devices that use wireless communications provided by a third party communications provider (e.g. a cellular provider) certificates will be provided by the telecommunications provider (not shown on the diagram for clarity).

The Core System does provide digital (X.509) certificates to Field Nodes and Centers.

Cores enter into agreements with one another to exchange CRLs and misbehavior reports, and to establish the operational scope over which they provide certificates.

The provision of digital certificates for 5.9GHz DSRC is not performed by the Core System for two reasons:

1. Registration and certificate distribution functions must be separated in order to ensure that data provided by System Users is difficult to associate with the identity of the System User. This preservation of anonymity requires the use of a minimum of two, and likely at least four, tightly coordinated but separately managed entities. The exact configuration of entities required to grant DSRC certificates while preserving privacy is still being researched. Regardless of the configuration, this requires multiple entities to work closely together in a specified fashion. While this could be done by Cores operating in separate roles, these roles are distinct from the Core's other missions and need not be done more than once in any event.
2. Keeping the number of CAs and RAs small will simplify the management of the certificate distribution process. This does not refer to physical devices but to enterprises since there may be hundreds of physical CA devices. The inclusion of certificate distribution functions inside the Core is impractical since Cores are distinct and can be quite small in scope (e.g., the northern Virginia Core System), while the CA must be ubiquitously available across the United States.

Unfortunately, this leads to a scenario where a number of new entities must be created to provide and maintain the certificate distribution system. New entities are the ESS DSRC Certificate Authority, ESS DSRC Registration Authority, and Core Certification Authority.

The RA is the entity that interacts with the Mobile Device and its owner. The RA is the entity that may know the identity of the user. The CA provides the certificates to the RA, which then provides them to the device. The contents of the certificates are kept secure from the RA by encryption, so the RA cannot associate certificates with the end user. This maintains the owner's privacy.

The Core Certification Authority determines that the RA and CA are operating as they should. In effect it functions as a governing body for the *connected vehicle* environment (see Enterprise View – Governance).

The inclusion of these entities in the architecture must be considered a risk to the successful deployment and function of the *connected vehicle* environment, because none of these entities exist today and the parties responsible for establishing, operating and maintaining them are as yet undefined. The deployment risk can be managed by:

1. Establishing a clear plan, including responsible parties and timeframe, for development of the ESS RA and ESS CA.
2. Determining the enterprises responsible for operating and maintaining the ESS RA and ESS CA, documenting a business model for their continued solvency and developing a backup approach in case they fail.
3. Determining the makeup of the Core Certification Authority. As noted in Enterprise View - Governance, this is probably many different bodies, all of which need to be identified.

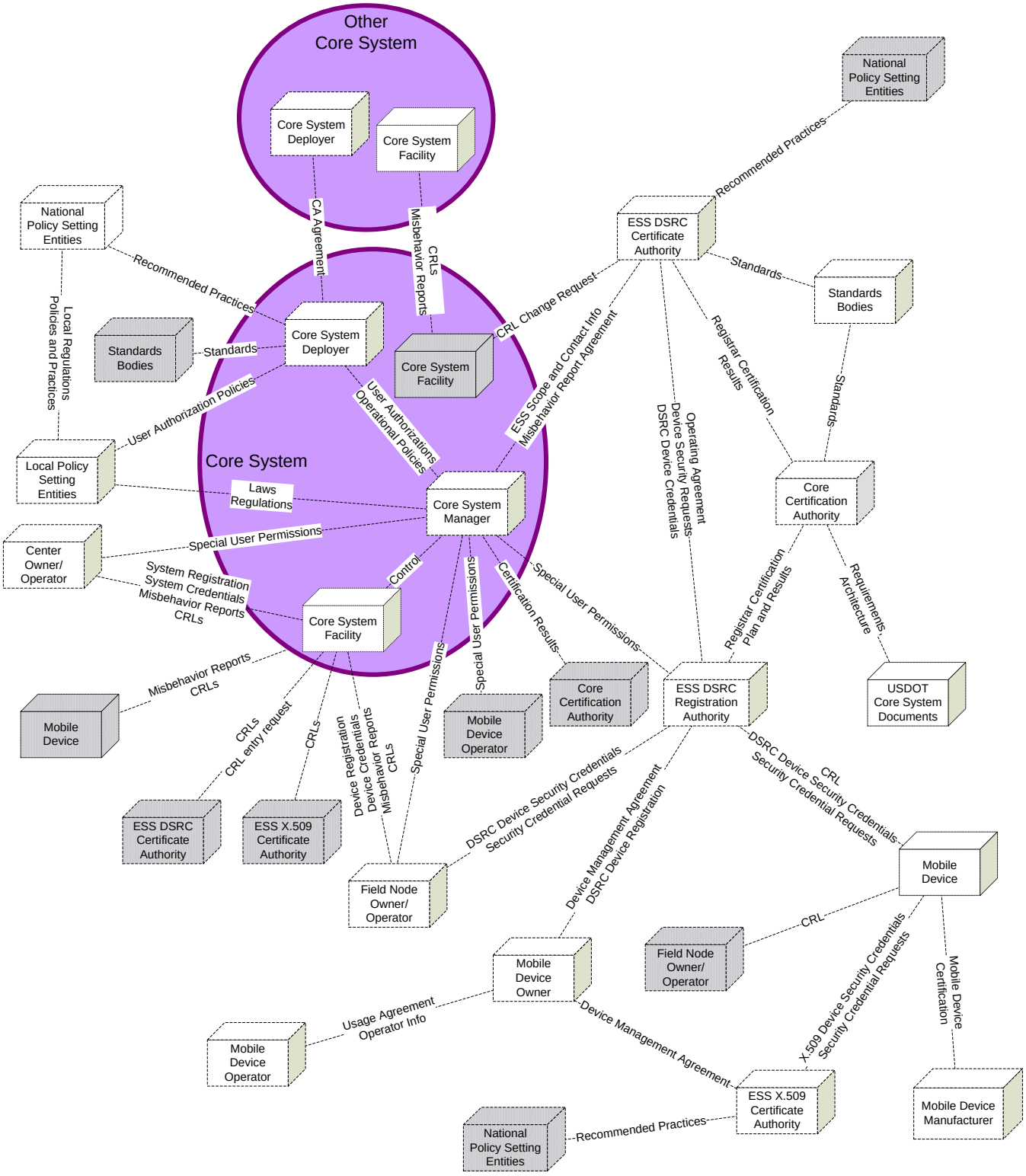


Figure 4-1: Enterprise View -- Security Credentials Distribution

4.1.1.5 Configuration Information

The following views must be considered when changing this view:

Functional Viewpoint: Functional View – Credentials Distribution.
Functional Viewpoint: Functional View – Misbehavior Management
Functional Viewpoint: Functional View – Core Backup

4.1.2 Enterprise View – Operations

4.1.2.1 Introduction

This view depicts the Enterprise Objects involved in data distribution, core2core interactions and Core service monitoring, focusing on the relationships between the Core System and external entities.

4.1.2.2 Concerns Addressed by this View

Organization/Resources	How do the entities responsible for a Core System need to interact with entities responsible for other Core Systems?
Risks	What relationships between Core Enterprise Objects and external Enterprise Objects provide risks to Core System development, deployment, operations, and maintenance? What steps can be taken to lessen risks that are a function of Enterprise relationships?

4.1.2.3 Object Definitions and Roles

4.1.2.3.1 Core Certification Authority

The Core Certification Authority verifies that a Core System and other connected vehicle devices meet specification set forth in the USDOT's documents and associated standards. For additional information on the Core Certification Authority see Enterprise View – Governance.

Relationships:

- The Core Certification Authority provides information about other Cores including Deployer contact information and Core operational scope to the Core System Deployer.

4.1.2.3.2 Core System Deployer

The Core System Deployer deploys the Core System.

Relationships:

- The Core System Deployer enters into a data coverage agreement with other Core System Deployers. This specifies the boundaries of each Core's data distribution services. At minimum these boundaries will include geographic boundaries and the sources that they will accept data from (e.g., all Mobile Devices, or transit vehicles only).
- The Core System Deployer enters into a certificate distribution agreement with other Core System Deployers. This specifies the boundaries of each Core's credential distribution services. While System Users may have permissions with any Core, Cores must coordinate the distribution of security credentials (certificates).
- The Core System Deployer enters into a takeover agreement with other Core Systems, specifying the services that each will provide for the other, and the performance expected in case of takeover.
- The Core System Deployer receives information about other Cores, including Deployer contact information and Core operational scope, from the Core Certification Authority.

4.1.2.3.3 Core System Facility

The Core System Facility is the hardware and software that provides Core services.

Relationships:

- The Core System Facility is controlled by the Core System Manager.
- The Core System Facility exchanges configuration information with other Core System Facilities. This information describes the services the Core provides and the boundaries over which it provides them.
- The Core System Facility informs other Core System Facilities when it detects a conflict in the boundaries of the two Cores. It receives conflict reports from other Core Facilities.
- The Core System Facility provides a takeover request to another Core when it requires a service to be operated by that Core. It receives takeover requests from other Core Facilities.
- The Core System Facility provides backup data necessary for another Core to provide services to the other Core System Facility. It receives backup data necessary to provide services on behalf of another Core.
- The Core System Facility provides performance data, describing service loading and reliability to other Core System Facilities. It receives similar information from other Core System Facilities.
- The Core System Facility receives data from the Mobile Device. It provides data according to subscription to the Mobile Device.
- The Core System Facility receives subscription requests from the Mobile Device.
- The Core System Facility provides performance data describing service status (up or down) to the Mobile Device.
- The Core System Facility receives geo-cast requests from the Mobile Device.
- The Core System Facility receives data from the Field Node. It provides data according to subscription to the Field Node.
- The Core System Facility receives subscription requests from the Field Node.
- The Core System Facility provides performance data describing service status (up or down) to the Field Node.
- The Core System Facility receives geo-cast requests from the Field Node.
- The Core System Facility receives data from the Center. It provides data according to subscription to the Center.
- The Core System Facility receives subscription requests from the Center.
- The Core System Facility provides performance data describing service status (up or down) to the Center.
- The Core System Facility receives geo-cast requests from the Center.

4.1.2.3.4 Core System Manager

The Core System Manager manages the operations of the Core System. This includes daily operations activities such as configuration changes, backups and other system administration tasks, and maintenance activities.

Relationships:

- The Core System Manager establishes procedures for takeover and return of services with other Core System Managers.

- The Core System Manager may initiate a takeover request manually, by contacting another Core System Manager.
- The Core System Manager controls the Core System Facility.

4.1.2.3.5 Center

The Center is the backoffice system that acquires and/or provides transportation-related data to/from the *connected vehicle environment.*

Relationships:

- The Center receives data from the Core System Facility. It provides data to the Core System Facility.
- The Center sends subscription requests to the Core System Facility.
- The Center receives Core System performance from the Core System Facility.
- The Center sends geo-cast requests to the Core System Facility.

4.1.2.3.6 Field Node

The Field Node is the roadside device equipped with a wireless radio that enables it to communicate in the *connected vehicle* environment (e.g., 5.9GHz DSRC radio, cellular or other wide-area wireless radio).

Relationships:

- The Field Node receives data from the Core System Facility. It provides data to the Core System Facility.
- The Field Node sends subscription requests to the Core System Facility.
- The Field Node receives Core System performance from the Core System Facility.
- The Field Node sends geo-cast requests to the Core System Facility.

4.1.2.3.7 Mobile Device

The Mobile Device is the device equipped with a wireless radio that enables it to communicate in the *connected vehicle* environment (e.g., 5.9GHz DSRC radio, cellular or other wide-area wireless radio)

Relationships:

- The Mobile Device receives data from the Core System Facility. It provides data to the Core System Facility.
- The Mobile Device sends subscription requests to the Core System Facility.
- The Mobile Device receives Core System performance from the Core System Facility.
- The Mobile Device sends geo-cast requests to the Core System Facility.

4.1.2.4 View Description

Core System Deployers enter into agreements with one another over data coverage, certificate distribution, and backup. Actual implementation of backup/takeover is delegated to the Core System Manager, which may operate the Core System Facility in such a way as to enable automatic takeover and backup operations.

The primary risk associated with Core System operations lie in the dependence on human interaction to establish agreement for operations between Cores. This means that Core System Deployers will have to be aware of other Core Systems. One approach to mitigating this risk would be to maintain a national

list of Cores, including some concept of their operational scope. This is logically a task for the Core Certification Authority.

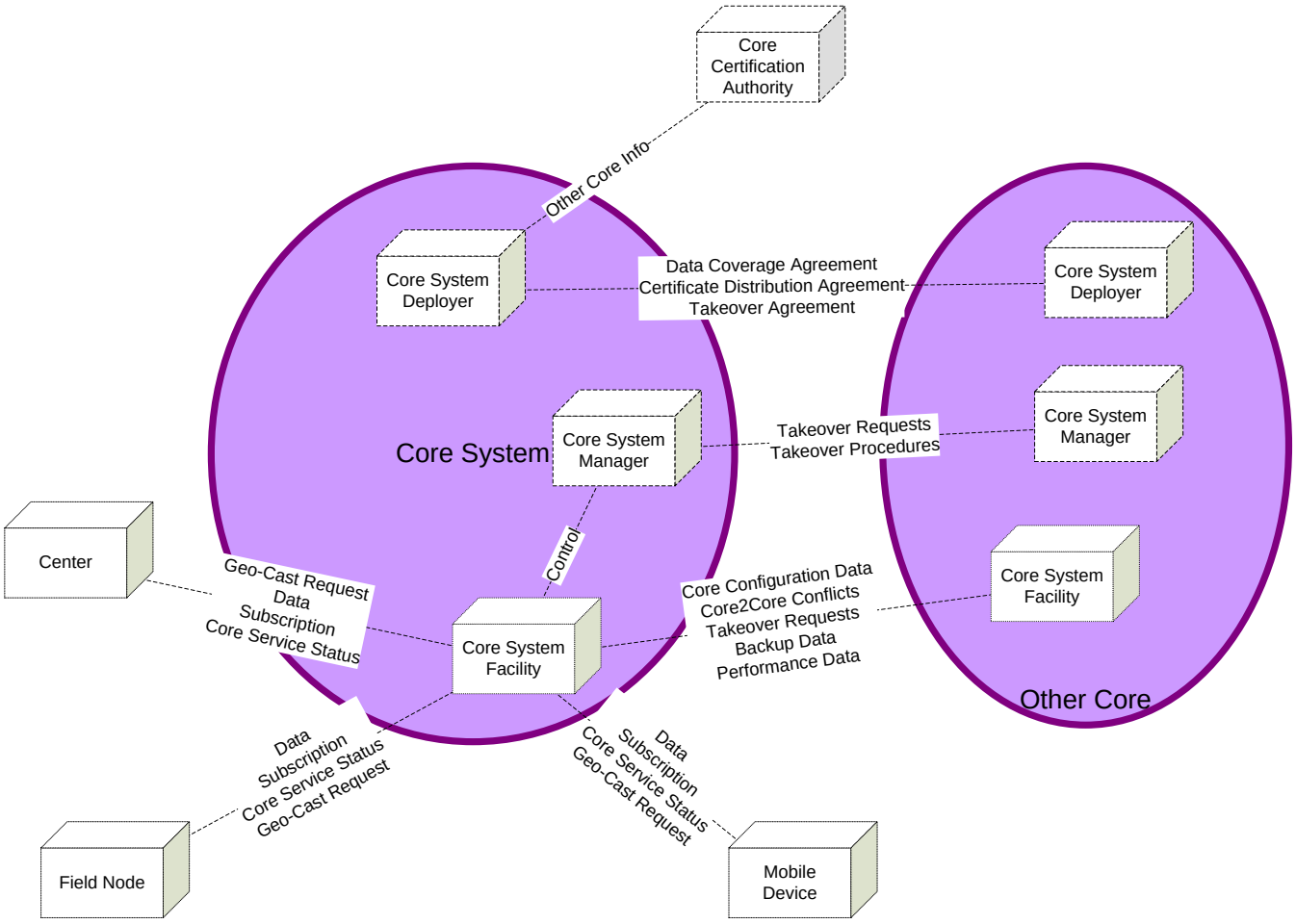


Figure 4-2: Enterprise View -- Operations

4.1.2.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Governance.
Functional Viewpoint: Functional View – Data Distribution.
Functional Viewpoint: Functional View – System Configuration.
Functional Viewpoint: Functional View – User Configuration.
Functional Viewpoint: Functional View – Core Backup.

4.1.3 Enterprise View – Core System and Application Development and Deployment

4.1.3.1 Introduction

The *connected vehicle* environment will require software development and distribution to implement Core Systems and also to provide Core-enabled software at end user devices. This view illustrates the relationships between entities involved in developing, testing, certifying, specifying, and deploying these various software components and applications. (Core System certification is covered under the Governance view.)

4.1.3.2 Concerns Addressed by this View

Organization/Resources	Who needs to contribute resources to Core System development, testing, transition, and operations? What resources are required to support Core System development, testing, transition, and operations? What Core System resources are required to support external application development? How do the entities responsible for a Core System need to interact with entities responsible for other Core Systems?
Risks	What relationships between Core Enterprise Objects and external Enterprise Objects provide risks to Core System development, deployment, operations, and maintenance? What steps can be taken to lessen risks that are a function of Enterprise relationships?
Deployability	Who needs to be involved with the transition from development to operations? What resources are required to support the transition from development to operations?

4.1.3.3 Object Definitions and Roles

4.1.3.3.1 Application Certification Body

The Application Certification Body certifies applications for deployment within the *connected vehicle* environment. This is likely to include several different bodies each focused on certification for a particular application area (e.g., safety, mobility, e-commerce).

Relationships:

- The Application Certification Body enters into an agreement with the End User Application Developer to perform certification testing on the End User Application Developer's application and to allow the End User Application Developer to advertise that his applications are so certified once they pass certification.

- The Application Certification Body also provides independent verification of certification to the End User Application Deployer.
- The Application Certification Body receives criteria for application certification from the USDOT Core System Documents and standards from Standards Bodies.
- The Application Certification Body receives standards from Standards Bodies.
- The Application Certification Body receives criteria for the certification of application from the Core Certification Authority.
- The Application Certification Body receives application performance information, including notification of application misbehavior, from the Core System Manager.
- The Application Certification Body informs the Application Registration Authority and End User Application Deployer of application certification and de-certification. A certification indicates that the application qualifies for PSID assignment, installation and use. Decertification is an indication that the application should no longer be used and should be de-installed.

4.1.3.3.2 Application Registration Authority

The Application Registration Authority is responsible for establishing, distributing, and maintaining the application identifiers (Provider Service Identifiers (PSIDs)) that are used by applications.

Relationships:

- The Application Registration Authority works with the Core System Manager to establish and/or provide PSIDs for applications to be deployed in the area of the Core.
- The Application Registration Authority provides a PSID assignment to the End User Application Developer.
- The Application Registration Authority receives certification and decertification information from the Application Certification Body. A certification indicates that the application qualifies for PSID assignment, installation and use. Decertification is an indication that the application should no longer be used and should be de-installed.

4.1.3.3.3 Core Certification Authority

The Core Certification Authority verifies that a Core System and other connected vehicle devices meet specification set forth in the USDOT's documents and associated standards. For additional information on the Core Certification Authority see Enterprise View – Governance.

Relationships:

- The Core Certification Authority receives the Core System requirements and architecture from the USDOT Core System Documents.
- The Core provides criteria for the certification of applications to the Application Certification Body.
- The Core Certification Authority provides the initial certification that the Core meets the requirements of USDOT Core System Documents. It provides this certification to the Core System Developer.

4.1.3.3.4 Core Library Developer

The Core System Developer develops library software for use on System User devices.

Relationships:

- The Core Library Developer receives specifications for library software from the Core System Developer
- The Core Library Developer receives behavioral and structural requirements from the standards provided by Standards Bodies.
- The Core Library Developer enters into a contract or agreement with the End User System Deployer, allowing the End User System Deployer to use library software.

4.1.3.3.5 Core System Deployer

The Core System Deployer deploys the Core System.

Relationships:

- The Core System Deployer acquires the tested and validated system from the Core System Developer, for which it may enter into a funding arrangement and provides constraints on system capabilities for consideration during development to reflect the Core System Deployer's needs.
- The Core System Deployer receives locally imposed constraints on system deployment and operations that may affect the Core System it wishes to deploy from Local Policy Setting Entities.
- The Core System Deployer may require some or all End Users to enter into a Core System Usage Agreement in order to use Core services.
- The Core System Deployer may enter into a coordination agreement with another Core System Developer. This has a larger scope than the agreements covered in the Operations View; this coordination agreement deals with changes in scope over time and may include discussion of compensation between Cores. While in practice the Operations agreements and the coordination agreement may be negotiated together and/or even be parts of the same document, they are different in scope and so identified separately.

4.1.3.3.6 Core System Developer

The Core System Developer develops the Core System software and specifies its hardware configuration. It develops the system based on the USDOT Core System Documentation, related standards, and any special requirements or constraints imposed by the Core System Deployer.

Relationships:

- The Core System Developer receives requirements and architecture from this SAD and the SyRS.
- The Core System Developer receives standards from Standards Bodies
- The Core System Developer enters into an agreement to produce the Core System, and receives additional system constraints and acceptance criteria from the Core System Deployer.
- The Core System Developer works with the Core System Tester to ensure that it has developed the software according to the requirements and architecture specified in these source documents.
- The Core System Developer may exchange library software specifications with the Core Library Developer.
- The Core System Developer receives the initial certification that the Core meets the requirements of USDOT Core System Documents from the Core Certification Authority.

4.1.3.3.7 Core System Manager

The Core System Manager manages the operations of the Core System. This includes daily operations activities such as configuration changes, backups and other system administration tasks, and maintenance activities.

Relationships:

- The Core System Manager provides lessons learned from operations and its proposed changes to the documentation back into the USDOT Core System Documentation.
- The Core System Manager enters into an agreement with the End User Application Deployer to facilitate the application requirements as they pertain to the Core, in particular the establishment of user accounts and permissions for End Users and/or the Application Deployer.
- The Core System Manager works with the Application Registration Authority to establish and/or receive PSIDs for applications to be deployed in the area of the Core.
- The Core System Manager receives legal and regulatory information from Local Policy Setting Entities.
- The Core System Manager provides application performance information, including notification of application misbehavior, to the Application Certification Body.

4.1.3.3.8 Core System Tester

The Core System Tester performs verification and validation of the Core System Developer's software.

Relationships:

- The Core System Tester uses USDOT Core System Documentation as foundation for establishing Core System test plans and procedures.
- The Core System Tester works with the Core System Developer to certify that the Core System meets the requirements and architecture specified in the Core System Documentation.

4.1.3.3.9 End User

The End User is the Center, Field, or Mobile User human operator whose software interacts with the Core System.

Relationships:

- This End User may enter into license agreements with the End User Application Deployer to use the latter's application.
- This End User may enter into license agreements with the End User System Deployer to use the device provided by the End User System Deployer.
- This End User may enter into a Core System License agreement with the Core System Deployer for the End User to use the Core System.
- The End User receives de-installation notification from the End User Application Deployer in the case of an application being decertified.

4.1.3.3.10 End User Application Deployer

The End User Application Deployer deploys applications onto End User devices. For Field and Center Users this would typically be a specialized contractor or software licensing organization. For Mobile Users this is more likely to be a service provider (e.g. a cellular service provider, automobile dealer).

Relationships:

- The End User Application Deployer enters into an agreement with the End User Application Developer to deploy the application the End User Application Developer develops.
- The End User Application Deployer receives application certification information from the Application Certification Body.
- The End User Application Deployer enters into a deployment agreement with the Core System Manager, which includes specification of End User permissions required to use the application in question (if any), and establishes Core System user accounts for End Users and the End User Application Deployer, if required by the application.
- The End User Application Deployer may require the End User to enter into a license agreement to use the application.
- The End User Application Deployer receives local regulations, policies and practices from Local Policy Setting Entities.
- The End User Application Deployer receives decertification information from the Application Certification Body.
- The End User Application Deployer provides de-installation notification to the End User in the case of an application being decertified.

4.1.3.3.11 End User Application Developer

The End User Application Developer develops applications targeted to End User devices and systems.

Relationships:

- The End User Application Developer receives requirements and constraints on interaction with the Core System from the USDOT Core System Documents
- The End User Application Developer receives standards from Standards Bodies.
- The End User Application Developer enters into an agreement with the Application Certification Body for the latter to certify that an application is compliant with the Core System Documents.
- The End User Application Developer enters into an agreement with the End User Application Deployer for the latter to deploy the application the End User Application Developer develops.
- The End User Application Developer receives recommended practice information from National Policy Setting Entities.
- The End User Application Developer receives a PSID assignment from the Application Registration Authority.

4.1.3.3.12 End User System Deployer

The End User System Deployer provides the physical devices and device configuration with which End Users interact. This could include an automotive manufacturer for OEM-supplied devices, a dealer or aftermarket shop for retrofit devices, a device manufacturer for aftermarket devices, a cellular service provider for smart phones, or a traffic signal controller manufacturer for DSRC-enabled signal controllers, etc.

Relationships:

- The End User System Deployer may enter into a licensing agreement with the Core Library Developer to use library components developed by the Core Library Developer.
- The End User System Deployer may require the End User to enter into a license agreement before using the device the End User System Deployer provides.

4.1.3.3.13 Local Policy Setting Entities

Local Policy Setting Entities establish policies, procedures and regulations that may impact the configuration, operation and/or maintenance of the Core System. Entities filling this role could include local DoTs, local, county and state government, regional transportation authorities.

Relationships:

- Local Policy Setting Entities provide the legal and regulatory constraints that the Core System Deployer must consider prior to deployment.
- Local Policy Setting Entities provide updates to laws and regulations that the Core System Manager must consider during operations.
- Local Policy Setting Entities provide local regulations, policies and practices to the End User Application Deployer.
- Local Policy Setting Entities provide local regulations, policies and practices to National Policy Setting Entities, and receive national regulations policies and practices from National Policy Setting Entities.

4.1.3.3.14 National Policy Setting Entities

National Policy Setting Entities establish policies, procedures and regulations that may impact the configuration, operation and/or maintenance of the Core System and *connected vehicle* applications. Entities filling this role could include federal agencies such as the USDOT and FCC, trade groups such as IEEE or AASHTO.

Relationships:

- The National Policy Setting Entities provides recommended practice information to the End User Application Developer.
- National Policy Setting Entities receive local regulations, policies and practices from Local Policy Setting Entities, and provide national regulations policies and practices to Local Policy Setting Entities.

4.1.3.3.15 Standards Bodies

Standards bodies develop standards that are used to implement Core System interfaces.

Relationships:

- Standards Bodies enter into a development contract with a separate funding agency.
- Standards Bodies make their standards available to the Core System Developer, Core Library Developer, End User Application Developer and Application Certification Body.

4.1.3.3.16 Standards Funding Agencies

Standards Funding Agencies are the entities that provide funding and direction to Standards Bodies. This could be the USDOT or any combination of private and public agencies under a separate agreement.

Relationships:

- Standards Funding Agencies identify required standards from the USDOT Core System Documents.
- Standards Funding Agencies enter into a development contract with a Standards Body to produce standards.

4.1.3.3.17 USDOT Core System Documents

The library of documents that make up the USDOT's specification for the Core System: The ConOps, SyRS and SAD are included, but this library should also contain documents relevant to the certification of End User Applications.

Relationships:

- USDOT Core System Documents are provided to and used by the Core Certification Authority, End User Application Developer, Core System Manager, Core System Tester, Core System Developer, and Standards Funding Agencies.

4.1.3.4 View Description

This view addresses the enterprise relationships involved in the development and deployment of the Core System, and also the development and deployment of external applications.

There are five types of enterprises involved in this view: developers, deployers, end users, specification entities, and support entities.

- **Developers** create and update end user applications, Core System software, and Core Library software (Core Library software is that software used by System Users to access Core System functions. This is software resident on System User devices that interfaces with or uses data provided by the Core).
- **Deployers** install applications, library software, and Core software in operational systems.
- **End Users** are the Mobile, Center, and Field Users that require the use of library and/or end user software
- **Specification entities** provide requirements or constraints used by deployers or developers.
- **Support entities** provide testing, funding, or certification of products produced by deployers.

This Enterprise View illustrates various relationships, mostly agreements or contracts, between Enterprises that are necessary to deploy functionality to Cores or End Users. External Application Developers and Deployers are included because they may have a relationship with the Core Manager that enables provision of applications to End Users. For example the developer of a signal priority application may need the Core to ensure that the End Users of its application have the relevant special user permissions, which are encoded as part of their DSRC certificate. This requires the Deployer of the signal priority application to work with the Core System Manager to ensure those certificates include the requisite permissions. Both the Deployer and the Core System Manager need to work with the Application Registration Authority to ensure that the DSRC-specific details, including PSID assignment, are properly handled.

Additionally certification of external applications is based on the foundational documents describing the Core System (ConOps, SyRS, and SAD).

The Application Certification Body uses standards to evaluate and certify applications; certification processes could be further affected by operational lessons learned (lessons learned come from the Core System Manager). Not all Cores have individual relationships with the Application Certifying Body; therefore lessons learned should be submitted by the Core System Manager and applied formally as notes or changes to the Core Requirements or Architecture in the USDOT Core System Documents. The resulting changes would be recognized as part of the application certification process.

This view also addresses the transitions between development and deployment. The Core Certification Authority, Core System Tester, Core System Manager, Core System Deployer, Core System Developer, and Core Library Developer may all be involved in the transition phase. Since the Core Certification Authority's membership is as yet undefined, its membership and workings represent a significant risk to successful deployment. Swift definition of this body's membership and operations would help ameliorate this risk.

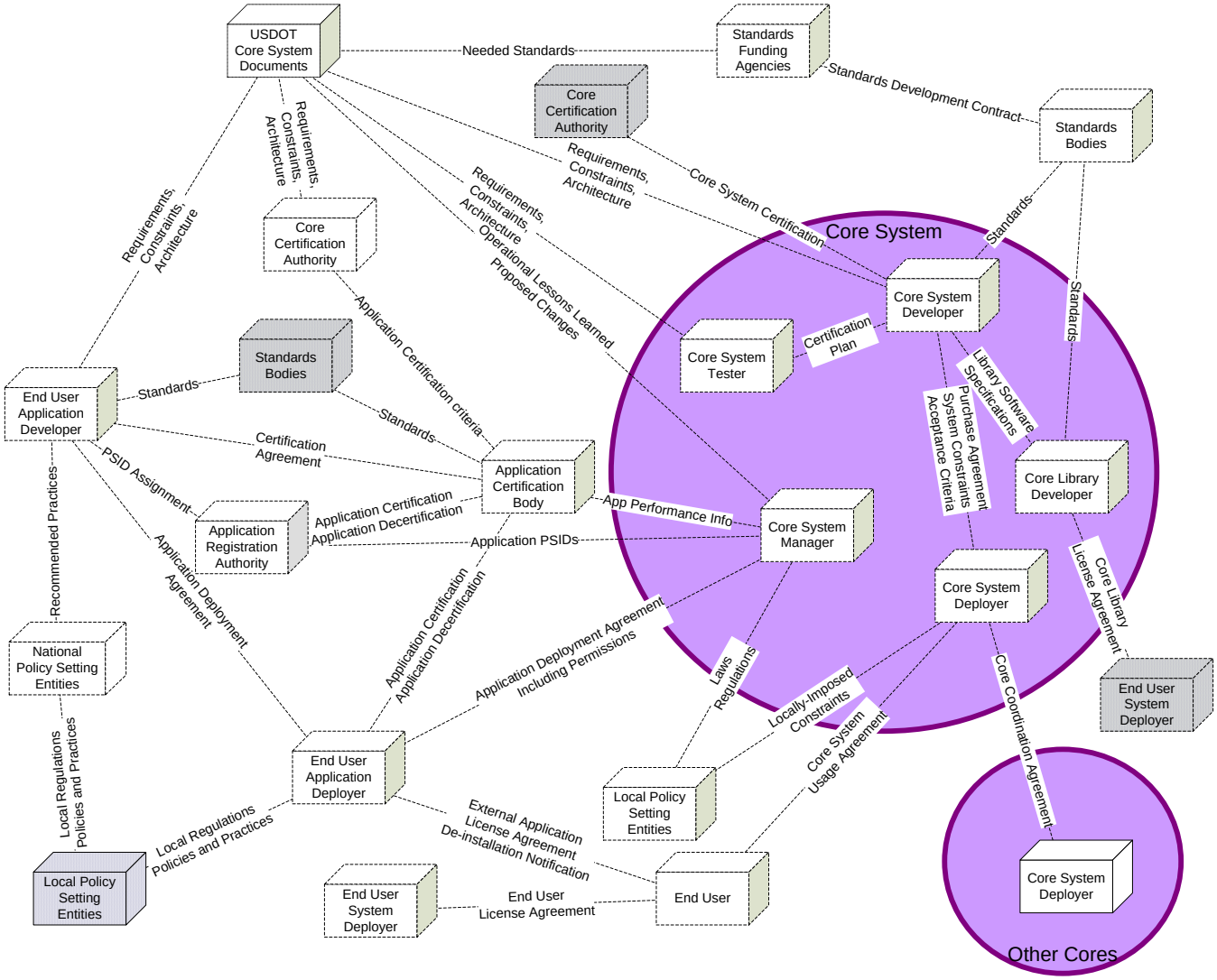


Figure 4-3: Enterprise View -- Development

4.1.3.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Security Credentials Distribution
Enterprise Viewpoint: Enterprise View – Configuration and Maintenance
Enterprise Viewpoint: Enterprise View – Governance

4.1.4 Enterprise View – Configuration and Maintenance

4.1.4.1 Introduction

The Core System will require maintenance for security and application software patches, configuration changes, hardware maintenance, and similar activities. This view explores the enterprise relationships necessary to perform those maintenance activities and maintain the operational configuration between the Core and external entities dependent on it.

4.1.4.2 Concerns Addressed by this View

Organization/Resources	Who needs to contribute resources to Core System development, testing, transition, and operations? What resources are required to support Core System development, testing, transition, and operations?
Risks	What relationships between Core Enterprise Objects and external Enterprise Objects provide risks to Core System development, deployment, operations, and maintenance? What steps can be taken to lessen risks that are a function of Enterprise relationships?
Maintainability	Who needs to contribute resources to Core System maintenance activities? What resources are required to support Core System maintenance? What interactions between Enterprise Objects are required to support maintenance activities while maintaining Core operations?

4.1.4.3 Object Definitions and Roles

4.1.4.3.1 Center Owner/Operator

The Center Owner/Operator is the owner and operator of the Center that interfaces with the Core System. For example this could be an information service provider or traffic management center. This object also includes the Center facility which is not drawn separately to help keep the diagram readable.

Relationships:

- If the Center connects to the Core by a private network (see Connectivity View – High Level) then the Center Owner/Operator will enter into an agreement with the Core System Manager where the Center Owner/Operator provides a communications link to the Core, and the Core commits to providing routing between the Center User and other System Users.
- If the Center connects to the Core by a private network (see Connectivity View – High Level) then the Center Owner/Operator will enter into an agreement with the Core System Manager where the Center Owner/Operator commits to maintaining the communications link to the Core, and the Core commits to maintaining the routing services between the Center User and other System Users.

4.1.4.3.2 Core Certification Authority

The Core Certification Authority verifies that a Core System and other connected vehicle devices meet specification set forth in the USDOT's documents and associated standards. For additional information on the Core Certification Authority see Enterprise View – Governance.

Relationships:

- The Core Certification Authority receives Core System performance report data from the Core System Facility.

4.1.4.3.3 Core System Deployer

The Core System Deployer deploys the Core System.

Relationships:

- The Core System Deployer acquires this system from the Core System Developer, for which it may enter into a funding arrangement, and provides constraints on system capabilities depending on the Core System Deployer's scope.
- The Core System Deployer receives IT maintenance policies and best practices from Local Policy Setting Entities.
- The Core System Deployer receives recommended practices for maintenance procedures from National Policy Setting Entities.
- The Core System Deployer then enters into a contractual relationship with the Core System Manager to operate the system. This contract should include IT maintenance policies and best practices received from Local Policy Setting Entities.

4.1.4.3.4 Core System Developer

The Core System Developer develops the Core System software and specifies its hardware configuration.

Relationships:

- The Core System Developer provides its hardware configuration and software to a Core System Deployer, and may modify its software to take into account the Core System Deployer's supplied constraints.
- The Core System Developer trains the Core System Manager in the operation of the system.
- The Core System Developer trains the Core System Maintainer in the maintenance of the system.
- The Core System Developer receives maintenance concepts from the USDOT Core System Documents.

4.1.4.3.5 Core System Facility

The Core System Facility is the hardware and software that provides Core services.

Relationships:

- The Core System Facility provides performance reports to the Core Certification Authority.

4.1.4.3.6 Core System Maintainer

The Core System Maintainer maintains the Core System's operational configuration, patches its software and maintains the Core's physical environment.

Relationships:

- The Core System Maintainer operates under a contract with the Core System Manager. It coordinates maintenance activities with the Core System Manager.
- The Core System Maintainer receives training on how to maintain the Core from the Core System Developer.
- The Core System Maintainer receives Test Results of the original system configuration from the Core System Tester.
- The Core System Maintainer should enter into an arrangement with the Field Node Owner/Operators to exchange maintenance information that is relevant to each. If Field Nodes are connected to the Core by a private network (see Connectivity View – High Level) then this agreement will also include the Field Node Owner/Operator's commitment to maintaining the communications links between Field Nodes and the Core, and the Core's commitment to maintaining routing capabilities between Field and other System Users.
- The Core System Maintainer receives maintenance standards from Standards Bodies.
- If the Center connects to the Core by a private network (see Connectivity View – High Level) then the Center Owner/Operator will enter into an agreement with the Core System Manager where the Center Owner/Operator commits to maintaining the communications link to the Core, and the Core's commits to maintaining the routing services between the Center User and other System Users.

4.1.4.3.7 Core System Manager

The Core System Manager manages the operations of the Core System. This includes daily operations activities such as configuration changes, backups and other system administration tasks, and maintenance activities.

Relationships:

- The Core System Manager enters into a contract with the Core System Deployer for the Core System Manager to operate the Core System.
- The Core System Manager enters into a contract with the Core System Maintainer for the Maintainer to maintain the Core, within the constraints of and using the best practices and IT maintenance policies stipulated by the Core System Manager's Operations Contract.
- The Core System Manager coordinates operations and maintenance activities with the Core System Maintainer.
- The Core System Manager receives training in Core operations from the Core System Developer.
- The Core System Manager receives updates to laws and regulations that may impact maintenance activities from Local Policy Setting Entities.
- The Core System Manager enters into an agreement with the Field Node Owner/Operator to share operations and maintenance information, so that each enterprise can take maintenance and failure outages of the other's systems into account. If Field Nodes are connected to the Core by a private network (see Connectivity View – High Level) then this agreement will also include the Field Node Owner/Operator's commitment to providing the communications between Field Node and Core, and the Core's commitment to providing routing between Field Node and other System Users.
- If the Center connects to the Core by a private network (see Connectivity View – High Level) then the Center Owner/Operator will enter into an agreement with the Core System Manager where the Center Owner/Operator provides a communications link to the

Core, and the Core's commits to providing routing between the Center User and other System Users.

4.1.4.3.8 Core System Tester

The Core System Tester verifies and validates that the Core System meets the requirements and needs set forth in the SyRS and ConOps.

Relationships:

- The Core System Tester provides the test results from the original system deployment to the Core System Maintainer, for the Core System Maintainer to use as a baseline for system performance.

4.1.4.3.9 Field Node Owner/Operator

The Field Node Owner/Operator is the owner and operator of the field infrastructure used by the DSRC Device Owner to interface with the Core System. This could be a cellular provider or DSRC field infrastructure manager.

Relationships:

- The Field Node Owner/Operator enters into an agreement with the Core System Manager to share operations and maintenance information, so that each enterprise can take maintenance and failure outages of the other's systems into account. If Field Nodes are connected to the Core by a private network (see Connectivity View – High Level) then this agreement will also include the Field Node Owner/Operator's commitment to providing the communications link between Field Node and Core, and the Core's commitment to providing routing between Field Node and other System Users.
- The Field Node Owner/Operator enters into an agreement with the Core System Maintainer to share operations and maintenance information, so that each enterprise can take maintenance and failure outages of the other's systems into account. If Field Nodes are connected to the Core by a private network (see Connectivity View – High Level) then this agreement will also include the Field Node Owner/Operator's commitment to maintaining the communications links between Field Nodes and the Core, and the Core's commitment to maintaining routing capabilities between Field and other System Users.

4.1.4.3.10 Local Policy Setting Entities

Local Policy Setting Entities establish policies, procedures and regulations that may impact the configuration, operation and/or maintenance of the Core System. Entities filling this role could include local DoTs, local, county and state government, regional transportation authorities, and in particular the IT interests of those agencies.

Relationships:

- The Local Policy Setting Entities provide IT maintenance policies and best practices to the Core System Deployer.
- Local Policy Setting Entities provide updates to laws and regulations that may impact maintenance activities of the Core System to the Core System Deployer according to local policies and practices.
- Local Policy Setting Entities provide local regulations, policies and practices to National Policy Setting Entities, and receive national regulations policies and practices from National Policy Setting Entities.

4.1.4.3.11 National Policy Setting Entities

National Policy Setting Entities establish policies, procedures and regulations that may impact the configuration, operation, and/or maintenance of the Core System and *connected vehicle* applications. Entities filling this role could include federal agencies such as the USDOT and FCC, trade groups such as IEEE or AASHTO.

Relationships:

- The National Policy Setting Entities provides recommended practice information to the Core System Deployer.
- National Policy Setting Entities receive local regulations, policies and practices from Local Policy Setting Entities, and provide national regulations policies and practices to Local Policy Setting Entities.

4.1.4.3.12 Standards Bodies

Standards bodies develop standards that are used to implement Core System interfaces.

Relationships:

- Standards Bodies make maintenance standards available to the Core System Maintainer.

4.1.4.3.13 USDOT Core System Documents

The library of documents that make up the USDOT's specification for the Core System: The ConOps, SyRS and SAD are included, but this library should also contain documents relevant to the certification of End User Applications.

Relationships:

- USDOT Core System Documents related to Core System Maintenance are provided to and used by the Core System Developer.

4.1.4.4 View Description

This view addresses the enterprise relationships involved in the operations and maintenance of the Core System.

Policy Setters may impose their desires through law, funding incentive, or political pressure. This will vary from jurisdiction to jurisdiction depending upon the nature of the organizations involved. Maintaining consistency among these relationships will be difficult as the number and interests of organizations filling this role could be diverse. This is a risk item and needs to be recognized by potential deployers. Without consistent approaches to IT management, Deployers (and in turn Managers and Maintainers) could spend significant resources accommodating these policies or even choosing not to deploy because of the difficulty in doing so.

There are two dissimilar Field Node Owner/Operator types: the local deployer of wireless infrastructure (chiefly DSRC, but this could be municipal Wi-Fi) and cellular network providers. The DSRC infrastructure owner/operator is likely to be tightly coupled to the Core System effort and sympathetic to the needs of the Core operators; DSRC infrastructure is focused on delivering Core services. The cellular provider will generally be a commercial company interested primarily in providing data to an increasing number of subscribers. Establishing a working relationship between maintenance personnel at these corporations will be more problematic and should be noted as another risk item. Without this relationship, Core System Maintainers may not understand the communications environment they operate in, and thus encounter difficulty monitoring and debugging system performance issues.

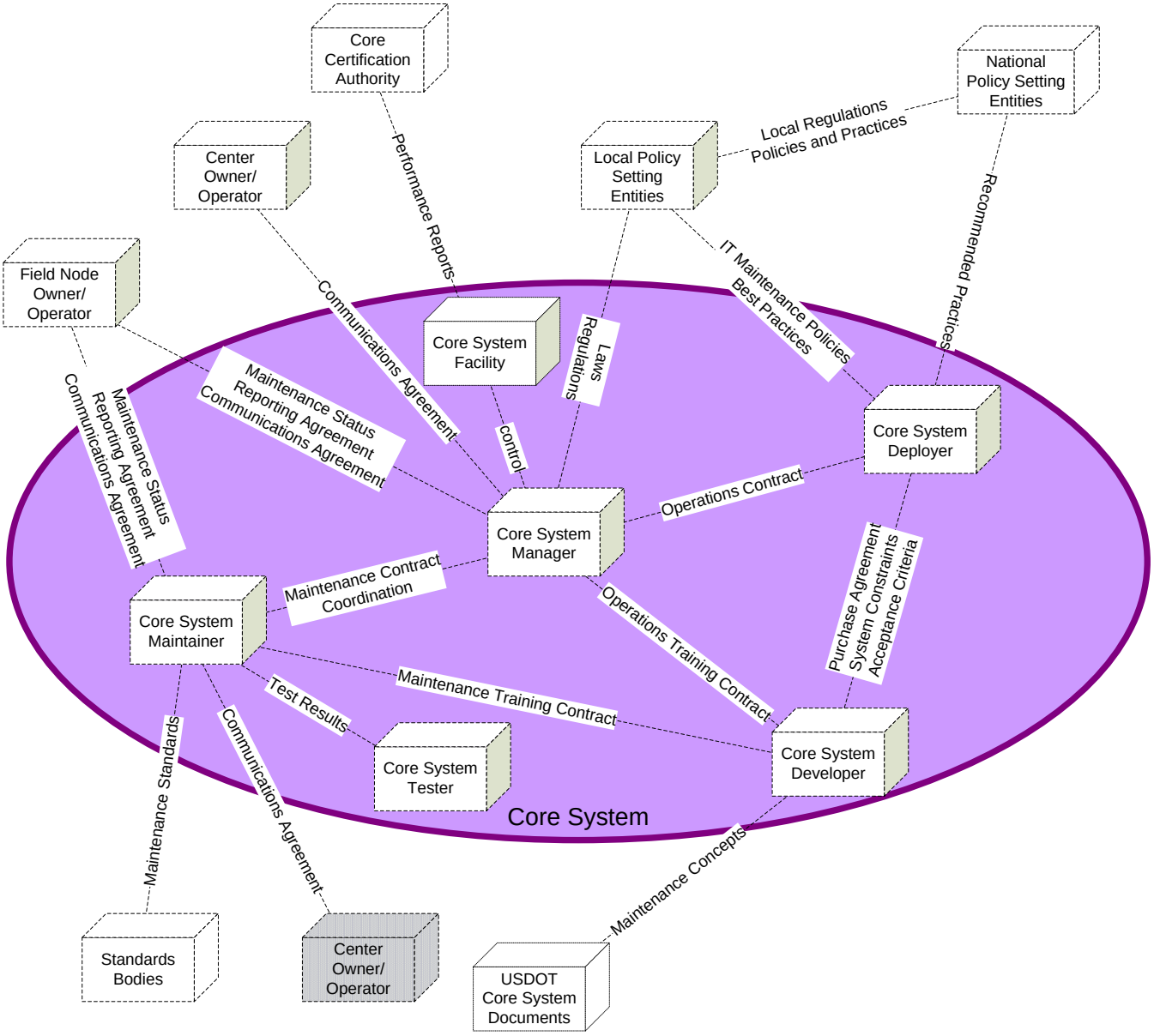


Figure 4-4: Enterprise View – Operations and Maintenance

4.1.4.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Governance
Functional Viewpoint: Functional View – System Configuration
Functional Viewpoint: Functional View – User Configuration
Functional Viewpoint: Functional View – System Monitor and Control

4.1.5 Enterprise View – Governance

4.1.5.1 Introduction

The Core System Concept of Operations established the concept of multiple Cores, allowing overlap and interaction between Cores to allow flexibility in deployments. The ConOps did not discuss however, how those Cores would be governed. This view is the first exploration of governance of multiple Cores.

As defined in other views, Core Systems may interact to provide backup services to one another and to coordinate boundaries of operation. But who decides what makes a Core a Core? What are the ramifications of a Core acting outside of the parameters established for Core operation by this SAD and the SyRS? This view answers that question by drawing on a concept from the VII program, updated for today's environment.

During VII there was a concept for a "VII Operating Entity" that would oversee the operation of the national system, and verify components and interfacing systems. This entity would include representatives of all stakeholders, and have substantial authority when it came to operational policies.

The Core System is not VII. It may be limited in scope, and may have many other Cores to work or contend with. Unlike VII's national system, the Core does not grant certificates to mobile participants in the environment. Nor does it include applications. Consequently the job of governing the Core is substantially different. It is at once simpler, because so many parts of the *connected vehicle* environment are now outside the Core, and more complex, because interactions between those parts must be continuously verified to ensure the security and safety of the environment. The Core Certification Authority (defined below) takes some of the concepts of the VII Operating Entity but focuses on certification rather than operations.

A complete governance model is well beyond the scope of this document, but the elementary relationships that must exist for governance impact the Functional and Information Viewpoints. Such relationships are thus drawn to aid in the development of the architecture; however, identification of those relationships may assist in future work defining the governance model for the *connected vehicle* environment.

4.1.5.2 Concerns Addressed by this View

Organization/Resources	Who needs to contribute resources to Core System development, testing, transition, and operations? What resources are required to support Core System development, testing, transition, and operations?
Risks	What relationships between Core Enterprise Objects and external Enterprise Objects provide risks to Core System development, deployment, operations, and maintenance? What steps can be taken to lessen risks that are a function of Enterprise relationships?

Evolvability	How do the relationships between Enterprise Objects need to change to support the integration of new Enterprises? Specifically, what is the decision mechanism for integrating new Enterprises and modifying roles of existing Enterprises?
Deployability	Who needs to be involved with the transition from development to operations?

4.1.5.3 Object Definitions and Roles

4.1.5.3.1 Core Certification Authority

The Core Certification Authority is a group of stakeholders that verifies that a Core System and other *connected vehicle* devices meet specifications set forth in the USDOT's documents and associated standards.

The Core Certification Authority creates a Core Certification Plan, which identifies what must be done to certify that a Core meets the requirements and architecture defined in the USDOT Core System Documents (SyRS and SAD). At a minimum, this plan identifies the optional/desirable Core System requirements that the Core is expected to meet, the services it will include and their associated service areas, and projected Core2Core relationships. This plan is provided to the Core System Deployer prior to certification, and once certification takes place the results of that certification are also provided to the Deployer and Core System Developer. Also, the Core Certification Authority provides the Core System Manager with the Core's X.509 Root Certificate.

The makeup of the Core Certification Authority is an unresolved policy question. Based on considerations from the VII program, the makeup of this body could include representatives of the following stakeholder groups:

- State/local DOTs
- USDOT including the Research and Innovative Technology Administration (RITA) Intelligent Transportation System (ITS) Joint Program Office (JPO), Federal Highway Administration (FHWA), Federal Transit Administration (FTA), National Highway Traffic Safety Administration (NHTSA), and the Office of the Secretary of Transportation's (OST) Office of Policy
- Automakers
- Equipment vendors
- Mobile User telecommunications providers

This appears to be a rather large group with many certification responsibilities. However, instead of a single entity the Core Certification Authority is likely a collection of enterprises, each of which provides certification for a particular subset of the devices and systems that participate in the *connected vehicle* environment. The relationships between these enterprises, including the delineation of roles and responsibilities and any hierarchical or peer-to-peer relationships is beyond the scope of this SAD.

Relationships:

- The Core Certification Authority receives the Core System requirements and architecture from the USDOT Core System Documents.
- The Core Certification Authority receives applicable standards from Standards Bodies.

- The Core Certification Authority works with the Core System Deployer to create the Core's Certification Plan. The Core System Deployer needs to provide the Core's configuration, documenting the services it will implement, the scope over which the services will be provided (including area, System User types and any time restrictions), and proposed relationships with other Cores.
- Once the Core is certified but prior to the Core beginning operations, the Core Certification Authority provides a Core Monitoring Plan to the Core System Manager. Upon receipt of the Plan the Core is permitted to operate. Periodic re-certification of the Core by this body will occur as specified in that Core Monitoring Plan. Should a Core fail recertification, the Core Certification Authority may penalize the Core; this may result in the Core's root certificate being revoked, in which case operation of the federated system of Cores will be impacted.
- The Core Certification Authority may provide similar certification functions for Field and Mobile devices that operate in the *connected vehicle* environment. This includes establishing certification and monitoring plans for each device, managed through relationships with the device manufacturers and operators.
- The Core Certification Authority may certify that an ESS meets the requirements necessary to provide service on behalf of the Core System, and enter into a periodic monitoring relationship with that ESS. In particular, the ESS DSRC CA and ESS DSRC RA would require this sort of relationship.
- The Core Certification Authority receives local regulations concerning periodic certification of Mobile Devices from Local Policy Setting Entities,
- The Core Certification Authority receives national regulations regarding wireless spectrum use and management from National Policy Setting Entities.
- The Core Certification Authority provides certifications for External Support Systems to the Core System Manager.

4.1.5.3.2 Core System Deployer

The Core System Deployer deploys the Core System.

Relationships:

- The Core System Deployer receives Core operating local operational policies and regulations from Local Policy Setting Entities.
- The Core System Deployer provides Core operating policies, constraints and configuration information to the Core System Manager, based on input received from Local Policy Setting Entities. This includes:
  - The types of data to distribute as part of data distribution
  - Who can subscribe to data, and what relationship that subscriber must establish with the Core in order to subscribe to data
  - Who can publish data using the Core, and what relationship that publisher must establish with the Core in order to publish data
- The Deployer works with the Core Certification Authority to establish the Core Certification Plan, and receives the results of initial and subsequent certifications.
- The Core System Deployer also establishes usage agreements with Field Node Owner/Operators that provide services within the Core's area of operations.

4.1.5.3.3 Core System Developer

The Core System Developer deploys the Core System.

Relationships:

- The Core System Developer provides Core operating procedures to the Core System Manager.
- The Core System Developer receives initial Core certification from the Core Certification Authority.

4.1.5.3.4 Core System Manager

The Core System Manager manages the operations of the Core System. This includes daily operations activities such as configuration changes, backups and other system administration tasks, and maintenance activities.

Relationships:

- The Core System Manager receives initial and subsequent configuration information from the Core System Deployer, which provides the basis for setting user permissions and Core System configuration.
- The Core System Manager interacts with the Data Provider and Data Subscriber to exchange the information necessary to allow their data distribution actions, per the rules and procedures provided by the Core System Deployer.
- The Core System Manager works with the Core Certification Authority in the performance of re-certification according to the Core Monitoring Plan. It also receives notification of certification changes of other Cores. This includes certification of new Cores and failure to re-certify an existing Core.
- The Core System Manager receives rule and policy updates from Local Policy Setting Entities.
- The Core System Manager receives Core operating procedures from the Core System Developer.
- The Core System Manager receives national laws and regulations from National Policy Setting Entities.
- The Core System Manager receives External Support System certifications from the Core Certification Authority.

4.1.5.3.5 Data Subscribers

Data Subscribers are End Users that wish to receive data through the Core using the Core's Data Distribution services.

Relationships:

- Data Subscribers may enter into an agreement with the Core, depending on the Core's operational policies.

4.1.5.3.6 Data Providers

Data Providers are End Users that wish to provide data through the Core using the Core's Data Distribution services.

Relationships:

- Data Providers may enter into an agreement with the Core, depending on the Core's operational policies.

4.1.5.3.7 External Support Systems

External Support Systems are systems outside the control of the Core that provide services critical to Core Operations (e.g., ESS DSRC CA).

Relationships:

- The Core Certification Authority certifies which ESS are qualified to provide services to a Core, based on the USDOT Core System Documents (SyRS and SAD).

4.1.5.3.8 Field Node Manufacturer

This is the manufacture of a field node device. The Field Node Manufacturer device may want his device certified so that prospective users of his device know that the device is independently certified to be compatible with the *connected vehicle* environment.

Relationships:

- The Field Node Manufacturer submits samples of his device to the Core Certification Authority, establishes a plan for certification and receives the results of that certification from the Core Certification Authority.

4.1.5.3.9 Field Node Owner/Operator

The Field Node Owner/Operator is the owner and operator of the field infrastructure used by the DSRC Device Owner to interface with the Core System. This could be a cellular provider or DSRC field infrastructure manager.

Relationships:

- The Field Node Owner/Operator enters into a usage agreement with the Core System Deployer (which describes what use the Core can make of the field node infrastructure) and shares sufficient field configuration information to enable geo-broadcast functions. This includes information updates when the configuration and/or locations of field nodes changes.
- The Field Node Owner/Operator may have a relationship with the Core Certification Authority, where the performance of its Field Nodes is regularly monitored and reported upon.

4.1.5.3.10 Local Policy Setting Entities

Local Policy Setting Entities establish policies, procedures, and regulations that may impact the configuration, operation and/or maintenance of the Core System. Entities filling this role could include local DoTs, local, county and state government, regional transportation authorities, and in particular the IT interests of those agencies.

Relationships:

- Local Policy Setting Entities provide restrictions on procedure, data use and/or data provision to the Core System Deployer, who uses it to establish system operational procedures and configuration, and is provided to the Core System Manager for operations.
- Local Policy Setting Entities send rule and policy updates as needed to the Core System Manager.
- Local Policy Setting Entities send local regulations concerning periodic certification of Mobile Devices to the Mobile Device Owner.
- Local Policy Setting Entities send local regulations concerning periodic certification of Mobile Devices to the Core Certification Authority.

- Local Policy Setting Entities provide local regulations, policies and practices to National Policy Setting Entities, and receive national regulations policies and practices from National Policy Setting Entities.

4.1.5.3.11 National Policy Setting Entities

National Policy Setting Entities establish policies, procedures and regulations that may impact the configuration, operation and/or maintenance of the Core System and *connected vehicle* applications. Entities filling this role could include federal agencies such as the USDOT and FCC, and trade groups such as IEEE or AASHTO.

Relationships:

- The National Policy Setting Entities national laws and regulations to the Core System Manager.
- The National Policy Setting Entities provide national regulations regarding wireless spectrum use and management to the Core Certification Authority.
- National Policy Setting Entities receive local regulations, policies and practices from Local Policy Setting Entities, and provide national regulations policies and practices to Local Policy Setting Entities.

4.1.5.3.12 Mobile Device Manufacturer

This is the manufacture of a Mobile device. The Mobile Device Manufacturer may want his device certified so that prospective users of his device know that the device is independently certified to be compatible with the *connected vehicle* environment.

Relationships:

- The Mobile Device Manufacturer submits samples of the device to the Core Certification Authority. It establishes a plan for certification and receives the results of that certification from the Core Certification Authority.

4.1.5.3.13 Mobile Device Owner

This is the owner of the Mobile device. He may have to have his device recertified periodically, to verify that it was operating properly within the *connected vehicle* environment.

Relationships:

- The Mobile Device Owner receives local regulations related to the recertification of his Mobile Device from Local Policy Setting Entities.
- The Mobile Device Owner submits his Mobile Device to the Core Certification Authority for recertification and receives the results of that recertification according to the Core Certification Authority's Recertification Plan.

4.1.5.3.14 Standards Bodies

Standards bodies develop standards that are used to implement Core System interfaces.

Relationships:

- Standards Bodies make their standards available to the Core Certification Authority.

4.1.5.3.15 USDOT Core System Documents

The library of documents that make up the USDOT's specification for the Core System: The ConOps, SyRS and SAD are included, but this library should also contain documents relevant to the certification of End User Applications.

Relationships:

- USDOT Core System Documents are provided to and used by the Core Certification Authority.

4.1.5.4 View Description

This view supports a federation of Core Systems that interoperate because they are certified against a common set of standards. Cores do not necessarily have to provide all of the same services or operate the same software. For example one Core could provide User Trust Management over a large area but not Data Distribution, while other Cores could provide Data Distribution services over parts of that same area. The Cores provide different services and they could even run different software. They must however, meet certain standards of operation.

The Core Certification Authority acts as the sole gatekeeper to determining who meets those standards of operation. This implies that recognition of another Core is not possible without that Core being able to identify itself as certified by the Core Certification Authority. This in turn implies that the means for that identification, an X.509 digital certificate associated with that Core, cannot be granted without the Core Certification Authority's involvement. Thus the Core Certification Authority must be the controller of the X.509 certificate authority that distributes Core root certificates.

Similarly, the Core Certification Authority determines what External Support Systems may provide services to the Core. Thus addition of new ESS to the *connected vehicle* environment will have a process to follow.

Other relationships are more straightforward. Core relationships with Data Publishers and Data Subscribers are negotiated with individual Cores; if a Data Subscriber or Data Publisher interacts with more than one Core, they will have a relationship with each Core.

One of the largest risks associated with successful deployment and operation of the Core System is the establishment and makeup of the Core Certification Authority. This body could become very large and unwieldy and stand as more a barrier than a guide. This document provides a start for what tasks the body must shoulder (i.e., "the what"). Determining "the who" is a task that must be undertaken in advance of deployment in order for a well-designed Core Certification Authority to learn how to function as an enabling organization. First steps could include determining areas of responsibility that can be easily distinguished from one another, and then determining what stakeholders have most interest in those areas. These could form the initial organizations that provide Core Certification Authority functions.

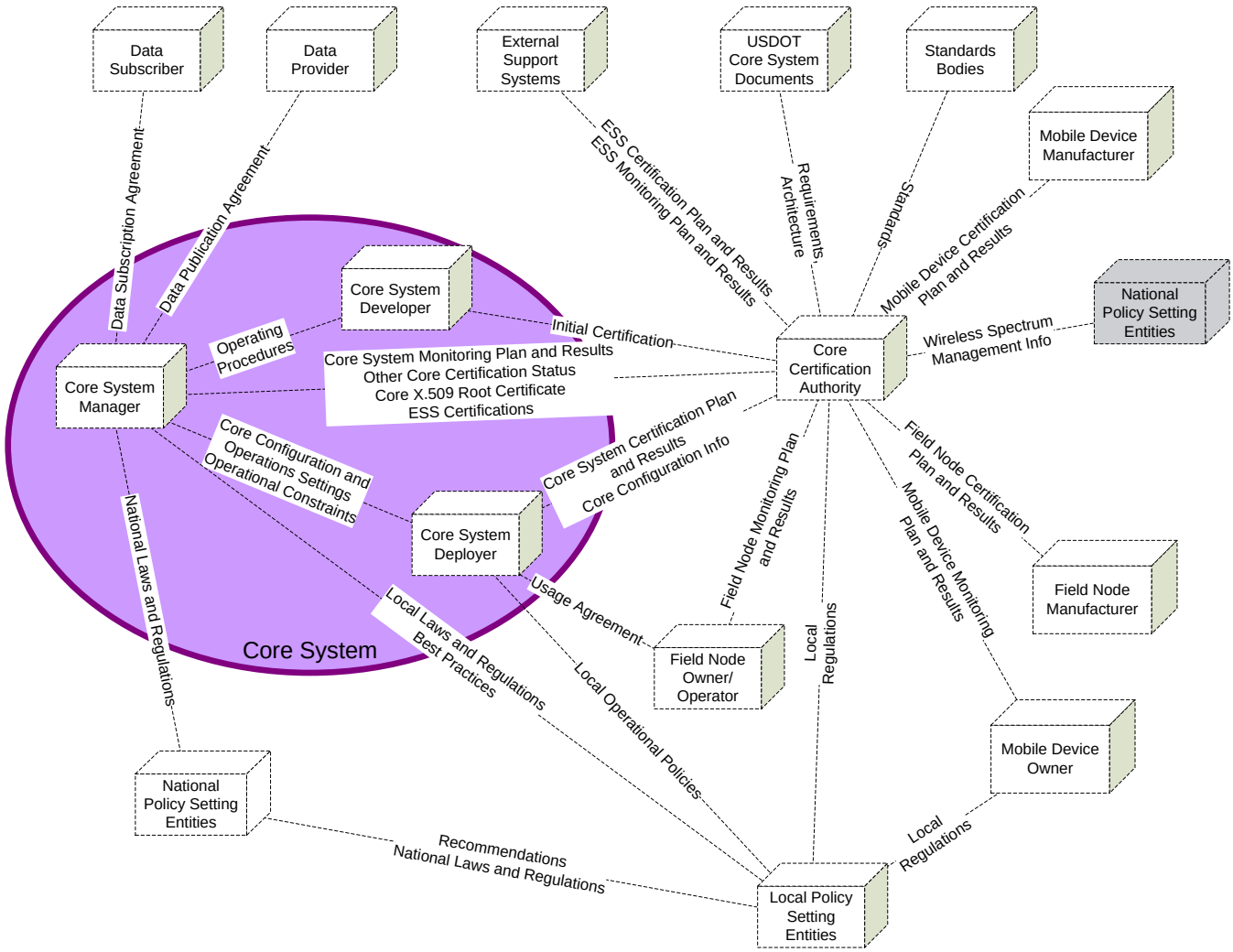


Figure 4-5: Enterprise View -- Governance

4.1.5.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Security Credentials Distribution
Enterprise Viewpoint: Enterprise View – Operations
Enterprise Viewpoint: Enterprise View – Core System and Application Development and Deployment
Enterprise Viewpoint: Enterprise View – Configuration and Maintenance
Enterprise Viewpoint: Enterprise View – Business Model Facilitation
Functional Viewpoint: Functional View – System Configuration
Functional Viewpoint: Functional View – User Configuration
Functional Viewpoint: Functional View – System Monitor and Control
Functional Viewpoint: Functional View – Core Backup

4.1.6 Enterprise View – Business Model Facilitation

4.1.6.1 Introduction

This view explores the enterprise relationships that may exist as part of business models that leverage the Core System to deploy applications and exchange data between System Users.

Many of these relationships require or acknowledge the possibility that one enterprise in the relationship compensates the other enterprise in the relationship. Compensation includes (but is not limited to) monetary exchange (A pays B) and information exchange (A provides information to B). This could result in the development of business models where for example the Core's operations are funded by application providers in exchange for data provided through the Core.

The purpose of this view is not to define business models, but to illustrate the relationships that could lead to them.

4.1.6.2 Concerns Addressed by this View

Organization/Resources	Who needs to contribute resources to Core System development, testing, transition, and operations?
Risks	What relationships between Core Enterprise Objects and external Enterprise Objects provide risks to Core System development, deployment, operations, and maintenance? What steps can be taken to lessen risks that are a function of Enterprise relationships?
Maintainability	Who needs to contribute resources to Core System maintenance activities? What resources are required to support Core System maintenance?

4.1.6.3 Object Definitions and Roles

4.1.6.3.1 Application Certification Body

The Application Certification Body certifies applications for deployment within the *connected vehicle* environment. This is likely to include several different bodies, each focused on certification for a particular application area (e.g., safety, mobility, e-commerce).

Relationships:

- The Application Certification Body enters into an agreement with the End User Application Developer to perform certification testing on the End User Application Developer's application. The Application Certification Body may require compensation from the End User Application Developer in order to provide this certification.

4.1.6.3.2 Center Owner/Operator

The Center Owner/Operator is the owner and operator of the Center that interfaces with the Core System. For example, this could be an information service provider or traffic management center. This object also includes the Center facility which is not drawn separately to help keep the diagram readable.

Relationships:

- The Center Owner/Operator may establish a license agreement with the Application Deployer. The license agreement establishes what the application component may be used for, on what device(s), and what requirements are levied on Center Owner/Operator in order to use the application. There may be an exchange of compensation as part of this agreement.
- The ESS DSRC Registration Authority may require the Center Owner/Operator to enter into a distribution agreement in order to acquire digital certificates. This could require the Center Owner/Operator to compensate the ESS DSRC Registration Authority.
- The Core System Manager may require the Center Owner/Operator to enter into a usage agreement to use the Core System. This may require the Field Node Owner/Operator to compensate the Core System Manager.

4.1.6.3.3 Core Certification Authority

The Core Certification Authority verifies that a Core System and other *connected vehicle* devices meet specifications set forth in the USDOT's documents and associated standards.

Relationships:

- The Core Certification Authority works with the Core System Manager to create the Core's Certification Plan. The Core Certification Authority may require compensation from the Core System Manager in order to establish and execute this plan.
- The Core Certification Authority provides initial Core certification to the Core System Deployer, which may require compensation.

4.1.6.3.4 Core System Deployer

The Core System Deployer deploys the Core System.

Relationships:

- The ESS DSRC Registration Authority may require the Core System Deployer to enter into a distribution agreement in order to acquire digital certificates. This could require the Core System Deployer to compensate the ESS DSRC Registration Authority.
- The Core System Deployer provides data distribution configuration guidance to the Core System Manager. This includes what kinds of data the Core System may accept and distribute as well as what types of users that data will be accepted from and provided to. This results in the establishment of user account types with associated permissions. All user types will be granted some minimal level permission to interact with the Core by the Core System Manager. The Core System Deployer provides takeover information to the Core System Manager, describing what has been agreed to between the Core System Deployer and the Deployer for another Core. There may be an exchange of compensation between the Core System Manager and Core System Deployer.
- The Core System Deployer negotiates agreements for takeover and operational scope with other Core System Deployers. This may require compensation.
- The Core System Deployer obtains initial Core certification from the Core Certification Authority, which may require compensation.

4.1.6.3.5 Core System Manager

The Core System Manager manages the operations of the Core System. This includes daily operations activities such as configuration changes, backups and other system administration tasks, and maintenance activities.

Relationships:

- The Core System Manager receives data distribution configuration guidance from the Core System Deployer. This includes what kinds of data the Core System may accept and distribute as well as what types of users that data will be accepted from and provided to. This results in the establishment of user account types with associated permissions. All user types will be granted some minimal level permission to interact with the Core by the Core System Manager.
- The Core System Manager receives takeover information from the Core System Deployer, describing what has been agreed to between the Core System Deployer and the Deployer for another Core.
- The Core System Manager enters into an agreement with an End User Application Deployer to grant permissions to users of the End User Application Deployer's application. This may require the establishment of user accounts for the End User Application Deployer as well.
- The Core System Manager may require the Field Node Owner/Operator to enter into a usage agreement to use the Core System. This may require the Field Node Owner/Operator to compensate the Core System Manager.
- The Core System Manager may require the Center Owner/Operator to enter into a usage agreement to use the Core System. This may require the Field Node Owner/Operator to compensate the Core System Manager.
- The Core System Manager may require the Mobile Device Operator to enter into a usage agreement to use the Core System. This may require the Mobile Device Operator to compensate the Core System Manager, or the Core System Manager to compensate the Mobile Device Operator (e.g. if the device is providing data).
- The Field Node Owner/Operator may require the Core System Manager to enter into a usage agreement to use Field Nodes. This may require the Core System Manager to compensate the Field Node Owner/Operator.
- The Core System Deployer agrees to the Core System Certification Plan with the Core Certification Authority; periodic recertification may require compensation.

4.1.6.3.6 End User Application Deployer

The Application Deployer provides application components to System Users.

Relationships:

- The End User Application Developer enters into an agreement with the End User Application Deployer for the End User Application Deployer to deploy the End User Application Developer's application. The End User Application Deployer may compensate the End User Application Developer to acquire the application and deploy it.
- The Application Deployer may establish a license agreement with the Field Node Owner/Operator. The license agreement establishes what the application component may be used for, on what device(s), and what requirements are levied on Field Node Owner/Operator in order to use the application. There may be an exchange of compensation as part of this agreement.

- The Application Deployer may establish a license agreement with the Mobile Device Operator. The license agreement establishes what the application component may be used for, on what device(s), and what requirements are levied on Mobile Device Operator in order to use the application. There may be an exchange of compensation as part of this agreement.
- The Application Deployer may establish a license agreement with the Center Owner/Operator. The license agreement establishes what the application component may be used for, on what device(s), and what requirements are levied on Center Owner/Operator in order to use the application. There may be an exchange of compensation as part of this agreement.
- The Application Deployer may need an agreement with the Core System Manager establishing permissions for his application components. The Core System Manager enters into an agreement with an End User Application Deployer to grant permissions to users of the End User Application Deployer's application. This may require the establishment of user accounts for the End User Application Deployer as well. The End User Application Deployer may be required to compensate the Core System Manager in exchange.

4.1.6.3.7 End User Application Developer

The End User Application Developer develops applications targeted to End User devices and systems.

Relationships:

- The End User Application Developer enters into an agreement with the Application Certification Body for the Application Certification Body to perform certification testing on the End User Application Developer's application. The End User Application Developer may have to provide compensation to the Application Certification Body in order to obtain this certification.
- The End User Application Developer enters into an agreement with the End User Application Deployer for the End User Application Deployer to deploy the End User Application Developer's application. Application deployment may require an exchange of compensation.

4.1.6.3.8 ESS DSRC Registration Authority

The ESS DSRC Registration Authority takes the role of RA in the public-key infrastructure architecture used to support IEEE 1609.2 certificate distribution.

Relationships:

- The ESS DSRC Registration Authority may require the Center Owner/Operator to enter into a distribution agreement in order to acquire digital certificates. This could require the Center Owner/Operator to compensate the ESS DSRC Registration Authority.
- The ESS DSRC Registration Authority may require the Core System Deployer to enter into a distribution agreement in order to acquire digital certificates. This could require the Core System Deployer to compensate the ESS DSRC Registration Authority.
- The ESS DSRC Registration Authority may require the Field Node Owner/Operator to enter into a distribution agreement in order to acquire digital certificates. This could require the Field Node Owner/Operator to compensate the ESS DSRC Registration Authority.

- The ESS DSRC Registration Authority may require the Mobile Device Operator to enter into a distribution agreement in order to acquire digital certificates. This could require the Mobile Device Operator to compensate the ESS DSRC Registration Authority.

4.1.6.3.9 Field Node Owner Operator

The Field Node Owner/Operator is the owner and operator of the field infrastructure used by the Mobile Device Operator to interface with the Core System. This could be a cellular provider or DSRC field infrastructure manager for example.

Relationships:

- The Field Node Owner/Operator may establish a license agreement with the Application Deployer. The license agreement establishes what the application component may be used for, on what device(s), and what requirements are levied on Field Node Owner/Operator in order to use the application. There may be an exchange of compensation as part of this agreement.
- The Field Node Owner/Operator may enter into a roaming agreement with other Field Node Owner/Operators, so that if a Mobile Device Operator is permitted to use one Field Node Owner/Operator's network, he could use the others' as well. There may be an exchange of compensation to support this agreement.
- The Field Node Owner/Operator may require the Mobile Device Operator to enter into a usage agreement to use Field Node infrastructure. This may require the Mobile Device Operator to compensate the Field Node Owner/Operator.
- The Core System Manager may require the Field Node Owner/Operator to enter into a usage agreement to use the Core System. This may require the Field Node Owner/Operator to compensate the Core System Manager.
- The Field Node Owner/Operator may require the Core System Manager to enter into a usage agreement to use Field Nodes. This may require the Core System Manager to compensate the Field Node Owner/Operator.
- The ESS DSRC Registration Authority may require the Field Node Owner/Operator to enter into a distribution agreement in order to acquire digital certificates. This could require the Field Node Owner/Operator to compensate the ESS DSRC Registration Authority.
- The ESS DSRC Registration Authority may require the Core System Deployer to enter into a distribution agreement in order to acquire digital certificates. This could require the Core System Manager to compensate the ESS DSRC Registration Authority.

4.1.6.3.10 Mobile Device Operator

This is the operator of the Mobile device. If the device is mounted in a vehicle, this is the driver. If the device is handheld or carried, this is the user of the device.

Relationships:

- The ESS DSRC Registration Authority may require the Mobile Device Operator to enter into a distribution agreement in order to acquire digital certificates. This could require the Mobile Device Operator to compensate the ESS DSRC Registration Authority.
- The Field Node Owner/Operator may require the Mobile Device Operator to enter into a usage agreement to use Field Node infrastructure. This may require the Mobile Device Operator to compensate the Field Node Owner/Operator.

- The End User Application Deployer may establish a license agreement with the Mobile Device Operator. The license agreement establishes what the application component may be used for, on what device(s), and what requirements are levied on Mobile Device Operator in order to use the application. There may be an exchange of compensation as part of this agreement.
- The Core System Manager may require the Mobile Device Operator to enter into a usage agreement to use the Core System. This may require the Mobile Device Operator to compensate the Core System Manager, or the Core System Manager to compensate the Mobile Device Operator (e.g. if the device is providing data).

4.1.6.3.11 Mobile Device Owner

The Mobile Device Owner is the owner of the Mobile Device that participates in the *connected vehicle* environment.

Relationships:

- The Mobile Device Owner may enter into a usage agreement with the Mobile Device Operator, allowing the Operator to use the Mobile Device. This may require the Operator to provide some form of identification to the Owner, and may require compensation.

4.1.6.3.12 Mobile Device Vendor

The Mobile Device Vendor is the entity that provides the Mobile Device to the Mobile Device Operator. This may be the manufacturer of the device, or it could be a reseller.

Relationships:

- The Mobile Device Vendor enters into a purchase agreement with the Mobile Device Owner, providing the Owner with a Mobile Device in exchange for compensation.

4.1.6.3.13 Registration Authority

The Registration Authority provides digital certificates establishing trust to qualifying entities. The RA has relationships with other entities to acquire these certificates, which is not defined in this view (see Enterprise View – Security Credentials Distribution for more information).

Relationships:

- The Registration Authority provides certificates to the Mobile Device Owner/Operator. The Mobile Device Owner/Operator may need to provide compensation to the RA in order to acquire these certificates.
- The Registration Authority provides certificates to the Field Node Owner/Operator. The Field Node Owner/Operator may need to provide compensation to the RA in order to acquire these certificates.
- The Registration Authority provides certificates to the Center Owner/Operator. The Center Owner/Operator may need to provide compensation to the RA in order to acquire these certificates.
- The Registration Authority provides certificates to the Core System Deployer. The Core System Deployer may need to provide compensation to the RA in order to acquire these certificates.

4.1.6.3.14 Other Field Node Owner/Operator

This is another owner and operator of field infrastructure used by the Mobile Device Owner to interface with the Core System. This could be a cellular provider or DSRC field infrastructure manager, for example.

Relationships:

- The Field Node Owner/Operator may enter into a roaming agreement with other Field Node Owner/Operators, so that if the Mobile Device Owner is permitted to use one Field Node Owner/Operator's network, he could use the others' as well.

4.1.6.4 View Description

Figure 4-6 illustrates the relationships between parties identified as playing a role in the *connected vehicle* environment that may exchange compensation and therefore play a part in a business model. These entities may thus contribute resources to the operations or even development of the Core and *connected vehicle* applications. Alternatively or additionally, entities may contribute information which can be used by the recipient to produce a product that can in turn provide resources.

Some of these relationships are between organizations, and some are between organizations and individuals. These latter relationships, for instance those involving the Mobile Device Operator, have more risk: creating individual relationships could be difficult, thus the organization that controls the individual's access to the *connected vehicle* environment is best positioned to establish that relationship. This is the Mobile Device Owner or, when both the Owner and the Operator are the same, the Mobile Device Vendor. The Vendor or collection of vendors may thus have a dominating position on the marketplace.

The reason this is an issue is that one of the driving factors behind the *connected vehicle* environment is its potential for delivering data to traffic operators. Depending on the scope and scale of the agreements between Mobile Device Vendor and Mobile Device Owner, and between Owner and Field Node Owner/Operator, access to the types of data desired by public agencies could be difficult or expensive. One potential solution to this lies in public agency control of the right-of-way on which Field Nodes will be installed. Public agencies could use this as leverage to ensure they have access to data, if that data is provided through such Field Nodes. Public agencies have no such leverage however, for data that is provided using existing communications mechanisms.

Compensation between the Core System Manager and enterprises external to the Core System can be leveraged toward maintenance activities. As noted in Enterprise View – Configuration and Maintenance, the Core System Manager works with the Core System Maintainer to maintain the system. Since the Core System Manager may have compensatory relationships with external enterprises, it can use revenues based on these relationships (e.g., with the Center Owner/Operator, End User Application Deployer, Field Node Owner Operator) to fund maintenance activities.

Similar relationships can fund other aspects of the Core System that might require more investment. In particular, the implementation of data aggregation, data parsing and sampling functions (see Functional View – Data Distribution) could be expensive. These may be justified if the products produced by these activities were of value to entities with compensatory relationships with the Core System Manager (e.g. Center Owner/Operator). It is conceivable that several Center Owner/Operators may be willing to pay for an aggregated data stream so that their communications needs to the Core System were reduced, for

example. Without such relationships, Core System Deployers and Core System Managers may find the costs of such data manipulations too expensive to deploy and maintain.

An additional complicating fact is that each state in the United States has substantively leeway to control data policies within its jurisdiction. Policies in one state may conflict with policies an adjacent state. This may curtail data exchange and result in functional differences between jurisdictions, which could in turn affect compensatory agreements and business models.

Note that the following diagram emphasizes the exchange of compensation with green arrows.

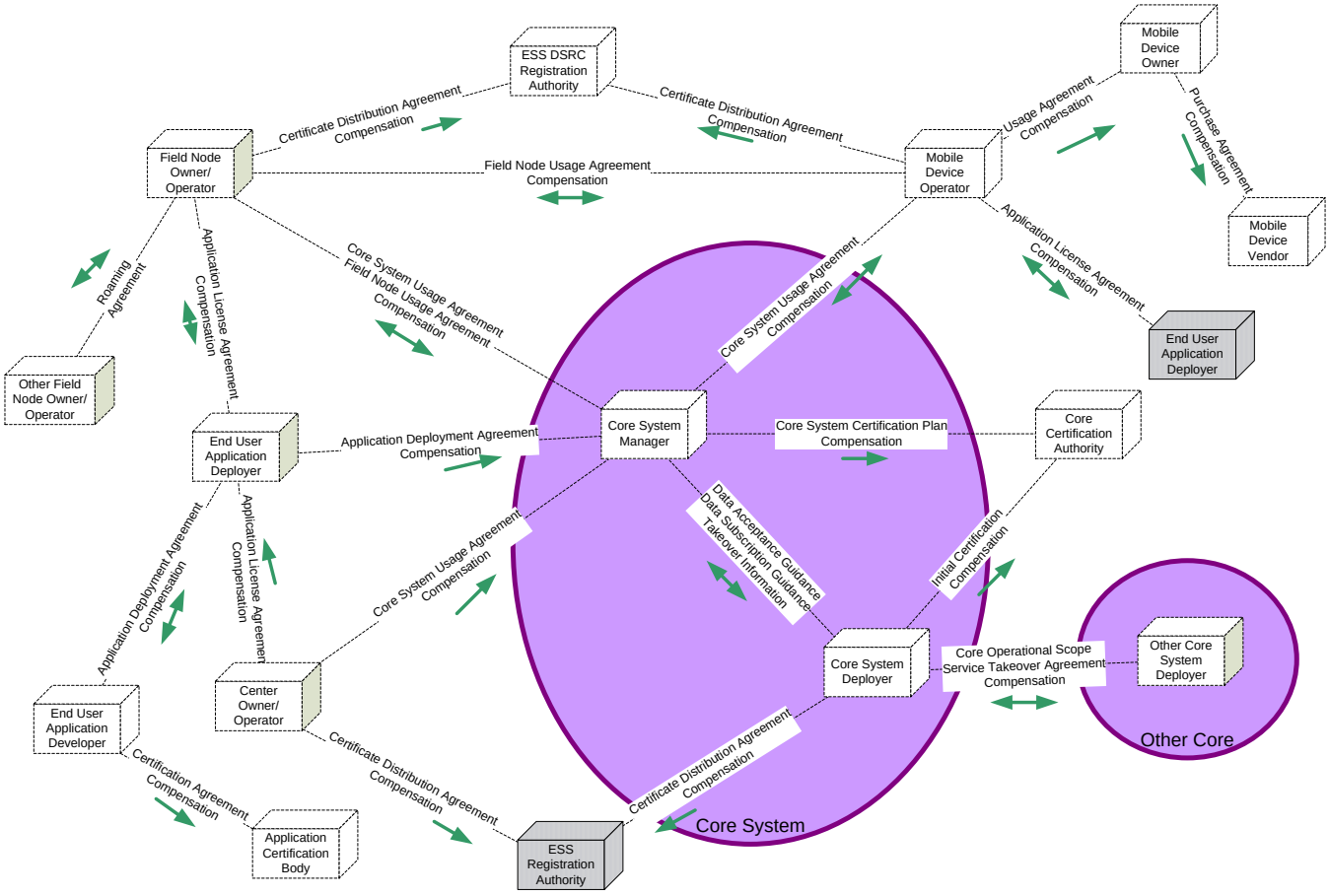


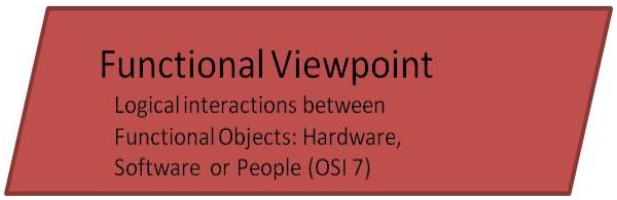
Figure 4-6: Enterprise View – Business Model Facilitation

4.1.6.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Security Credentials Distribution
Enterprise Viewpoint: Enterprise View – Core System and Application Development and Deployment
Enterprise Viewpoint: Enterprise View – Governance

4.2 Functional Viewpoint



The first Functional View starts with the subsystems defined in the Concept of Operations. Subsequent views explore the functions associated with a particular scenario (e.g., Data Distribution). Scenario-based views were chosen to illustrate functions where significant interactions take place between subsystems.

Functional Views are presented for ten aspects of Core System operation:

- High Level (subsystem) functionality
- Distribution of data (primarily through a publish-subscribe mechanism)
- Configuration of Core services
- Configuration of System User access
- Management of Core operations,
- Management and distribution of digital certificates
- Detection of misbehavior and action in response to detection
- Decryption of encrypted messages
- Provision of Internet connectivity and routing
- Backup of data and services between Cores

Many functions appear in more than one view. A function that appears in multiple views always provides the same function, but may interact with different functions in different views. Unless otherwise stated, all views are equally valid. For example, function A may appear in two different views. In one view it interacts with functions B and C, and in another view it interacts with C, D and E. Both views are equally valid and thus function A interacts with B, C, D and E.

Some of these views refer to the type or identity of System Users. When referring to the identity of a System User, the Core System is indicating unique information attributable to a specific individual person. When referring to a System User type, the Core is indicating unique information attributable to a group of individuals that share characteristics but are not necessarily uniquely identifiable. For example, User-ID 1234 may refer to John Smith, a unique user of the Core, and Type-ID Bus may refer to System Users that are all buses.

System User types are not defined in this document, with one exception: Anonymous Mobile Users. Anonymous Mobile Users do not provide their identity when they communicate with the Core System. Anonymous Mobile Users may be required to provide their identity in order to initially obtain digital certificates, but the entity that requests identity is the ESS DSRC Registration Authority, not the Core System.

Table 4-2 shows which Functional Views are relevant to each stakeholder.

Table 4-2: Functional View Stakeholder Matrix

Functional View \ Stakeholder	Mobile User	Field User	Center User	Operator	Acquirer	Maintainer	Developer	Manager	Tester	Policy Setter	Application Developer	Device Developer	Service Provider
Top Level	■	■	■	■	■		■	■			■	■	
Data Distribution	■	■	■	■	■	■	■	■	■		■	■	■
System Configuration				■	■	■	■	■	■				
User Configuration	■	■		■	■	■	■	■	■			■	■
System Monitor and Control	■	■	■	■	■	■	■	■	■			■	
Credentials Distribition						■	■		■				
Misbehavior Management	■	■	■	■		■	■		■				
Core Decryption						■	■		■				
Networking				■		■	■		■		■		■
Core Backup	■	■	■	■	■	■	■	■	■				

The following Functional Views are relevant to each Stakeholder as shown in the Table above:

- Mobile Users: Top Level, Data Distribution, User Configuration, System Monitor and Control, Misbehavior Management, and Core Backup
- Field Users: Top Level, Data Distribution, User Configuration, System Monitor and Control, Misbehavior Management, and Core Backup
- Center Users: Top Level, Data Distribution, System Monitor and Control, Misbehavior Management, and Core Backup
- Operators: Top Level, Data Distribution, System Configuration, User Configuration, System Monitor and Control, Misbehavior Management, Networking, and Core Backup
- Acquirers: Top Level, Data Distribution, System Configuration, User Configuration, System Monitor and Control, and Core Backup
- Maintainers: Data Distribution, System Configuration, User Configuration, System Monitor and Control, Credentials Distribution, Misbehavior Management, Core Decryption, Networking, and Core Backup
- Developers: Top Level, Data Distribution, System Configuration, User Configuration, System Monitor and Control, Credentials Distribution, Misbehavior Management, Core Decryption, Networking, and Core Backup
- Managers: Top Level, Data Distribution, System Configuration, User Configuration, System Monitor and Control, and Core Backup
- Testers: Data Distribution, System Configuration, User Configuration, System Monitor and Control, Credentials Distribution, Misbehavior Management, Core Decryption, Networking, and Core Backup
- Policy Setters: none.
- Application Developers: Top Level, Data Distribution, and Networking
- Device Developers: Top Level, Data Distribution, User Configuration, and System Monitor and Control
- Service Providers: Data Distribution, User Configuration, and Networking

4.2.1 Functional View – Top Level

4.2.1.1 Introduction

This view documents the top level Functional View of the Core System. The objects in this view map to the subsystems in the Concept of Operations, and provide the basis for all subsequent Functional Views.

4.2.1.2 Concerns Addressed by this View

Functionality	How does the Core System monitor the services it provides? How does the Core System support the coordination of resources between different Cores? How does the Core System function internally? How does the Core System transition between operational modes?
Security	What functional elements are involved in the distribution and revocation of digital certificates, and what roles do those entities have? What functional elements are involved in the detection of misbehavior by System Users, and what roles do those entities have?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?

4.2.1.3 Object Definitions and Roles

4.2.1.3.1 Actors

4.2.1.3.1.1 **Core Certification Authority**

This is the Core Certification Authority as defined in Enterprise View – Governance.

4.2.1.3.1.2 **End User Application Deployer**

This is the entity that provides an application to the System User.

4.2.1.3.1.3 **ESS DSRC RA**

This actor represents the External Support System functioning as a registration authority for IEEE 1609.2 certificate distribution.

4.2.1.3.1.4 **ESS X.509 CA**

This is the External Support System that provides the Core System with its X.509 certificate.

4.2.1.3.1.5 **External CAs**

The External CAs actor represents External Support Systems that function as Certificate Authorities. This includes the ESS DSRC Certificate Authority and ESS X.509 Certificate Authority.

4.2.1.3.1.6 **Field Node Owner/Operator**

This is the owner/operator of Field Nodes used to perform geographic broadcast.

4.2.1.3.1.7 **Operator**

The Operator is the day-to-day administrator of the Core System. The Operator interacts with the Core through various "Provide Operator Interface to…" Functional Objects.

4.2.1.3.1.8 **Other Core**

This actor represents another Core System.

4.2.1.3.1.9 **System User**

This actor represents a System User including Center, Mobile and Field.

4.2.1.3.1.10 **Time Source**

This actor represents the external time source the Core uses as the basis for time.

4.2.1.3.2 Core2Core

Core2Core (C2C) interfaces with other Core Systems, advertising its jurisdictional scope, offered services, and services it desires from other Cores. Core2Core maintains a knowledge base of services available from other Cores including other Cores' jurisdictional scope.

Core2Core is responsible for ensuring that the Core does not encroach on the scope of another Core. Core2Core also accepts error messages from Mobile Users that might indicate a cross-jurisdictional compatibility or scope coverage issue.

Associated Information Objects:

All Core2Core Information Objects are shared between Cores.

- **Backup Data**: This is data extracted from a Core data store.
- **C2C Misbehavior Report**: This includes the certificate ID associated with misbehavior, the type of misbehavior, time of the misbehavior, time of the misbehavior detection, certificate ID of the misbehavior report generator, and if available the identity of the misbehaving entity.
- **Core Service Status Query**: This requests the status of the targeted Core's services.
- **Service Status for Cores**: This describes the status of all of the Core's services.
- **Core Status Registration**: This requests that the targeted Core provide the sending Core with status information on a periodic basis.
- **Complete CRL**: This is the list of all active digital certificates issued by the Core that are now invalid.
- **CRL Deltas**: This is the list of all active digital certificates issued by the Core that are invalid and are not included on the Complete CRL message.
- **Core Conflict Info**: This is a description of the conflict in service provision between the two Cores.
- **Core Configuration Info**: This is a description of the services offered by the Core reporting the configuration info.
- **Data Backup Request**: This is the request sent to a Core that has a backup relationship with the sending Core.
- **Data Request**: This is the request sent to a Core that has previously backed up data.

- **Other Core Takeover Request**: This includes the service, coverage area, start and expected end time and expected performance load that the Core wants the destination Core to provide service for.
- **Restore Data**: This is formerly backed up data.

4.2.1.3.3 Data Distribution

Data Distribution (DD) maintains a directory of System Users that want data and facilitates the delivery of that data to those users. It supports two distribution mechanisms:

- Source-to-Points: The data provider communicates data directly to data consumers. In this case, no data is sent to the Core System.
- Publish-Subscribe: The data provider communicates data to the Core, which forwards it to all users that are subscribed to receive the data.

Data Distribution maintains a registry of which data subscribers get what data, according to the criteria defined above. Data Distribution does not store or buffer data beyond that which is necessary to complete publish-subscribe actions. If a given data subscriber is unable to receive data that it has subscribed to because of a communications or other system failure, the data may be lost.

Data Distribution repackages data it receives from data providers, stripping away the source header information while maintaining the message payload. It then sends the repackaged payload data to subscribers of that data.

Data Distribution also provides information to System Users that enables those users to communicate with a group of System Users, by maintaining information regarding available communications methods, coverage areas, addresses and performance characteristics for geo-cast communications.

Associated Information Objects:

- **Data Acceptance Info**: This is the response to the Data Provision Request.
- **Data Provision Request**: This includes the type of data to be provided, source type and area over which the System User will provide data.
- **Data Subscription Confirmation Info**: This is a response to the Data Subscription Request, describing the exact parameters of the System User's subscription.
- **Data Subscription Redirect Info**: This is a response to the Data Subscription Request, rejecting the subscription.
- **Data Subscription Request**: This is the request to subscribe to data.
- **Direct Data Distribution Info**: This is data describing 3rd parties that accept data that the Core does not accept.
- **Field Node Configuration Information**: This specifies the location, IP address, communications range, bandwidth and constraints for use of Field Node Infrastructure.
- **Geo-Cast Message**: This is a message that the System User wishes to distribute over a specific area, either once or repeatedly over a period of time.
- **Other Core Acceptance Info**: This is the response to the Data Provision Request, sent when the Core does not accept the data the System User wishes to provide.
- **Provided Data**: This is data received from System User Data Providers intended for the publish/subscribe engine.

- **Repackaged, Addressed Data**: This is data repackaged to match a specific subscriber's subscription criteria. This is data that was originally provided by other System Users (Data, above) that has been selected based on subscription criteria.

4.2.1.3.4 Misbehavior Management

Misbehavior Management (MM) analyzes messages sent to the Core System to identify System Users operating outside of their assigned permissions. It works with the User Permissions subsystem to identify suspicious requests and to maintain a record of specifically identifiable users that:

1. Provide false or misleading data
2. Operate in such a fashion as to impede other users
3. Operate outside of their authorized scope

Because most end users will rarely interface with the Core System, the Core will also accept reports of misbehaving users from System Users. System Users can send misbehavior reports that reference credentials attached to messages, and note the type of misbehavior in question. Misbehavior Management will record such reports, and according to a set of rules, determine when to revoke credentials from such reported misbehaving users. For anonymous users revocation is more complex, and may result instead in a lack of credential renewal.

Associated Information Objects:

- **System User Misbehavior Report:** The data describing misbehaving users.

4.2.1.3.5 Network Services

Network Services (NS) provides management for communications resources. It provides the Core System's Internet access and network security, including port monitoring and firewall functionality. It enables decisions about which communications medium to use when more than one is available. This includes identifying available communications methods current performance characteristics and applicable user permission levels. All data intended for the Core passes through Network Services functions.

4.2.1.3.6 Service Monitor

Service Monitor (SM) monitors the status of Core System services, interfaces, and communications networks connected to the Core. It informs System Users of the availability and status of Core services.

Service Monitor also monitors the integrity of internal Core System hardware and software and provides vulnerability protection. This includes periodic verification of the authenticity of Core software, monitoring for vulnerabilities including but not limited to virus detection and monitoring for patches to third party components. Should vulnerability be detected, or a component of the Core found to have lost integrity, the Service Monitor takes steps to mitigate against damage and performance degradation.

Service Monitor monitors the environmental conditions that Core components operate in (temperature and humidity) as well as the condition of its power system. It takes steps to mitigate against system failures in the event that environmental conditions exceed operating thresholds.

Service Monitor monitors the performance of all services and interfaces and makes performance metrics available to Core Personnel.

Associated Information Objects:

- **Performance Records:** This is detailed information describing the long-term performance of Core services, provided to the Core Certification Authority.
- **Service Status:** The status of the Core's subsystems. This may be limited to specific subsystems if in response to a query. For each service indicates current state and mode and if there is a known time for that state or mode to change.
- **Service Status Query:** A request for Core service status information that includes a listing of the services the System User desires status information about.
- **System User Status Registration:** A request from a System User to register for periodic reports of the Core's service status that includes the System User's IP Address and desired update frequency.

4.2.1.3.7 Time Synchronization

Time Synchronization (TS) uses a time base available to all System Users and makes this time available to all Core System services for use as a time reference.

Associated Information Objects:

- **Time:** This is time that is provided by an external Stratum-2 time source.

4.2.1.3.8 User Permissions

User Permissions (UP) provides tools allowing Core Personnel and other Core System subsystems to verify whether a given user, identified by digital certificate-based credentials, is authorized to request or perform the action requested in the message payload. It also maintains the status of System Users, whether they have a specific account or belong to an anonymous group, and their permissions (publish, subscribe, actions allowed to request, administrate, etc.). User Permissions provides tools for Core Personnel to create new users and groups, modify existing users and groups, and modify permissions associated with users and groups.

Associated Information Objects:

- **Application Permission Request:** A submission of application permissions that must be managed using DSRC certificates.
- **User Identity and Permission Request:** This is a submission of identity credentials and a request for permissions to use the Core System services.

4.2.1.3.9 User Trust Management

User Trust Management (UTM) manages access rules and credentials in the form of digital certificates, including X.509 certificates, for all System Users and Core System components that require and are entitled to them. It creates and distributes cryptographic keys to qualifying System Users. It works with User Permissions to determine whether a given user applying for credentials or keys is entitled to them. It also manages revocation of credentials, distributing CRLs of disallowed credentials to interested System Users. User Trust Management works with External Support System to manage certificates; information describing this approach is depicted in succeeding architecture views.

Associated Information Objects:

- **Credential Request**: This is request for credentials.
- **Credential Referral**: This message is provided in response to a credential request that the Core cannot satisfy.
- **Credentials**: This message contains a new digital certificate for the System User.
- **CRL**: This is the list of all active digital certificates issued by the Core that are now invalid.

- **CRL Request**: This request from a System User for that user to receive CRLs from the Core.
- **Ext X.509 Cert Request**: This is a request from the Core System to an external X.509 Certificate Authority, formatted according to that CA's requirements, requesting an X.509 certificate for the Core.
- **External CRL**: This a Certificate Revocation List received from an external CA.
- **Misbehaving User ID**: This contains the ID and type of misbehavior the System User has committed.
- **User Identification**: This is a request for special permissions for a System User from the ESS DSRC RA.
- **User Special Permissions**: This is the response to the ESS DSRC RA. It includes an identification of the special permissions the user is entitled to.
- **X.509 Certificate**: This is an X.509 Certificate formatted according to ITU-T X.509, including the Core System's identity.
- **X.509 CRL**: This is a list of all activated X.509 certificates that are invalid.

4.2.1.4 View Description

This view illustrates the highest level functions the Core System provides. The objects defined above map directly to the subsystems defined in section 5 of the ConOps. The object definitions provide most of the descriptive information for this view.

The view image specifies which subsystems provide external interfaces, and the high-level view of the types of information provided, received or exchanged across those external interfaces. More detailed information about the Information Objects exchanged between the Core and external entities is in the Functional View where the object is used, as well as Information View – Top Level External Objects.

Internal interfaces are not shown on the drawing for the sake of clarity. All subsystems communicate with all other subsystems. Details of subsystem-to-subsystem interaction are in subsequent Functional Views. Definitions of objects exchanged are in the relevant Functional View and the Information View – Top Level Internal Objects.

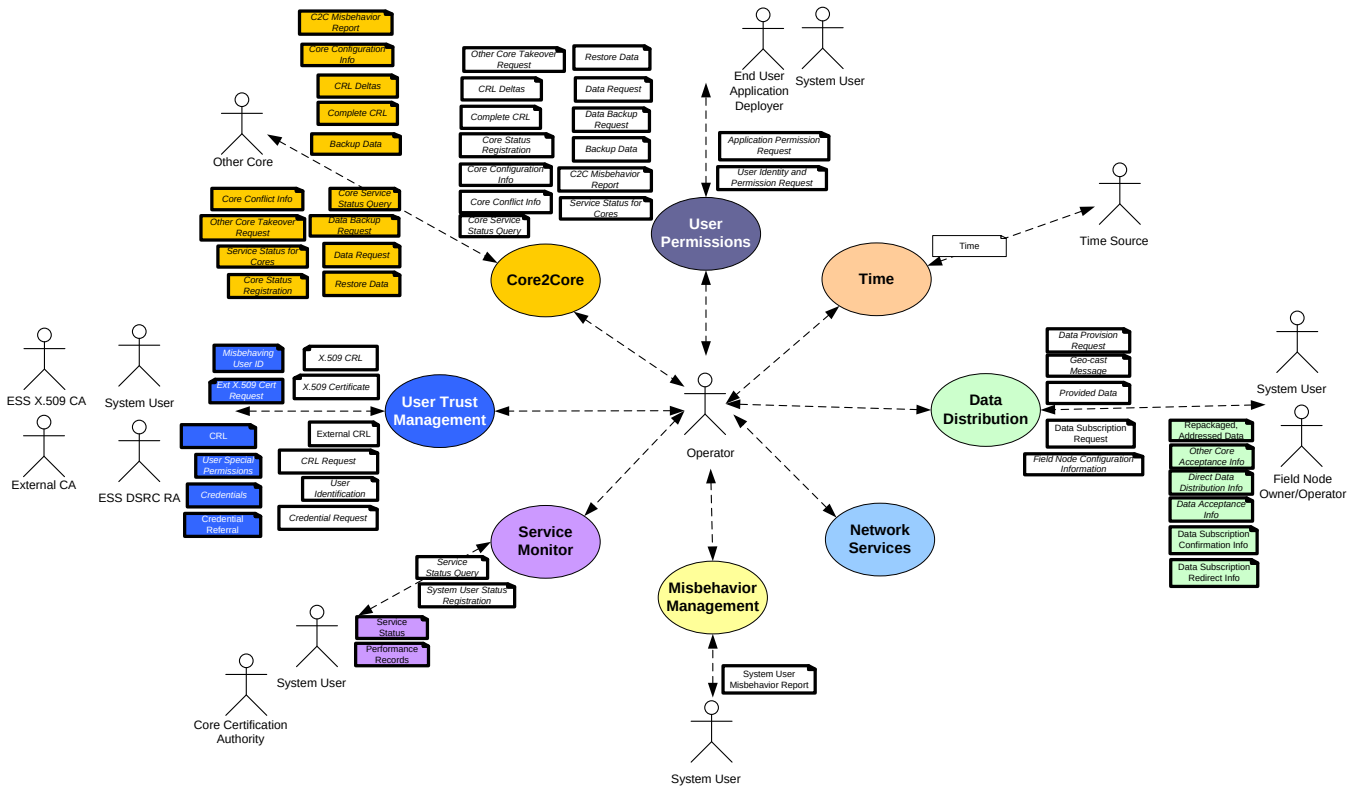


Figure 4-7: Functional View – Top Level

The Core System may be in one of four states, as illustrated in Figure 4-8:

- Installation: This state includes all pre-operational activities necessary to plan, develop, install and verify the procedures and system configurations used to support the Core System.
- Operational: This state includes all activities during the normal conduct of operations.
- Standby: The Core System operating in a Standby state will be providing backup to one or more other Cores. From the standby state the Core may take over the functions of another Core if required. Additionally, individual subsystems may be in the standby state so that individual subsystems can backup to other Cores' subsystems.

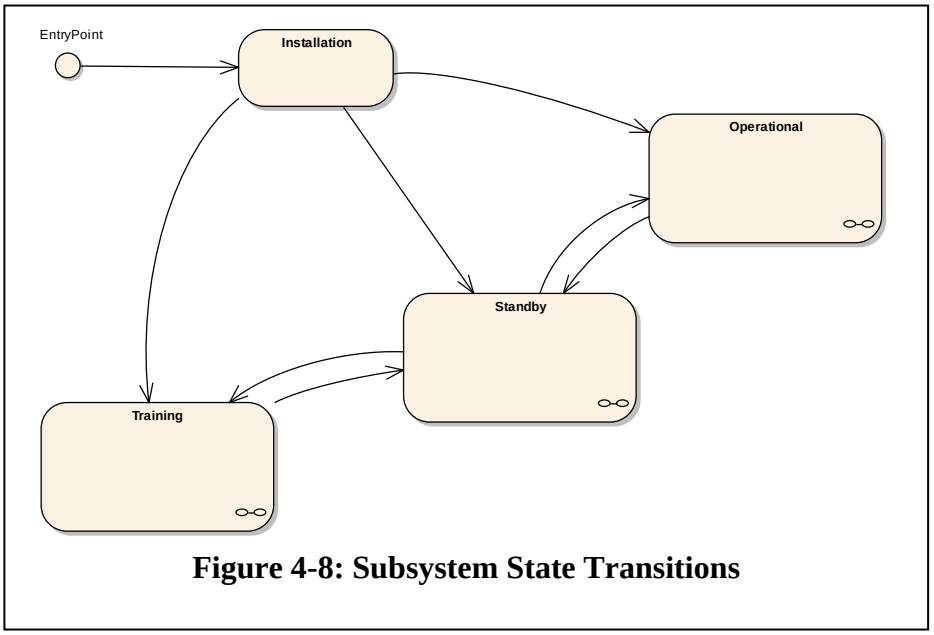


Figure 4-8: Subsystem State Transitions

- Training: The Core System will be placed in a Training state when it is used for imparting training on the Core features. Certain features like real-time display of log messages and debug messages may be enabled in the Training state which may not otherwise be accessible under normal conditions.

Within each state, Engineering Objects may be in different modes. For a discussion of these modes and how they apply to the objects that make up the Core System, see Connectivity View – State and Mode Transitions. For a discussion of how transitions between states and modes are accomplished see the relevant Functional View.

4.2.1.5 Configuration Information

The following views must be considered when changing this view:

Functional Viewpoint: Functional View – Data Distribution
Functional Viewpoint: Functional View – System Configuration
Functional Viewpoint: Functional View – User Configuration
Functional Viewpoint: Functional View – System Monitor and Control
Functional Viewpoint: Functional View – Credentials Distribution
Functional Viewpoint: Functional View – Misbehavior Management
Functional Viewpoint: Functional View – Core Decryption
Functional Viewpoint: Functional View – Networking
Functional Viewpoint: Functional View – Core Backup
Connectivity Viewpoint: Connectivity View – High Level
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Information Viewpoint: Information View – Top Level External Objects

4.2.2 Functional View – Data Distribution

4.2.2.1 Introduction

This view explores the Core System's Data Distribution functionality. System Users provide data, other System Users subscribe to data; the Core matches those providers and consumers without requiring them to enter into a relationship with the other.

The configuration of the System Users, including data subscribers and providers, is described in section 4.2.4 Functional View – User Configuration on page 117.

This view includes several optional functions: data aggregation, data parsing and data sampling. The extent to which these are included in a Core System is dependent on the services that a Core System Deployer wishes to deploy, and constrained by the available resources. These functions are processing intensive; their implementation may require a significantly greater capital investment. A Deployer may wish to provide these services because they can significantly reduce the size of the data stream sent to subscribers. This reduction in recurring communications costs may balance the processing investment for some deployments. Such an analysis depends on the costs of hardware used to operate Core SEOs, the scope of the Core System and costs of available communications. Analysis of these trade-offs is recommended for all Core System Deployers.

4.2.2.2 Concerns Addressed by this View

Interfaces	How difficult is it to develop applications that use Core System interfaces? How flexible are Core System data distribution interfaces? How does the Core System enable control of the services it provides?
Functionality	How does the Core System monitor the services it provides? How does the Core System function internally? How do the Core System's components work together? How does the Core System transition between operational modes?
Security	What functional elements are involved in the detection of misbehavior by System Users, and what roles do those entities have? How does the Core System maintain the integrity of information provided to it by System Users?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?
Evolvability	How easily can the Core's functionality be expanded to cover new needs if they arise?

	Does the functionality of the Core scale to support foreseeable demands from System Users?

4.2.2.3 Object Definitions and Roles

4.2.2.3.1 Actors

4.2.2.3.1.1 System User

This actor represents a System User, including Center, Mobile and Field. This includes Data Providers and Data Subscribers to the Core System data.

4.2.2.3.2 Special Functional Objects

These special objects interact with various other functions to perform operations on messages sent or received. The last maintains some common information that is used by many other functions. They are not shown interacting on the view description diagram for the sake of clarity.

4.2.2.3.2.1 Decrypt Messages Received Encrypted

This function accepts an encrypted message intended for the Core System and decrypts it using the Core's private key.

4.2.2.3.2.2 Encrypt Messages

This function encrypts a message using the public key of the intended recipient.

4.2.2.3.2.3 Sign Messages

This function attaches the Core System's digital signature to a message by encrypting a hash of the message content with the Core's private key.

4.2.2.3.2.4 Verify Authenticity of Received Messages

This function verifies that the message is not a duplicate, has a valid digital signature, that it meets formatting rules and that its contents are within prescribed limits for the message type.

4.2.2.3.3 Aggregate Data

This Optional Functional Object receives data from the Parse Data and/or Repackage Data functions, and aggregates messages or data elements as appropriate, before passing such aggregations to the Sample Data function. If this function is not implemented, all messages simply pass through it to the Sample Data function.

. Associated Information Objects:

- **Aggregated Data** is the aggregated, parsed, and repackaged data, provided to the Sample Data function.
- **Parsed Data** is an individual parsed data object received from the Parse Data function.
- **Repackaged Data** is composed of groups of same-type data, received from the Repackage Data function.
- **Subscription Details** is received from the Data Subscription Catalog function. It includes the subscriber's contact information, the data it is subscribed to and details about aggregation and sampling that should be applied to that data.

4.2.2.3.4 Check User Permission

This object accepts a System User ID or Operator ID, along with a type of operation that the user is attempting to access, and responds with whether or not the user is permitted that action.

Associated Information Objects:

- **Provider ID, function** is the identity or type of the System User data provider (which could be anonymous), which is sent to the Check User Permission function to verify whether the System User is permitted a given **function**.
- **Operator ID, function** is the identity of the operator, which is sent to the Check User Permission function to verify whether the operator is permitted a given **function**.
- **Permission** is the response from the Check User Permission function describing whether or not the **User** (System User, Core or Operator, depending on the original request) is permitted this action.

4.2.2.3.5 Data Acceptance Catalog

This data store maintains a catalog of data types and sources that are used by the Core's Data Distribution subsystem. It also maintains a list of the data types it does not accept but some other facility does. It allows changes to the catalog to update existing and add new data types, sources and combinations of types and sources. This function responds to queries about data types telling whether or not that type/source combination is accepted for data distribution. It also analyzes its data distribution coverage area, time, and source/type combinations against those reported by other Cores to determine if there are any overlaps (where multiple Cores provide the same services in the same area) or conflicts (where Cores provide different and conflicting information for the same area, for instance if one Core says that a given data type must go to external sink A, but another Core says that it accepts that data type).

Associated Information Objects:

- **Data Acceptance Query** is a request for information about the data types accepted by the Core. Includes the type of data to be provided, source type and location.
- **Data Acceptance Info** is a response to a Data Acceptance Query received from Receive Data from System Users. It indicates whether the data requested to be published is distributed by the Core, and if not includes information describing a 3rd party facility that does accept the data, if such a facility exists. Includes the type of data to be provided, source type and location.
- **Data Type and Source Request** is a query sent to the Data Subscription Catalog to determine if a given data type/source combination is currently being used. This includes the types of data and their source or source type.
- **Existing Acceptance and/or Changes** is the response to the Data Type and Source Request query. It specifies whether the data type/source in question is already being distributed or if not, that such distribution has been added.
- **Data Acceptance Details** is information describing what data (source/type, time, coverage area) the Core accepts, sent to the Provide Operator Interface to DD function.

4.2.2.3.6 Data Subscription Catalog

This data store maintains a catalog of data types and sources and the System Users that want to receive that data. It may allow specification of aggregation period and sampling rate for each System User. Allowable sampling rates and aggregation specifications may be restricted based on configuration and performance of this subsystem. Allows changes to the catalog to update existing and add new subscribers and responds to queries about data subscribers and their subscriptions.

Associated Information Objects:

- **Data Type and Source Request** is a query received from the Data Acceptance Catalog to determine if a given data type/source combination is currently being used. This includes the types of data and their source or source type.
- **Data Types and Sources** is received from the Parse Data and Repackage Data functions. It asks whether a given type of data from a given type of source is used by any data subscription.
- **Existing Acceptance and/or Changes** is the response to the Data Type and Source Request query from the Data Acceptance Catalog. It specifies whether the data type/source in question is already being distributed or if not, that such distribution has been added.
- **Data Acceptance or Discard** includes disposition for each data type/source combination included in **Data Types and Sources**. Possible dispositions are accept or discard. Within accept, this could include parse, aggregate and/or sample.
- **Subscription Details** is sent to the Match Data to Data Subscribers, Aggregate Data and Sample Data functions. It includes the subscriber's contact information, the data it is subscribed to and details about aggregation and sampling that should be applied to that data.

4.2.2.3.7 Geo-cast Device Catalog

This data store keeps track of the information necessary for System Users to perform geo-casting. This includes locations, addresses, and ranges of devices that accept geo-cast messages.

Associated Information Objects:

- **Geo-cast Addresses** provides the IP addresses of the devices in the geo-cast area specified by the Geo-Cast Area Specification query, sent to the Match Data to Data Subscribers function.
- **Geo-cast Area Specification** is a query received from Match Data to Data Subscribers. It specifies an area that a message must be distributed over.

4.2.2.3.8 Geo-cast Message Log

This data store logs geo-cast messages.

Associated Information Objects:

- **Geo-cast Message** comes from the Manage Geo-cast Messages function. It includes the geo-cast message as well as information describing the desired distribution area and time over which the distribution should be made.

4.2.2.3.9 Manage Geo-cast Messages

This object accepts geo-cast messages from the Receive Data from System Users function. It then establishes a schedule for geo-cast message publication and regularly provides messages to the Match Data to Data Subscribers function.

Associated Information Objects:

- **Geo-cast Message** comes from Receive Data from System Users. It includes information describing the desired distribution area and time over which the distribution should be made.
- **Repackaged Geo-cast** is a reformatted geo-cast message including location specification, provided to the Match Data to Data Subscribers and Geo-Cast Message Log functions.

4.2.2.3.10 Match Data to Data Subscribers

This object examines repackaged data and geo-cast messages to determine what data and messages go to which subscribers. Adds subscriber destination information and passes matched, sampled and aggregated data to be distributed.

Associated Information Objects:

- **Data Ready to be Matched** is the aggregated, parsed, sampled and repackaged data, received from the Sample Data function.
- **Reformatted Data with Destinations** is the Data Ready to be Matched with the IP addresses of subscribers to the data attached.
- **Repackaged Geo-cast** is a reformatted geo-cast message including location specification, received from the Manage Geo-cast Messages function.
- **Geo-cast Addresses** is the response to the Geo-cast Area Specification query that includes the IP addresses of the area specified in the query.
- **Geo-cast Area Specification** is a query sent to the Geo-cast Catalog that includes an area specification.
- **Subscription Details** is received from the Data Subscription Catalog function. It includes the subscriber's contact information, the data it is subscribed to and details about aggregation and sampling that should be applied to that data.

4.2.2.3.11 Misbehavior Reports Log

This object maintains the misbehavior reports received by the Core, including those submitted by System Users, and those from other Cores and those generated by internal Core monitoring processes. It accepts queries on those reports and provides the reports matching that query to requesting functions.

Associated Information Objects:

- **Suspicious Data** is Provided Data that does not pass the permission check. It is received from the Receive Data from System Users function.
- **Suspicious Geo-Cast** is a Geo-Cast Message that does not pass the permission check. It is received from the Receive Data from System Users function.

4.2.2.3.12 Modify DD Operational State

This object is the enabling function that allows the Operator to instruct various functions of Data Distribution to change the way they operate; this can include adding or deleting instantiations of an object or by commanding functions to enter into a different state or mode.

4.2.2.3.13 Parse Data

This Optional Functional Object receives data from Receive Data from System Users and parses it for individual data objects that are required to satisfy data subscriptions. Some data objects could be subject to parsing and others not; so the Repackage Data function is still required even if this object is implemented.

Associated Information Objects:

- **Provided Data** comes from the Receive Data from System Users Function. Data that is not subject to parsing is passed to the Repackage Data function. If the Parse Data function is not implemented, all Provided Data is passed to the Repackage Data function.
- **Repackaged Data** is composed of groups of same-type data, send to the Match Data to Data Subscribers function.

- **Data Types and Sources** is sent to the Data Subscription Catalog. This includes the types of data and their source or source type.
- **Data Acceptance or Discard** includes disposition for each data type/source combination included in **Data Types and Sources**. Possible dispositions are accept or discard. Within accept, this could include parse, aggregate and/or sample.
- **Parsed Data** is an individual parsed data object provided to the Aggregate Data function.

4.2.2.3.14 Provide Data to Subscribing System Users

This object receives blocks of same type/source packaged data with destination information attached. It also buffers data packages intended for the same destination and sends those as larger messages.

Associated Information Objects:

- **Repackaged, Addressed Data** is the packaged data on subscribed data types sent to data subscribers.
- **Reformatted Data w/Destinations** is the aggregated, parsed, sampled and repackaged data with a list of IP addresses that are to receive that data.

4.2.2.3.15 Provide Operator Interface to DD

This object provides an interface to the Operator, allowing him access to User Data Subscription, data acceptance and geographic broadcast functions. The Operator interface also allows control changes to the operational mode of Data Distribution functions.

Associated Information Objects:

- **Data Acceptance Details** is information describing what data (source/type, time, coverage area) the Core accepts, received from the Data Acceptance Catalog.
- **Operator ID, function** is the identity of the Operator attempting a data subscription or data acceptance modification, sent to the Check User Permission function to verify whether the System User is permitted the given **function**.
- **Permission** is received from the Check User Permissions function in response to the Operator ID, function message. This includes whether or not the Operator is permitted the **function** requested.

4.2.2.3.16 Receive Data from System Users

This object acquires data and geo-cast distribution requests from System Users. It queries the Check User Permissions function to determine if the System User is permitted to send the data or geo-cast that the System User provides. It only forwards data and geo-cast requests if those data and requests are permitted. Data and requests that are not permitted are noted as misbehavior reports.

Associated Information Objects:

- **Geo-cast Message** comes from System Users. It includes information describing the desired distribution area and time over which the distribution should be made. It is provided to the Manage Geo-cast Messages function if it passes the permission check.
- **Provided Data** comes from System Users and is distributed to the Parse Data function if it passes the permission check.
- **Provider ID, function** is the identity or type of the System User data provider (which could be anonymous), which is sent to the Check User Permission function to verify whether the System User is permitted a given **function**.

- **Permission** is the response from the Check User Permission function describing whether or not the **User** (System User, Core or Operator, depending on the original request) is permitted this action.
- **Data Acceptance Info** is received from the Data Acceptance Catalog and sent to a System User that provides data that the Core does not accept. It includes information describing a 3rd party facility that does accept the data, if such a facility exists. Includes the type of data to be provided, source type and location.
- **Data Acceptance Query** is a request for information concerning all the data types the Core accepts. Includes the type of data to be provided, source type and location.
- **Suspicious Data** is Provided Data that does not pass the permission check. It is provided to the Misbehavior Reports Log.
- **Suspicious Geo-Cast** is a Geo-Cast Message that does not pass the permission check. It is provided to the Misbehavior Reports Log.

4.2.2.3.17 Repackage Data

This object queries the Data Subscription Catalog to determine what data has subscribers. Of that data with subscribers, it strips source header from each message, groups messages according to type and provides those message groups to the Aggregate Data function.

Associated Information Objects:

- **Provided Data** comes from the Receive Data from System Users Function.
- **Repackaged Data** is composed of groups of same-type data, send to the Aggregate Data function.
- **Data Types and Sources** is sent to the Data Subscription Catalog. This includes the types of data and their source or source type.
- **Data Acceptance or Discard** includes disposition for each data type/source combination included in **Data Types and Sources**. Possible dispositions are accept or discard. Within accept, this could include parse, aggregate and/or sample.

4.2.2.3.18 Sample Data

This Optional Functional Object receives data from the Aggregate Data function, and samples messages or data elements as appropriate, before passing such sampled data to the Match Data to Data Subscribers function. If this function is not implemented, all messages simply pass through it to the Match Data to Data Subscribers function.

. Associated Information Objects:

- **Aggregated Data** is the aggregated parsed and repackaged data, received from the Aggregate Data function.
- **Data Ready to be Matched** is the aggregated, parsed, sampled and repackaged data, provided to the Match Data to Data Subscribers function.
- **Repackaged Data** is composed of groups of same-type data, received from the Repackage Data function.
- **Subscription Details** is received from the Data Subscription Catalog function. It includes the subscriber's contact information, the data it is subscribed to and details about aggregation and sampling that should be applied to that data.

4.2.2.4 View Description

This view addresses the functions required to implement, monitor and control the Core System's data distribution function, including geo-casting and publish/subscribe-based distribution of data. Establishment and maintenance of System User subscriptions are covered under a separate view, Functional View – User Configuration.

This view includes three optional functional objects: Parse Data, Sample Data and Aggregate Data. Data Distribution can still be implemented without these functions; they are classed as optional because their implementation requires substantial effort.

The system accepts data with the Receive Data from System Users function, and passes all data meeting permissions requirements to the Parse Data function. If the Parse Data function does not exist, data is passed to the Repackage Data function. Either way, data is made anonymous by one of these functions, either by having individual elements extracted by Parse Data or having the sender's header information removed by Repackage Data. Data that does not meet the permissions checks initiated by Receive Data from System Users is reported to the Misbehavior Reports Log for later analysis.

Data passes through the Parse Data, Repackage Data, Sample Data and/or Aggregate Data functions, depending on what is implemented by the Core and what subscriptions the Core has. Subscriptions dictate what amount of sampling and aggregation are required, so if no subscribers request those functions, then those functions are essentially pass-through.

Data passing through the publish/subscribe engine is passed through the following functions in order:

> Receive Data from System Users -> Parse Data / Repackage Data -> Aggregate Data -> Sample Data -> Match Data to Data Subscribers -> Provide Data to Subscribing System Users

Data is not modified under any circumstances. Data may be aggregated, in which case new information is created from existing data.

Data that is provided for a geo-cast that passes the permission check is passed through the following functions in order:

> Receive Data from System Users -> Manage Geo-Cast Messages -> Match Data to Data Subscribers -> Provide Data to Subscribing System Users

Geo-cast messages are held by the Manage Geo-Cast Messages function and sent when their message envelope indicates they should be sent. Geo-cast messages may be resent periodically, so this function may maintain a copy of the message and periodically resend it, until the message's send time is passed.

Note that geo-cast messages and other data, regardless of source, all enter the Core System at the same logical point. This is to enable flexibility with regard to the eventual distribution of data. It may also simplify the development of applications that use the Data Distribution features of the Core, as they will have to support only one interface. The exact specification of the interfaces provided by Receive Data from System Users and Provide Data from System Users will be a task for subsequent design. These interfaces should use commercially available protocols that can be accessed from all supported communications media (see the Communications Viewpoint).

Additional interfaces could always be added by adding a new input function, and then tying that into the Parse Data or Repackage Data function. Similarly, additional functionality could be added by inserting new functions into the data flows defined here. For instance, if it was decided that certain data should be subject to limit checking, a Limit Checking function could be added between Parse Data and Aggregate Data without substantially re-architecting the system, though related Connectivity and Information Views would have to be considered.

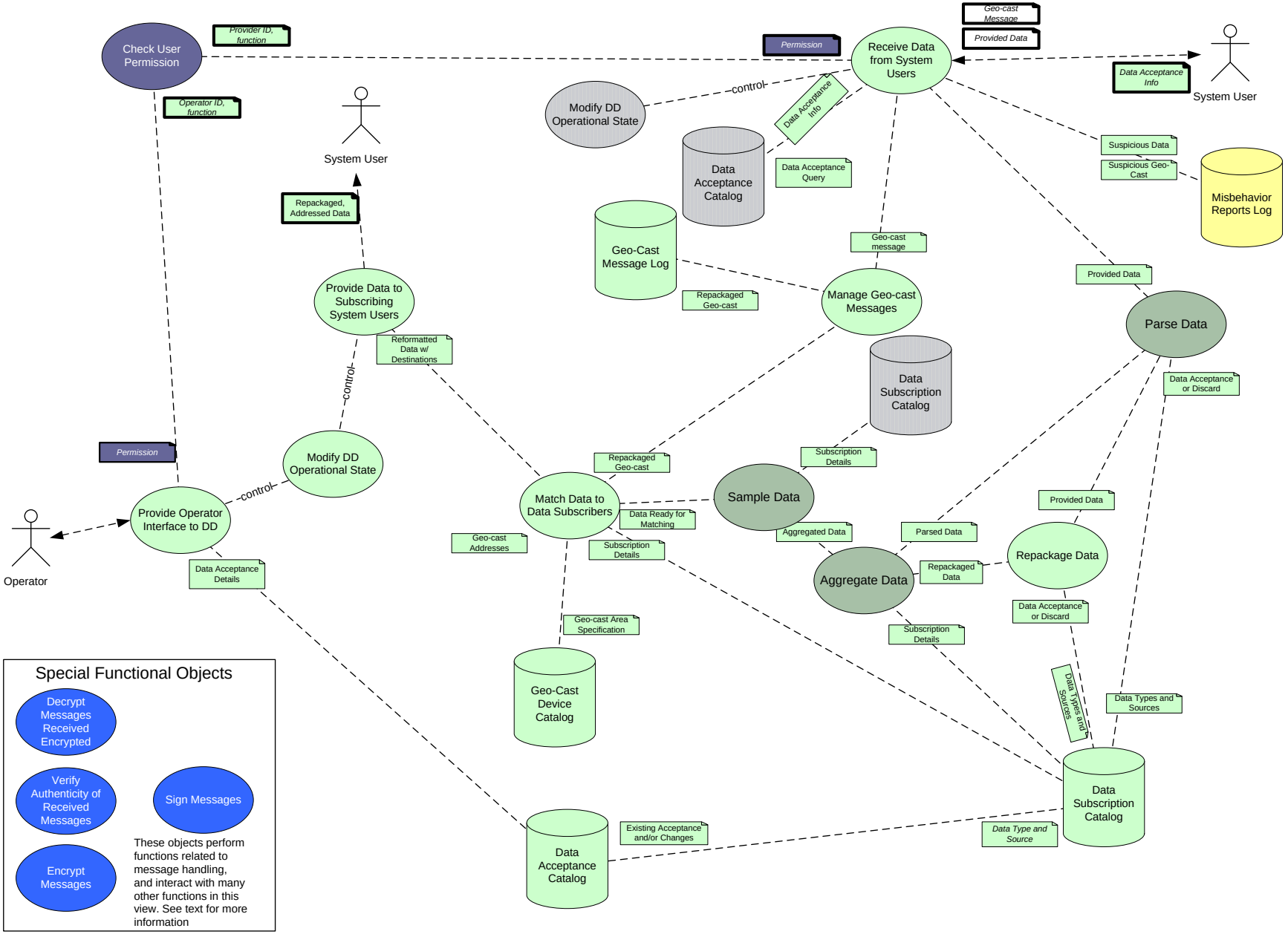


Figure 4-9: Functional View – Data Distribution

4.2.2.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Operations
Enterprise Viewpoint: Enterprise View – Governance
Functional Viewpoint: Functional View – Top Level
Functional Viewpoint: Functional View – System Configuration
Functional Viewpoint: Functional View – User Configuration
Functional Viewpoint: Functional View – Core Backup
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Information Viewpoint: Information View – Top Level External Objects
Information Viewpoint: Information View – Top Level Internal Objects

4.2.3 Functional View – System Configuration

4.2.3.1 Introduction

This view addresses the configuration of all Core subsystems, both for installation and changes to configuration.

4.2.3.2 Concerns Addressed by this View

Functionality	How does the Core System support the coordination of resources between different Cores? How does the Core System function internally? How do the Core System's components work together? How does the Core System transition between operational modes?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?
Evolvability	How easily can the Core's functionality be expanded to cover new needs if they arise? Does the functionality of the Core scale to support foreseeable demands from System Users?

4.2.3.3 Object Definitions and Roles

4.2.3.3.1 Actors

4.2.3.3.1.1 **ESS X.509 CA**

This is the External Support System that provides the Core System with its X.509 certificate.

4.2.3.3.1.2 **Operator**

The Operator is the day-to-day administrator of the Core System. The Operator interacts with the Core through various "Provide Operator Interface to…" Functional Objects.

4.2.3.3.1.3 **Other Core**

This actor represents another Core System.

4.2.3.3.2 Special Functional Objects

These special objects interact with various other functions to perform operations on messages sent or received. The last maintains some common information that is used by many other functions. They are not shown interacting on the view description diagram for the sake of clarity.

4.2.3.3.2.1 **Decrypt Messages Received Encrypted**

This function accepts an encrypted message intended for the Core System and decrypts it using the Core's private key.

4.2.3.3.2.2 **Encrypt Messages**

This function encrypts a message using the public key of the intended recipient.

4.2.3.3.2.3 **Sign Messages**

This function attaches the Core System's digital signature to a message by encrypting a hash of the message content with the Core's private key.

4.2.3.3.2.4 **Verify Authenticity of Received Messages**

This function verifies that the message is not a duplicate, has a valid digital signature, that it meets formatting rules and that its contents are within prescribed limits for the message type.

4.2.3.3.3 Check User Permission

This object accepts a System User ID or Operator ID, along with a type of operation that the user is attempting to access, and responds with whether or not the user is permitted that action.

Associated Information Objects:

- **User ID** is a unique representation of a user, sent to the User Permission Registry as a query to determine the user's permissions.
- **User Permissions** is the response from the User Permission Registry describing the permissions of the user associated with **User ID**.
- **Operator ID, function** is the identity of the operator which is received from the relevant subsystem Operator Interface function to verify whether the operator is permitted a given **function**.
- **Permission** is the response from the Check User Permission function describing whether or not the **User** (System User, Core or Operator, depending on the original request) is permitted this action.

4.2.3.3.4 Configure [Subsystem] Objects

These objects allow the operator to configure all subsystem functions. Configuration information is provided to the relevant subsystem Configuration store.

Includes:

- Configure Core2Core
- Configure Data Distribution
- Configure Misbehavior Management
- Configure Network Services
- Configure Service Monitor
- Configure Time Synchronization
- Configure User Permissions
- Configure User Trust Management

4.2.3.3.5 [Subsystem] Configuration / Generic Subsystem Configuration

These stores maintain configuration data for subsystem functions, including startup and operating parameters, and how the subsystem's functions change when in different states and modes. Configuration data is made available to all subsystem functions.

Includes:

- Core2Core Configuration
- Data Distribution Configuration

- Misbehavior Management Configuration
- Network Services Configuration
- Service Monitor Configuration
- Time Synchronization Configuration
- User Permissions Configuration
- User Trust Management Configuration

Generic Subsystem Configuration refers to all of the Configuration stores. It is drawn separately to simplify the diagram.

Associated Information Objects:

- **Core Config Info** is configuration data describing the services offered by the Core and the users it offers those services to. It is provided to the Identify Core Conflicts function.
- **Config Info for Other Cores** is configuration data describing the services offered by the Core and the users it offers those services to. It is provided to the Exchange Configuration Info with Other Cores function.

4.2.3.3.6 Exchange Configuration Info with Other Cores

The function exchanges service and scope information with other Cores. This includes the services offered by the Core, the geographic area over which those services are offered, and the types of System Users these services are offered to. Types include Field, Mobile, Center, and can be further specified as needed (e.g., Transit Buses).

Associated Information Objects:

- **Config Info for Other Cores** is configuration data describing the services offered by the Core and the users it offers those services to, obtained from the various subsystem Configuration stores.
- **Core Configuration Info** is the service and scope configuration information exchanged with other Cores.
- **Other Core Config Changes** are changes to other Core configurations, sent to the Other Core Configs store.

4.2.3.3.7 Generic [Subsystem] Component

This object represents all of the Functional Objects of the relevant subsystem. Each object receives configuration information from the relevant subsystem's configuration store. Each object receives configuration information from the subsystem's configuration store.

Includes:

- Generic C2C Component
- Generic DD Component
- Generic MM Component
- Generic NS Component
- Generic SM Component
- Generic TS Component
- Generic UP Component
- Generic UTM Component

4.2.3.3.8 Identify Core Conflicts

This function examines the configuration of other Cores and compares them to the configuration of its Core. It identifies conflicts and notifies the other Core and the Operator.

Associated Information Objects:

- **Core Config Info** is configuration data describing the services offered by the Core and the users it offers those services to, obtained from the various subsystem Configuration stores.
- **Core Conflict Info** is a description of the conflict in service provision between the two Cores. It is provided to the other Core that is in conflict with this Core.
- **Other Core Config Info** is configuration data describing the services offered by another Core and the users it offers those services to; this is obtained from the Other Core Configs store.

4.2.3.3.9 Maintain Core X.509 Certificate

This object obtains and stores the certificate the Core uses to garner trust with other Cores and System Users. It also obtains and stores certificate revocation lists from the ESS CA.

Associated Information Objects:

- **Ext X.509 Cert Request** is the request from the Core to the ESS X.509 CA to obtain a certificate.
- **X.509 Certificate** is the Core's certificate, provided by the ESS X.509 CA.
- **X.509 CRL** is the current Certificate Revocation List, provided regularly by the ESS X.509 CA.

4.2.3.3.10 Manually Modify Other Core Configs

The function allows the Operator to manually change the configuration data the Core maintains on other Cores.

Associated Information Objects:

- **Other Core Config Changes** includes changes to other Core configurations, sent to the Other Core Configs store.

4.2.3.3.11 Other Core Configs

This store maintains a listing of other Cores the Core System knows about, including their service offerings and the users they offer services to.

Associated Information Objects:

- **Other Core Config Changes** includes changes to other Core configurations, received from the Manually Modify Other Core Configs and Exchange Configuration Info with Other Cores functions.
- **Other Core Config Info** is configuration data describing the services offered by another Core and the users it offers those services to. It is provided to the Identify Core Conflicts function.

4.2.3.3.12 Provide Operator Interface to [Subsystem] / Generic Subsystem Operator Interface

These functions provide interfaces to the Operator, allowing him to change the configuration of subsystem functions. Each Provide Operator Interface function interacts with the Check User Permissions function, to verify that the Operator is permitted access to the function he is attempting to use.

The specific instances of an operator interface include:

- Provide Operator Interface to C2C
- Provide Operator Interface to DD
- Provide Operator Interface to MM
- Provide Operator Interface to NS
- Provide Operator Interface to SM
- Provide Operator Interface to TS
- Provide Operator Interface to UP
- Provide Operator Interface to UTM

Generic Subsystem Operator Interface refers to all of the operator interfaces. It is drawn separately to simplify the diagram.

4.2.3.3.13 Receive Core Conflict Info

This function receives information from another Core indicating that this Core and the reporting Core are in conflict over the provision of some service (e.g., Data Distribution). This function notifies the Operator of the conflict.

Associated Information Objects:

- **Core Conflict Info** is a description of the conflict in service provision between the two Cores. It is provided by the other Core that is in conflict with this Core.

4.2.3.3.14 User Permission Registry

This data store maintains a registry of System User, Operator and Other Core permissions. Allows update and creation of System Users, Operators and Other Cores and how they are allowed to interact with the Core System. This includes functions they may exercise and characteristics of their use such as frequency and source location (e.g., a given user may be permitted as a Center User but not as a Mobile User). This function maintains the certificate-managed application permissions that a user is permitted.

Associated Information Objects:

- **User ID** is a unique representation of a user received from the Check User Permission, Modify Application Permissions or Manually Modify User Permissions functions as a query to determine the user's permissions and/or characteristics
- **User Permissions** is the response to the Check User Permission or Modify Application Permissions functions describing the permissions of the user associated with **User ID**.

4.2.3.4 View Description

This view illustrates the functions that the Core includes to configure itself and maintain an understanding of the configuration of other Cores with which it has a relationship. Each subsystem has its own configuration storage and management functions. This maintains independence between subsystems, enables a variety of Core System configurations, and the addition of subsystems or new functionality to existing subsystems as needed.

The Core2Core subsystem exchanges configuration information with other Cores, and maintains an understanding of the configurations of those other Cores. This enables Cores to interoperate, forward requests for services that they do not provide but another Core does, and inform System Users of other Cores that provide services that they do not.

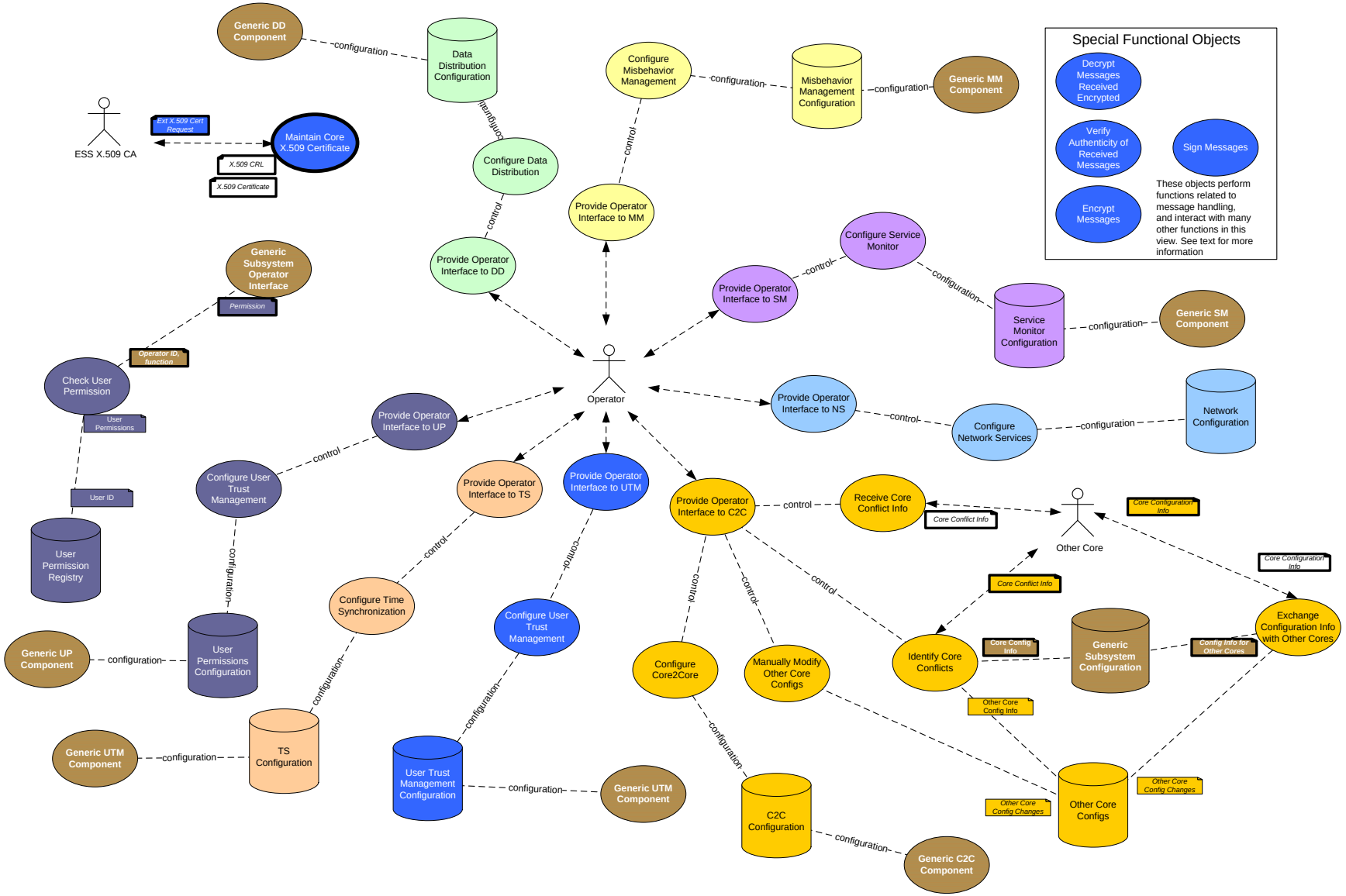


Figure 4-10: Functional View – System Configuration

4.2.3.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Operations
Enterprise Viewpoint: Enterprise View – Configuration and Maintenance
Enterprise Viewpoint: Enterprise View – Governance
Functional Viewpoint: Functional View – Top Level
Functional Viewpoint: Functional View – User Configuration
Functional Viewpoint: Functional View – System Monitor and Control
Functional Viewpoint: Functional View – Core Backup
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Information Viewpoint: Information View – Top Level External Objects
Information Viewpoint: Information View – Top Level Internal Objects

4.2.4 Functional View – User Configuration

4.2.4.1 Introduction

This view addresses the configuration of Core System users and their data subscriptions.

4.2.4.2 Concerns Addressed by this View

Interfaces	How flexible are Core System data distribution interfaces? How does the Core System enable control of the services it provides?
Functionality	How does the Core System monitor the services it provides? How does the Core System support the coordination of resources between different Cores? How does the Core System function internally? How do the Core System's components work together? How does the Core System transition between operational modes?
Security	What functional elements are involved in the detection of misbehavior by System Users, and what roles do those entities have? How does the Core System maintain the integrity of information provided to it by System Users? How does the Core System secure System Users' personal information?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?
Evolvability	How easily can the Core's functionality be expanded to cover new needs if they arise? Does the functionality of the Core scale to support foreseeable demands from System Users?

4.2.4.3 Object Definitions and Roles

4.2.4.3.1 Actors

4.2.4.3.1.1 End User Application Deployer

This is the entity that provides an application to the System User.

4.2.4.3.1.2 Field Node Owner/Operator

This is the owner/operator of Field Nodes used to perform geographic broadcast.

4.2.4.3.1.3 Operator

The Operator is the day-to-day administrator of the Core System. The Operator interacts with the Core through various "Provide Operator Interface to…" Functional Objects.

4.2.4.3.1.4 Other Core

This actor represents another Core System.

4.2.4.3.1.5 System User

This actor represents a System User, including Center, Mobile and Field.
This actor represents another Core System.

4.2.4.3.2 Special Functional Objects

These special objects interact with various other functions to perform operations on messages sent or received. The last maintains some common information that is used by many other functions. They are not shown interacting on the view description diagram for the sake of clarity.

4.2.4.3.2.1 Decrypt Messages Received Encrypted

This function accepts an encrypted message intended for the Core System and decrypts it using the Core's private key.

4.2.4.3.2.2 Encrypt Messages

This function encrypts a message using the public key of the intended recipient.

4.2.4.3.2.3 Sign Messages

This function attaches the Core System's digital signature to a message by encrypting a hash of the message content with the Core's private key.

4.2.4.3.2.4 Verify Authenticity of Received Messages

This function verifies that the message is not a duplicate, has a valid digital signature, that it meets formatting rules and that its contents are within prescribed limits for the message type.

4.2.4.3.3 Application Permission Registry

This object maintains the permissions for applications that use the permission fields in IEEE 1609.2 certificates.

Associated Information Objects:

- **Application Permission Changes** are changes required to the Application Permission Registry, received from Modify Application Permissions.

4.2.4.3.4 Check User Permission

This object accepts a System User ID or Operator ID, along with a type of operation that the user is attempting to access, and responds with whether or not the user is permitted that action.

Associated Information Objects:

- **Operator ID, function** is the identity of the operator which is received from the relevant subsystem Operator Interface function to verify whether the operator is permitted a given **function**.
- **Permission** is the response from the Check User Permission function describing whether or not the **User** (System User, Core or Operator, depending on the original request) is permitted this action.

- **Subscriber ID, function** is the identity of the System User that is sent to the Check User Permissions function to verify whether the System User is permitted a given **function**. In this case, the **function** is related to data subscription.
- **Core ID, function** is the identity of the Core that is sent to the Check User Permissions function to verify whether the Core is permitted a given **function**. In this case, the **function** is status update registration.
- **User ID** is a unique representation of a user, sent to the User Permission Registry as a query to determine the user's permissions.
- **User Permissions** is the response from the User Permission Registry describing the permissions of the user associated with **User ID**.

4.2.4.3.5 Configure Geo-cast Device Information

This object allows access to the Geo-Cast Device catalog.

Associated Information Objects:

- **Geo-cast Info** describes geo-cast supporting devices, including performance and operating characteristics, and the permissions System Users must have to access them.
- **Geo-cast Info Changes** describes changes to the geo-cast info of geo-cast supporting devices; this includes performance and operating characteristics, and the permissions System Users must have to access these devices.
- **Field Node Configuration Information** specifies the location, IP address, communications range, bandwidth and constraints for use of Field Node Infrastructure. Received from the Field Node Owner/Operator. Constraints on the use of Field Nodes may be imposed by the Field Node Owner/Operator and included as part of this Information Object.

4.2.4.3.6 Core Register to Receive Status

This object receives registration information from other Cores that request automatic updates of Core status.

Associated Information Objects:

- **Core Status Registration** is contact and preferences information describing the other Core, its contact information, desired update frequency and level of detail. This message is received from another Core and if it is allowed by the Check User Permissions function, forwarded as a new entry or (if the Core is already registered) modification of an existing entry in the Service Status Distribution Catalog.
- **Core ID, function** is the identity of the Core that is sent to the Check User Permissions function to verify whether the Core is permitted a given **function**. In this case, the **function** is status update registration.
- **Permission** is the response from the Check User Permission function describing whether or not the **User** (System User, Core or Operator, depending on the original request) is permitted this action.

4.2.4.3.7 Core Register to Receive Status from Other Core

This object registers the Core to receive status information from other Cores.

Associated Information Objects:

- **Core Status Registration** is contact and preferences information describing Core, its contact information, desired update frequency and level of detail. This message is provided to another Core.

4.2.4.3.8 Data Acceptance Catalog

This data store maintains a catalog of data types and sources that are used by the Core's Data Distribution subsystem. It also maintains a list of the data types it does not accept but some other facility does. It allows changes to the catalog to update existing and add new data types, sources and combinations of types and sources. This function responds to queries about data types telling whether or not that type/source combination is accepted for data distribution. It also analyzes its data distribution coverage area, time, and source/type combinations against those reported by other Cores to determine if there are any overlaps (where multiple Cores provide the same services in the same area) or conflicts (where Cores provide different and conflicting information for the same area, for instance if one Core says that a given data type must go to external sink A, but another Core says that it accepts that data type).

Associated Information Objects:

- **Data Acceptance Changes** are received from the Modify Data Acceptance Catalog function. This includes the types of data and their source or source type, and whether the Core should accept or not accept that data for distribution.
- **Data Acceptance Query** is a request for information about the data types accepted by the Core. Includes the type of data to be provided, source type and location, received from the Modify Data Acceptance Catalog function.
- **Data Acceptance Info** is a response to a Data Acceptance Query received from Manage Data Provision Requests. It indicates whether the data requested to be published is distributed by the Core, and if not includes information describing a 3rd party facility, either another Core or ESS, that does accept the data if such a facility exists. Includes the type of data to be provided, source type and location.
- **Data Type and Source Request** is a query sent to the Data Subscription Catalog to determine if a given data type/source combination is currently being used. This includes the types of data and their source or source type.
- **Existing Acceptance and/or Changes** is the response to the Data Type and Source Request query. It specifies whether the data type/source in question is already being distributed or if not, that such distribution has been added.

4.2.4.3.9 Data Subscription Catalog

This data store maintains a catalog of data types and sources and the System Users that want to receive that data. It may allow specification of aggregation period and sampling rate for each System User. Allowable sampling rates and aggregation specifications may be restricted based on configuration and performance of this subsystem. Allows changes to the catalog to update existing and add new subscribers and responds to queries about data subscribers and their subscriptions.

Associated Information Objects:

- **Data Type and Source Request** is a query received from the Data Acceptance Catalog to determine if a given data type/source combination is currently being used. This includes the types of data and their source or source type.
- **Existing Acceptance and/or Changes** is the response to the Data Type and Source Request query from the Data Acceptance Catalog. It specifies whether the data type/source in question is already being distributed or if not, that such distribution has been added.
- **Subscriber ID** is a query from the Manage System User Data Subscriptions function identifying a data subscriber, requesting the subscriber's existing data subscription information.

- **Data Subscription Details** is the query response to Subscriber ID or Data Subscription Changes sent to the Manage System User Data Subscriptions function. This includes the subscriber's SUID and complete data subscription information.
- **Data Subscription Changes** are the changes to a subscriber's subscription sent by the Manage System User Data Subscriptions function.

4.2.4.3.10 Generic Provide Operator Interface

This object provides a user interface for all subsystems to the Operator, allowing him to interact with them. This function interacts with many other monitoring, configuration and settings functions to enable the Operator to manage the Core. It provides a display of current operational status, access to historical performance and access to functions used to configure Core services.

Associated Information Objects:

- **Operator ID, function** is the identity of the operator which is sent to the Check User Permission function to verify whether the operator is permitted a given **function**.
- **Permission** is the response from the Check User Permission function describing whether or not the **User** (System User, Core or Operator, depending on the original request) is permitted this action.

4.2.4.3.11 Geo-cast Device Catalog

This data store keeps track of the information necessary for System Users to perform geo-casting. This includes locations, addresses, and ranges of devices that accept geo-cast messages.

Associated Information Objects:

- **Geo-cast Info Changes** describes changes to the geo-cast info of geo-cast supporting devices; this includes performance and operating characteristics, and the permissions System Users must have to access these devices. This message is received from the Configure Geo-Cast Device Information function.

4.2.4.3.12 Manage Data Provision Requests

This object accepts inquiries from System Users wishing to provide data. If the Core accepts data of the type the provider wishes to send, Core responds with an acknowledgement to send. If data is not accepted by the Core, but another Core is known to accept data of this type, that Core's address is provided. If a 3rd party (non-Core) accepts the data, information needed to contact that 3rd party is provided.

Associated Information Objects:

- **Data Provision Request** is a request from a System User that wants to provide data. Includes the type of data to be provided, source type and location.
- **Direct Data Distribution Info** is sent to the System User and includes the IP address and format expectations of the 3rd party that accepts data.
- **Data Acceptance Query** is a request for information concerning the data type the provider wishes to send to the Core. Includes the type of data to be provided, source type and location.
- **Data Acceptance Info** is an acknowledgement that the Core accepts the data type indicated in the message. Sent in response to a Data Provision Request.
- **Other Core Acceptance Info** includes the IP address of another Core that the Core believes will accept the data the provider wants to provide, received from the Data Acceptance Catalog.

- **Suspicious Data Provision Request** is a provision request that is either improperly formatted or asks to provide data that the requesting System User is not permitted to provide; this is provided to the Misbehavior Reports Log.

4.2.4.3.13 Manage System User Data Subscriptions

This object modifies existing or creates new System User data subscriptions. Accepts input from System Users requesting new or modified subscriptions. Also accepts Operator input for creation of new and modification of existing subscriptions. Queries Check User Permissions to ensure that the System User in question is permitted to subscribe to the data he asks for, and to ensure that the entity requesting the change (either System User or Operator) is permitted to do so.

Associated Information Objects:

- **Data Subscription Confirmation** is a response to the System User's subscription, describing the exact parameters of the System User's subscription.
- **Data Subscription Redirect Info** is a response to the System User's subscription, rejecting the subscription. If the Core knows of another Core that may be able to satisfy the subscription, this message includes the IP address of that Core.
- **Data Subscription Request** is a request for a data subscription from a System User. It includes the user's identification and the data types, sources and time of data collection. This may be a new subscription or modification of an existing subscription.
- **Data Subscription Details** is the query response to Data Subscription Changes received from the Data Subscription Catalog. This includes the subscriber's user ID and a complete description of the subscriber's data subscription.
- **Data Subscription Changes** is a request sent to the Data Subscriptions Catalog. It specifies a System User and changes (additions if new) to the user's subscription.
- **Permission** is the response from the Check User Permission function describing whether or not the **User** (System User, Core or Operator, depending on the original request) is permitted this action.
- **Subscriber ID, function** is the identity of the System User attempting to subscribe or have his subscription modified, sent to the Check User Permission function to verify whether the System User is permitted a given **function**. In this case the check would be to ensure that the user attempting the modification was allowed to make the subscription modification, and also to check that the subscriber was permitted to subscribe to the data it asks for.
- **Suspicious Data Subscription Request** is a subscription request that is either improperly formatted or asks for data that the requesting System User is not permitted to receive. This request is forwarded to the Misbehavior Results Log.

4.2.4.3.14 Manually Modify User Permissions

This object enables update and creation of System Users and Operators and how they are allowed to interact with the Core System. This includes functions they may exercise and characteristics of their use such as frequency and source location (e.g., a given user may be permitted as a Center User but not as a Mobile User). This function maintains the certificate-managed application permissions that a user is permitted. It is controlled by the Provide Operator Interface to UP function, included on the view diagram as part of Generic Provide Operator Interface.

Associated Information Objects:

- **User ID** is a unique representation of a user sent to the User Permission Registry as a query to determine the user's permissions and/or characteristics
- **User Info** describes the characteristics of a given **User** and is received from the User Permission Registry.
- **User Info Changes** are changes to a User's permissions sent to the User Permission Registry.

4.2.4.3.15 Misbehavior Reports Log

This object maintains the misbehavior reports received by the Core, including those submitted by System Users, and those from other Cores and those generated by internal Core monitoring processes. It accepts queries on those reports and provides the reports matching that query to requesting functions.

Associated Information Objects:

- **Suspicious Application Request** is either improperly formatted or asks for permissions that the requesting End User Application Deployer is not entitled to receive. This message is received from the Modify Application Permissions function.
- **Suspicious Data Provision Request** is a provision request that is either improperly formatted or asks to provide data that the requesting System User is not permitted to provide. This message is received from the Manage Data Provision Requests function.
- **Suspicious Data Subscription Request** is a subscription request that is either improperly formatted or asks for data that the requesting System User is not permitted to receive. This message is received from the Manage System User Data Subscriptions function.
- **Suspicious Permission Request** is either improperly formatted or asks for permissions that the requesting System User is not entitled to receive. This message is received from the Modify User Permissions function.

4.2.4.3.16 Modify Application Permissions

This object accepts requests to establish permissions for local field applications from End User Application Developers. If the End User Application Developer is permitted, this function creates or modifies and stores permissions in the Application Permission Registry.

Associated Information Objects:

- **Application Permission Changes** is the change, either creation or modification, of permissions associated with a particular application, sent to the Application Permission Registry.
- **Application Permission Request** is the request from the End User Application Developer to change existing permissions or grant new permissions.
- **Application Permission Confirmation** is submission of application permissions that must be managed using DSRC certificates.
- **User ID** is a unique representation of a user, sent to the User Permission Registry as a query to determine the user's permissions.
- **User Permissions** is the response from the User Permission Registry describing the permissions of the user associated with **User ID**.

4.2.4.3.17 Modify Data Acceptance Catalog

This object modifies the Data Acceptance Catalog, indicating what types/sources of data the Core accepts, and what types are handled directly by external 3rd parties.

Associated Information Objects:

- **Data Acceptance Query** is a query for information concerning other Core or 3rd party consumers of data; sent to the Data Acceptance Catalog. Includes the type of data to be provided, source type and location.
- **Data Acceptance Changes** sent to the Data Acceptance Catalog. This includes the types of data and their source or source type, and whether the Core should accept or not accept that data for distribution. Also allows specification of 3rd parties or other Cores that accept data.

4.2.4.3.18 Modify DD Operational State

This object is the enabling function that instructs various functions of Data Distribution to change the way they operate; this can include adding or deleting instantiations of an object or by commanding functions to enter into a different state or mode.

4.2.4.3.19 Modify UP Operational State

This object is a controlling function that instructs various User Permission functions to change the way they operate; this can include adding or deleting instantiations of an object or by commanding functions to enter into a different state or mode.

4.2.4.3.20 Modify User Permissions

This object enables a System User to register for Core System use. This function establishes the certificate-managed application permissions that a user is permitted. This function allows a System User to request additional permissions, which may be granted depending on User Permissions configuration maintained by the User Permission Registry.

Associated Information Objects:

- **Suspicious Application Request** is either improperly formatted or asks for permissions that the requesting End User Application Deployer is not entitled to receive. This message is provided to the Misbehavior Reports Log.
- **Suspicious Permission Request** is either improperly formatted or asks for permissions that the requesting System User is not entitled to receive. This message is provided to the Misbehavior Reports Log.
- **User Identity and Permission Request** is a request from a System User for a new account or modifications to his existing permission set. Includes the System User's identity or User ID (if already granted), and the permissions the System User desires.
- **User ID** is a unique representation of a user sent to the User Permission Registry as a query to determine the user's permissions and/or characteristics
- **User Info** describes the characteristics of a given **User** and is received from the User Permission Registry.
- **User Info Changes** are changes to a User's permissions sent to the User Permission Registry.

4.2.4.3.21 Other Core Configs

This store maintains a listing of other Cores the Core System knows about, including their service offerings and the users they offer services to.

Associated Information Objects:

- **Other Core Acceptance Info** includes the IP address of another Core that the Core believes will accept the data the provider wants to provide, provided to the Data Acceptance Catalog.

4.2.4.3.22 Service Status Distribution Catalog

This data store maintains a list of System Users and other Cores that are registered to receive Core status updates. This includes contact information, desired update frequency and allowed level of detail.

Associated Information Objects:

- **Core Status Registration** is a new entry in the catalog for a Core to receive status information. It includes the other Core's IP address, the services for which it requires status, the detail level of information to be provided and desired update frequency. This message is received from the Core Register to Receive Status function.
- **System User Status Registration** is a new entry in the catalog for a System User to receive status information. It includes the System User's IP address, the services for which it requires status and the detail level of information to be provided. This message is received from the System User Register to Receive Status function.

4.2.4.3.23 System User Register to Receive Status

This object receives registration information from System Users that request automatic updates of Core status.

Associated Information Objects:

- **System User Status Registration** is contact and preferences information describing the System User, its contact information, and desired update frequency. This message is received from a System User and if it is allowed by the Check User Permissions function, forwarded as an addition to the Service Status Distribution Catalog.
- **System User ID, function** is the identity of the System User that is sent to the Check User Permissions function to verify whether the System User is permitted a given **function**. In this case, the **function** is status update registration.
- **Permission** is the response from the Check User Permission function describing whether or not the **User** (System User, Core or Operator, depending on the original request) is permitted this action.

4.2.4.3.24 User Permission Registry

This data store maintains a registry of System User, Operator and Other Core permissions. Allows update and creation of System Users, Operators and Other Cores and how they are allowed to interact with the Core System. This includes functions they may exercise and characteristics of their use such as frequency and source location (e.g., a given user may be permitted as a Center User but not as a Mobile User). This function maintains the certificate-managed application permissions that a user is permitted.

Associated Information Objects:

- **User Info** describes the characteristics of a given **User** and is provided to the Manually Modify User Permissions and Modify User Permissions functions.

- **User Info Changes** are changes to a User's permissions received from the Manually Modify User Permissions function.
- **User ID** is a unique representation of a user received from the Check User Permission, Modify Application Permissions, Modify User Permissions and Manually Modify User Permissions functions as a query to determine the user's permissions and/or characteristics
- **User Permissions** is the response to the Check User Permission or Modify Application Permissions functions describing the permissions of the user associated with **User ID**.

4.2.4.4 View Description

This view addresses the functions required to configure and monitor user interactions with the Core System, including actions by Operators, System Users and other Cores. Most notably, this includes the creation and modification of System User data subscriptions. System User data that includes Personally Identifiable Information (PII), documents allowed actions with the Core or certificate-managed application permissions is all stored in encrypted form in the User Permission Registry. Since the registry maintains permissions on a per user basis for the entire Core, addition of new subsystems or functionality does not require substantial changes to the architecture, but only changes to fields within the registry.

Data subscription requests and provision requests may be the target of malicious activity, which is why both the Manage System User Data Subscriptions and Manage Data Provision Requests functions send suspicious requests to the Misbehavior Reports Log for later analysis.

The Manage Data Provision Requests function provides one of the interfaces that the Core System uses to coordinate activities between Cores. A System User seeking to provide data can find out which Core or 3rd party data sink takes the type of data they wish to provide. This function provides that service by accessing the Data Acceptance Catalog, and indirectly the Other Core Configs store.

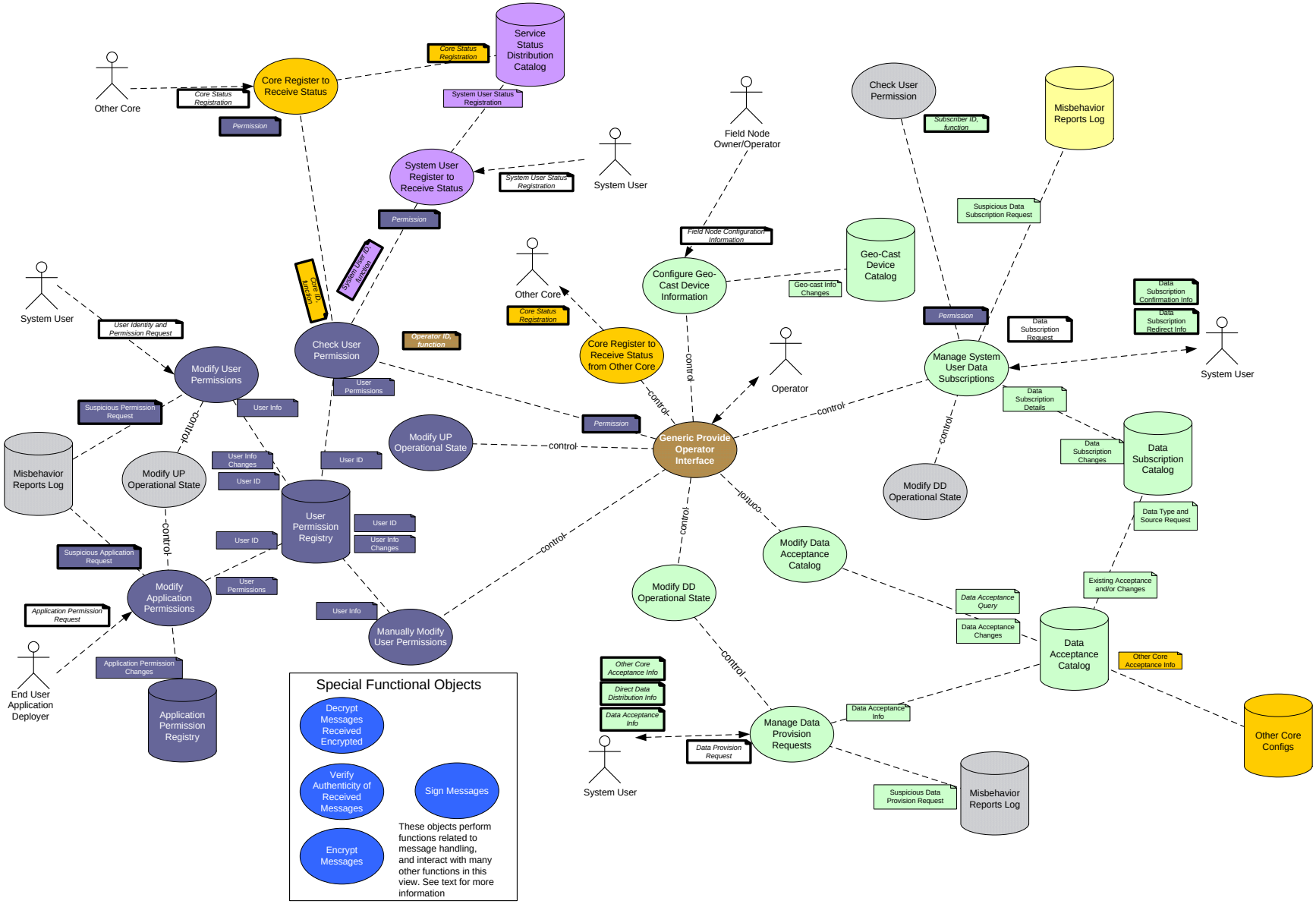


Figure 4-11: Functional View – User Configuration

4.2.4.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Operations
Enterprise Viewpoint: Enterprise View – Configuration and Maintenance
Enterprise Viewpoint: Enterprise View – Governance
Functional Viewpoint: Functional View – Top Level
Functional Viewpoint: Functional View – System Configuration
Functional Viewpoint: Functional View – System Monitor and Control
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Information Viewpoint: Information View – Top Level External Objects
Information Viewpoint: Information View – Top Level Internal Objects

4.2.5 Functional View – System Monitor and Control

4.2.5.1 Introduction

This view addresses most of the day-to-day Core System operations. It includes elements from Core2Core, Network Services, Service Monitor, Time Synchronization, User Trust Management, and User Permissions; only Data Distribution and Misbehavior Management are not included. While aspects of the six subsystems included here appear in other views, this view depicts the functionality these sub-systems provide to the Core.

Misbehavior Management includes detection of Operator misbehavior which is certainly related to this view. However, Operator misbehavior is not addressed here; it is addressed in Functional View – Credentials Distribution. That view addresses other forms of misbehavior. Several of the functions required to address other forms of misbehavior are the same as those needed to address Operator misbehavior. Keeping all of the misbehavior in the same view makes the diagrams a simpler.

4.2.5.2 Concerns Addressed by this View

Functionality	How does the Core System monitor the services it provides? How does the Core System function internally? How do the Core System's components work together? How does the Core System transition between operational modes?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?

4.2.5.3 Object Definitions and Roles

4.2.5.3.1 Actors

4.2.5.3.1.1 **Core Certification Authority**

This is the Core Certification Authority as defined in Enterprise View – Governance.

4.2.5.3.1.2 **Operator**

The Operator is the day-to-day administrator of the Core System. The Operator interacts with the Core through various "Provide Operator Interface to…" Functional Objects.

4.2.5.3.1.3 **Other Core**

This actor represents another Core System

4.2.5.3.1.4 **System User**

This actor represents a System User, including Center, Mobile and Field.

4.2.5.3.1.5 Time Source

This actor represents the external time source the Core uses as the basis for time.

4.2.5.3.2 Special Functional Objects

These special objects interact with various other functions to perform operations on messages sent or received. The last maintains some common information that is used by many other functions. They are not shown interacting on the view description diagram for the sake of clarity.

4.2.5.3.2.1 Decrypt Messages Received Encrypted

This function accepts an encrypted message intended for the Core System and decrypts it using the Core's private key.

4.2.5.3.2.2 Encrypt Messages

This function encrypts a message using the public key of the intended recipient.

4.2.5.3.2.3 Sign Messages

This function attaches the Core System's digital signature to a message by encrypting a hash of the message content with the Core's private key.

4.2.5.3.2.4 Verify Authenticity of Received Messages

This function verifies that the message is not a duplicate, has a valid digital signature, that it meets formatting rules and that its contents are within prescribed limits for the message type.

4.2.5.3.3 Check User Permission

This object accepts an Operator ID, along with a type of operation that the operator is attempting to access, and responds with whether or not the user is permitted that action.

Associated Information Objects:

- **User ID** is a unique representation of a user, sent to the User Permission Registry as a query to determine the user's permissions.
- **User Permissions** is the response from the User Permission Registry describing the permissions of the user associated with **User ID**.
- **Operator ID, function** is the identity of the operator which is received from the Provide Operator Interface function to verify whether the operator is permitted a given **function**.
- **Permission** is the response from the Check User Permission function describing whether or not the **User** (System User, Core or Operator, depending on the original request) is permitted this action.

4.2.5.3.4 Event Log

This data store receives activity, state change and anomaly records from all Core processes. It records all activities, state changes and anomalies and provides them to the Provide Operator Interface function.

- **Anomalies** are records of process behavior that is outside the norm for the given process, received from any function and sent to the Provide Operator Interface function.
- **State Changes** are records of a change in state or mode of the given process, received from any function and sent to the Provide Operator Interface function.
- **Actions** are records of process behavior, including System User interaction, provided by all processes as a record of Core activities.

4.2.5.3.5 Generic Modify Subsystem Operational State

This object is the enabling function that instructs various functions to change the way they operate; this can include adding or deleting instantiations of an object or by commanding functions to enter into a different state or mode. Includes:

- Modify C2C Operational State
- Modify DD Operational State
- Modify MM Operational State
- Modify NS Operational State
- Modify SM Operational State
- Modify TS Operational State
- Modify UP Operational State
- Modify UTM Operational State

Generic Modify Subsystem Operational State refers to all of the subsystem state modification functions. It is drawn separately to simplify the diagram.

4.2.5.3.6 Generic Provide Operator Interface

This object provides a user interface for all subsystems to the Operator, allowing him to interact with them. This function interacts with many other monitoring, configuration and settings functions to enable the Operator to manage the Core. It provides a display of current operational status, access to historical performance and access to functions used to configure Core services.

Associated Information Objects:

- **Anomalies** are records of process behavior that is outside the norm for the given process, received from the Event Log.
- **State Changes** are records of a change in state or mode of the given process, received from the Event Log.
- **Other Core Status** is received from the Monitor Status of Other Cores function and describes the status of another Core's services including operating state and mode.
- **Environmental Response** is a description of the recommended or automatically imposed action this function suggests or takes in response to physical or system health issues; received from the Take Action in Response to Environmental Issue function.
- **Health and Safety Status** is the status of the integrity of operating Core software and hardware received from the Monitor Core Health and Safety function.
- **Detailed Service Status** is detailed information describing the performance of all Core functions and interfaces, received from the Monitor Core Services Performance function.

4.2.5.3.7 Generic Service Component

This object is representative of any other Functional Object.

Associated Information Objects:

- **Actions** are records of process behavior, including System User interaction, provided to the Event Log as a record of Core activities.
- **Anomalies** are records of process behavior that is outside the norm for the given process, provided to the Event Log.
- **State Changes** are records of a change in state or mode of the given process, provided to the Event Log.

- **Time Local Form** is time synchronized with the external source in a format usable by Core functions.

4.2.5.3.8 Get Time from External Available Source

This object acquires time from an external Stratum-2 source and provides it in a usable form to the Make Time Available to All Subsystems function. If commanded to enter maintenance mode, this function will cease attempting to acquire time from its external source.

Associated Information Objects:

- **Time Local Form** is time synchronized with the external source in a format usable by Core functions.
- **Time** is time acquired from the external Stratum-2 source.

4.2.5.3.9 Make Time Available to All Subsystems

This object provides the time acquired externally to all Core System components.

Associated Information Objects:

- **Time Local Form** is time synchronized with the external source in Universal Time Synchronization, Coordinated (UTC) time format.

4.2.5.3.10 Modify SM Operational State

This object is a controlling function that instructs various Service Monitor functions to change the way they operate; this can include adding or deleting instantiations of an object or by commanding functions to enter into a different state or mode.

4.2.5.3.11 Modify TS Operational State

This object is a controlling function that instructs various Time Synchronization functions to change the way they operate; this can include adding or deleting instantiations of an object or by commanding functions to enter into a different state or mode.

4.2.5.3.12 Monitor Core Health and Safety

This object monitors the integrity of Core System components. If a component fails an integrity check or is determined to be out of date or otherwise compromised, it reports that information to other functions including Generic Provide Operator Interface.

Associated Information Objects:

- **Health and Safety Status** is the status of the integrity of operating Core software and hardware.

4.2.5.3.13 Monitor Core Services Performance

This object monitors the performance of all Core System functions and interfaces. This includes measures of throughput, buffer levels, and resource usage. This information is provided to the Log System State and Performance and Provide Operator Interface functions; summary information describing the state of Core services is provided to the Provide Service Status to System Users and Provide Service Status to other Cores functions.

Associated Information Objects:

- **Detailed Service Status** is detailed information describing the performance of all Core functions and interfaces provided to the Log System State, Provide Service Status to Other Cores and Performance and Generic Provide Operator Interface functions.

- **Service Status** is summary information describing the state of Core services provided to the Provide Service Status to System Users and Provide Service Status to other Cores functions.
- **Service Status for Cores** is the status of the Core's subsystems, including state, mode, and performance data, sent to other Cores.

4.2.5.3.14 Monitor Status of other Cores

This object queries other Cores to determine the status of their services.

Associated Information Objects:

- **Core Service Status Query** is the query sent to another Core to determine the status of that Core's services.
- **Service Status for Cores** is received from another Core; it describes the status of that Core's services including operating state and mode. This message is also passed to the Provide Operator Interface function for presentation to the Operator.
- **Other Core Status** describes the status of another Core and is provided to the Operator through the Generic Provide Operator Interface function.

4.2.5.3.15 Provide Record of System Performance

This function obtains records of system performance from the System State and Performance Log, and provides that information to the Core Certification Authority.

Associated Information Objects:

- **Performance Records** is detailed information describing the long-term performance of Core services, provided to the Core Certification Authority.

4.2.5.3.16 Provide Service Status to other Cores

This object provides the status of Core services on a subsystem basis to other Cores. This information is distributed upon request, and also periodically to Cores that have registered to receive it.

Associated Information Objects:

- **Core Distribution List** is a list of Cores that have registered to receive automatic service status updates. This includes their IP address, the services they are to receive information about, the detail level of information they are to receive and their desired update rate.
- **Core Service Status Query** is the query sent to another Core to determine the status of that Core's services.
- **Detailed Service Status** is detailed information describing the performance of all Core functions and interfaces received from the Monitor Core Services Performance function.
- **Service Status for Cores** is the status of the Core's subsystems, including state, mode, and performance data, sent to other Cores. This message is derived from the Detailed Service Status message; the amount of information present in this message is dependent on the other Core's entry in the Core Distribution List. If this message is sent in response to a Core Service Status Query the message is not modified by Core Distribution List entries; it is less detailed.

4.2.5.3.17 Provide Service Status to System Users

This object provides the status of Core services on a subsystem basis to System Users and ends that status according to the update frequency associated with that System User.

Associated Information Objects:

- **Distribution List** is a list of System Users that are registered to receive automatic status updates. This includes contact information, type of System User, and update frequency.
- **Service Status Query** is a request for Core service status information from a System User.
- **Service Status** is the status of the Core's subsystems, including state and mode, received from Monitor Core Services Performance and sent to System Users.
- **Operational Changes** describe how the Functional Object needs to alter operations. If in a normal mode, this will include restrictions on operations. If already restricted, this will be a change to restrictions, including possibly removing all restrictions and returning to normal mode.

4.2.5.3.18 Service Status Distribution Catalog

This data store maintains a list of System Users and other Cores that are registered to receive Core status updates. This includes contact information, desired update frequency and allowed level of detail.

Associated Information Objects:

- **Distribution List** is a list of System Users that are registered to receive automatic status updates. This includes contact information, type of System User, and update frequency.
- **Core Distribution List** is a list of other Cores that are registered to receive automatic status updates. This includes contact information, what services and how much detail to provide, and update frequency.

4.2.5.3.19 System State and Performance Log

This store accepts Core performance, monitoring and configuration changes, and logs that information for subsequent analysis performed by External Support Systems.

Associated Information Objects:

- **Detailed Service Status** is detailed information describing the performance of all Core functions and interfaces, received from the Monitor Core Services Performance function.
- **Health and Safety Status** is the status of the integrity of operating Core software and hardware, received from the Monitor Core Health and Safety function.

4.2.5.3.20 Take Action in Response to Environment Issue

This object reacts to changes in the Core's physical environment, including power, temperature, humidity or physical intrusions noted by Monitor Core Health and Safety. Depending on the issue and Core configuration, it may either recommend action to the Operator or automatically initiate a change of state.

Associated Information Objects:

- **Environmental Response** is a description of the recommended or automatically imposed action this function suggests or takes in response to physical or system health issues.
- **Health and Safety Status** is the status of the integrity of operating Core software and hardware, received from the Monitor Core Health and Safety function.

4.2.5.3.21 User Permission Registry

This data store maintains a registry of System User, Operator and Other Core permissions. Allows update and creation of System Users, Operators and Other Cores and how they are allowed to interact with the Core System. This includes functions they may exercise and characteristics of their use such as fre-

quency and source location (e.g., a given user may be permitted as a Center User but not as a Mobile User). This function maintains the certificate-managed application permissions that a user is permitted.

Associated Information Objects:

- **User ID** is a unique representation of a user received from the Check User Permission, Modify Application Permissions or Manually Modify User Permissions functions as a query to determine the user's permissions and/or characteristics
- **User Permissions** is the response to the Check User Permission or Modify Application Permissions functions describing the permissions of the user associated with **User ID**.

4.2.5.4 View Description

This view identifies day-to-day housekeeping functions that enable the Operator to manage the Core System's operations. It includes monitoring of subsystem anomalies and state changes, subsystem and interface performance information and distribution of that performance information to qualified users.

Each subsystem includes a function that provides the user interface to that function for the Operator. Service Monitor includes additional functions that monitor the environmental conditions the Core's hardware operates in, monitor the performance of Core functions, and monitor the integrity of Core components and functions. All of this information is stored and made available to the Operator. Some of this information is shared with System Users and other Cores.

Each subsystem includes a function that allows the Operator (or another Functional Object) to control the state and mode of a Functional Object. This Modify Operational State function allow additional instantiations of objects to be created, and allows different instantiations to be placed in different modes. This supports maintenance actions, where a given instantiation may be set to operate while another is being modified. Note that according to Connectivity View – State and Mode Transitions, modes are based on Engineering Objects. A given Engineering Object may implement more than one Functional Object. A mode that applies to one Engineering Object therefore applies to multiple Functional Objects. The effects of the various modes may be different across the spectrum of Engineering Objects.

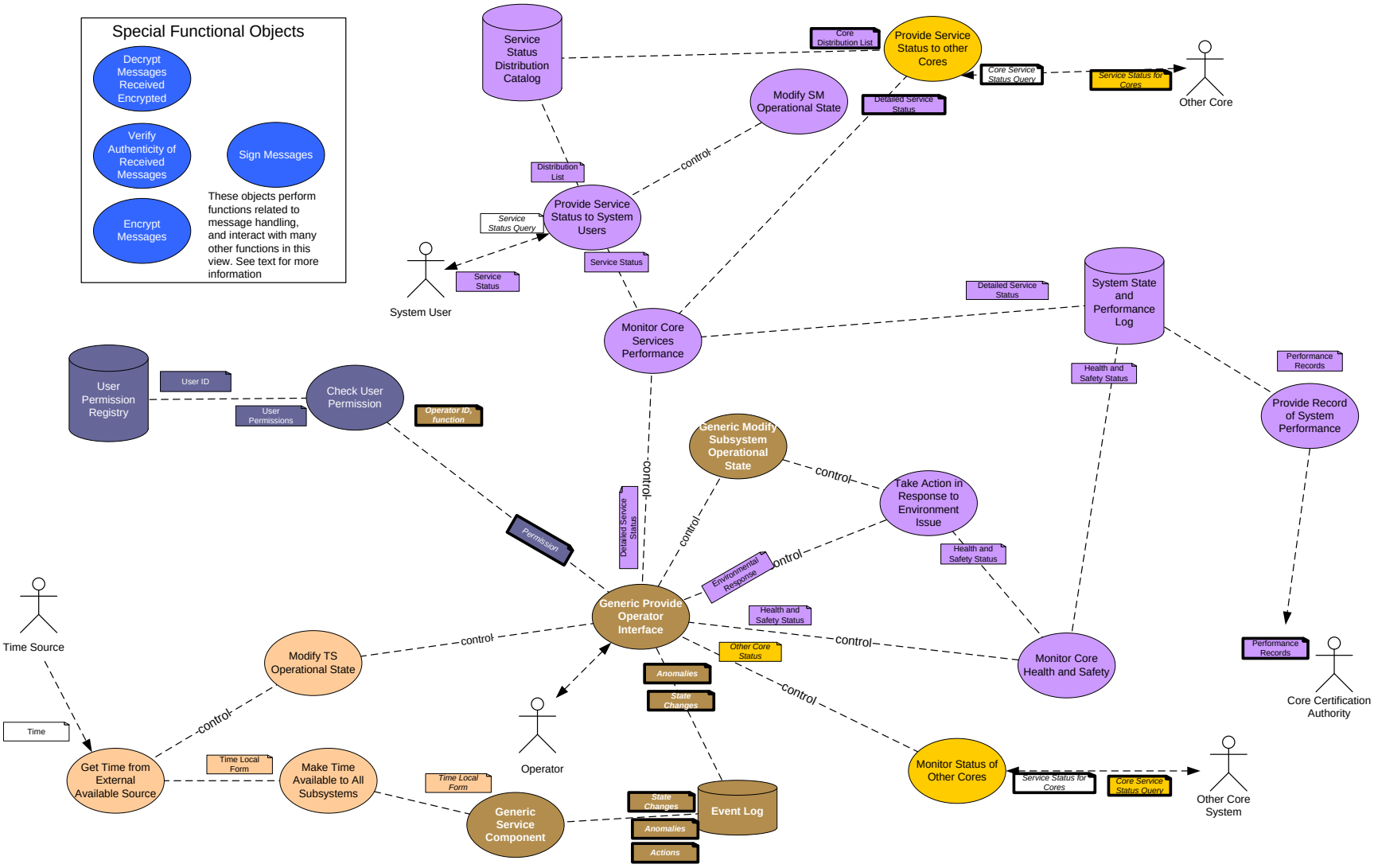


Figure 4-12: Functional View – System Monitor and Control

4.2.5.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Operations
Enterprise Viewpoint: Enterprise View – Configuration and Maintenance
Enterprise Viewpoint: Enterprise View – Governance
Functional Viewpoint: Functional View – Top Level
Functional Viewpoint: Functional View – System Configuration
Functional Viewpoint: Functional View – User Configuration
Functional Viewpoint: Functional View – Core Backup
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Information Viewpoint: Information View – Top Level External Objects
Information Viewpoint: Information View – Top Level Internal Objects

4.2.6 Functional View – Credentials Distribution

4.2.6.1 Introduction

Management of credentials, including digital certificates, certificate revocation lists and assignment and recognition of credential-related roles (i.e., registration versus certificate distribution) is one of the primary roles for the Core System. The Core needs to ensure trust between System Users and with other Cores; it does this by distributing credentials, by working with external credential providers and sharing credential information with other Cores and external credential providers.

This view illustrates the Core's role in the distribution of digital certificates. Note that the Core relies on an external CAs to distribute digital certificates to Mobile Users; this view only addresses X.509 certificates required by Field and Center users.

4.2.6.2 Concerns Addressed by this View

Functionality	How does the Core System monitor the services it provides? How does the Core System support the coordination of resources between different Cores? How does the Core System function internally? How do the Core System's components work together? How does the Core System transition between operational modes?
Security	What functional elements are involved in the distribution and revocation of digital certificates, and what roles do those entities have? How does the Core System maintain the integrity of information provided to it by System Users? How does the Core System maintain the privacy of communications between System Users? How does the Core System secure System Users' personal information?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?
Evolvability	Does the functionality of the Core scale to support foreseeable demands from System Users?

4.2.6.3 Object Definitions and Roles

4.2.6.3.1 Actors

4.2.6.3.1.1 ESS DSRC RA

This actor represents the External Support System functioning as a registration authority for IEEE 1609.2 certificate distribution.

4.2.6.3.1.2 System User

This actor represents a System User, including Center, Mobile and Field.

4.2.6.3.2 Special Functional Objects

These special objects interact with various other functions to perform operations on messages sent or received. They are not shown interacting on the view description diagram for the sake of clarity.

4.2.6.3.2.1 Decrypt Messages Received Encrypted

This function accepts an encrypted message intended for the Core System and decrypts it using the Core's private key.

4.2.6.3.2.2 Encrypt Messages

This function encrypts a message using the public key of the intended recipient.

4.2.6.3.2.3 Sign Messages

This function attaches the Core System's digital signature to a message by encrypting a hash of the message content with the Core's private key.

4.2.6.3.2.4 Verify Authenticity of Received Messages

This function verifies that the message is not a duplicate, has a valid digital signature, that it meets formatting rules and that its contents are within prescribed limits for the message type.

4.2.6.3.3 Application Permission Registry

This data store maintains the permissions for applications that use the permission fields in DSRC certificates.

Associated Information Objects:

- **Cert Owner, App** is a query from the Maintain Credentials function asking what the application permissions are for a given application.
- **App Permissions** contains the application permissions for the application queried in Cert Owner, App, sent to the Maintain Credentials function.
- **Special App Permissions** is sent to the Provide Special Permissions function. It includes the user ID received from the User Permission Registry and the application permissions the user is entitled to.
- **User Apps** is received from the User Permission Registry. It includes the User ID of a user or class of users and a specification of what applications that user should have access to.

4.2.6.3.4 Check User Permission

This object accepts an identifier (ID), either a System User Identifier (SUID), or Operator ID, along with a type of operation that the user is attempting to access, and responds with whether or not the user is

permitted that action. With regard to certificate distribution User Permissions maintains a listing of what each user is permitted to do; consequently it provides the information necessary to establish the permissions section of an identity certificate.

Associated Information Objects:

- **Certificate Owner ID** is received from Provide Credentials as a query to determine whether the System User is allowed to request a given set of credentials, and what permissions should be attached to those credentials.
- **Cert Permission** is the response sent to Provide Credentials indicating whether the System User is allowed to request a given set of credentials and if so the types of permissions that must be attached to the digital certificate.
- **User ID, function** is a query sent to the User Permission Registry to determine if the user associated with the ID is permitted access to the given *function*.
- **User Perm, function** is the response from the User Permission Registry, stating whether the *user* is permitted access to the previously identified *function*.

4.2.6.3.5 CRL Distribution List

This store maintains the list of System Users that are registered to receive CRLs from the Core, and what CRLs they are registered to receive. It periodically sends a list of System Users and the CRL types they should receive to the Distribute CRL function. It also maintains a record of CRL transmissions; System User CRL distributions may be halted for a specific System User after a pre-defined number of failures.

Associated Information Objects:

- **CRL Distribution List** is sent to the Distribute CRL function. It includes a list of recipients and the CRL types that each should receive.
- **CRL Distribution Failure** is received from the Distribute CRL function when the intended recipient of a CRL message does not receive the message.
- **CRL Distribution Registration** is received from the System User Register to Receive CRL function, and includes the ID and IP Address of a System User that requests to receive CRLs from the Core.

4.2.6.3.6 CRL Storage

This object maintains Certificate Revocation Lists for all certificates, including both CRLs maintained by this Core and CRLs maintained by other Cores or ESS and received by this Core. It responds to queries about associations between System User IDs with entries in the CRL.

Associated Information Objects:

- **CRL** is a complete Certificate Revocation List. Several different CRL messages may be created, one for each CRL maintained by the Core and one for each CRL received from an ESS or other Core.
- **System User ID** is received from Provide Credentials and includes the identity of a System User.
- **System User CRL Associations** is a response to the Provide Credentials function that indicates the certificate IDs that are on the CRL and are associated with the System User ID previously provided.

4.2.6.3.7 Distribute CRL

This object distributes the Certificate Revocation Lists to System Users. It determines who to send CRLs to and which CRLs to provide them based on input from the CRL Distribution List store.

Associated Information Objects:

- **CRL** is the list of active certificates that are no longer valid, received from CRL Storage and provided to System Users.
- **CRL Distribution List** is received from the CRL Distribution List store. It includes a list of recipients and the CRL types that each should receive.
- **CRL Distribution Failure** is sent to CRL Distribution List when the intended recipient of a CRL message does not receive the message.
- **Scope Query** is a query sent to User Trust Management Configuration requesting the operational scope of User Trust Management.
- **Scope** is the response to Scope Query describing the operational scope of User Trust Management.

4.2.6.3.8 Maintain Credentials

This object maintains the inventory of digital certificates distributed by the Core including the special permissions associated with these certificates. It provides these certificates upon valid request.

Associated Information Objects:

- **App Permissions** contains the application permissions for the application queried in Cert Owner, App, received from the Application Permission Registry function.
- **Cert Owner, App** is a query sent to the Application Permission Registry asking what the application permissions are for a given application.
- **Credentials** contains a new digital certificate for the System User, sent to the Provide Credentials function.
- **Managed Credential Request** is a request for a digital certificate for the System User, received from the Provide Credentials function. It includes any the types of permissions that must be attached to identity certificates as provided by Check User Permissions.

4.2.6.3.9 Modify User Permissions

This function modifies the permissions associated with a given user.

Associated Information Objects

- **Permission Change Request** is received from Provide Credentials. It describes the permissions that need to be modified for a specific user.
- **User Info Changes** is a request from Modify User Permissions to change the permissions associated with a specific user.

4.2.6.3.10 Modify UTM Operational State

This object is a controlling function that instructs various User Trust Management functions to change the way they operate; this can include adding or deleting instantiations of an object or by commanding functions to enter into a different state or mode.

4.2.6.3.11 Provide Credentials

This object accepts requests for credentials from System Users, and services valid requests by either:

1) Obtaining the appropriate certificates from the Maintain Credentials function and passing them along to the System User
2) Providing the System User with the IP Address of the External Support System that will service their request

The System User's location and request will dictate which option this function uses. If the System User is requesting IEEE 1609.2 certificates, this function will query Credential Source Configuration for the IP Address of the ESS that provides IEEE 1609.2 certificates for the System User's type and location. If the System User is requesting other credentials (e.g., X.509 certificates), this function will query the User Trust Management Configuration function to determine the proper source for those credentials; it could be another ESS, this Core or another Core.

This function also queries Check User Permissions to verify that the System User is allowed to request new certificates, and checks the System Users existing credentials against entries in the CRL. Lastly, it may request modification to the Core's record of the user's permissions.

Associated Information Objects:

- **Certificate Owner ID** is sent to Check User Permissions as a query to determine whether the System User is allowed to request the credentials it is requesting, and what permissions should be attached to those credentials.
- **Cert Permission** is the response from Check User Permissions indicating whether the System User is allowed to request the credentials it is requesting, and also the types of permissions that must be attached to digital certificates that the Core provides to this System User.
- **Credential Request and User** is a query received sent to User Trust Management Configuration to determine whether the Core can provide the credentials requested by the System User.
- **Credential Source** is a response to Credential Request and User indicating if the Core can provide the requested credentials, and if not, the IP address of the credential provider.
- **Credential Referral** contains the IP address of another Core or ESS that provides the type of credentials the System User is requesting.
- **Credential Request** is a request for credentials from a System User.
- **Credentials** contains a new digital certificate for the System User, received from the Maintain Credentials function and sent to the System User.
- **Managed Credential Request** is a request for a digital certificate for the System User, sent to the Maintain Credentials function. It includes any the types of permissions that must be attached to identity certificates as provided by Check User Permissions.
- **Operational Changes** describe how this function needs to alter operations. If in normal mode, this will include restrictions on operations. If already restricted, this will be a change to restrictions, including possibly removing all restrictions and returning to normal mode.
- **Permission Change Request** is sent to Modify User Permissions. It describes the permissions that need to be modified for a specific user.
- **Scope** is the response to Scope Query describing the operational scope of User Trust Management.

- **Scope Query** is a query sent to User Trust Management Configuration concerning the operational scope of User Trust Management.
- **System User ID** is a query sent to CRL Storage that includes the identity of a System User, asking for any certificate IDs associated with this user that are on the CRL.
- **System User CRL Associations** is a response from the CRL Storage function that indicates the certificate IDs that are on the CRL and are associated with the System User ID previously provided.

4.2.6.3.12 Provide Operator Interface to UTM

This object provides an interface to the Operator allowing him access to User Trust Management functions.

4.2.6.3.13 Provide Special Permissions

This object provides special application permission information to the ESS DSRC RA. It accepts a request from the ESS DSRC RA for special permission information for a specific user or user class. This function then passes that request to the User Permission Registry, which forwards to the Application Permission Registry a listing of the applications the user is permitted. This function then receives the appropriate permissions from the Application Permission Registry, and provides that information to the ESS DSRC CA.

Associated Information Objects:

- **Special App Permissions** is received from the Application Permissions Registry. It includes the originally provided user ID and the application permissions the user is entitled to.
- **User Identification** is the request for special permissions from the ESS DSRC RA. It includes an identification of the user or user class the RA wishes to acquire permissions for.
- **User Special Permissions** is the response to the ESS DSRC RA. It includes all of the special permissions the user is entitled to.

4.2.6.3.14 System User Register to Receive CRL

This object receives requests from System Users to receive CRLs. It passes those requests to the CRL Distribution List store if the requests are valid and distribution of CRLs to the requesting System User is within the operational scope of the Core.

Associated Information Objects:

- **CRL Distribution Registration** is received from the System User Register to Receive CRL function, and includes the ID and IP Address of a System User that requests to receive CRLs from the Core.
- **CRL Request** is a request from a System User for that user to receive CRLs from the Core.
- **Scope** is the response to Scope Query describing the operational scope of User Trust Management.
- **Scope Query** is a query sent to User Trust Management Configuration concerning the operational scope of User Trust Management.

4.2.6.3.15 User Permission Registry

This data store maintains a registry of System User, Operator and Other Core permissions. Allows update and creation of System Users, Operators and Other Cores and how they are allowed to interact with the Core System. This includes functions they may exercise and characteristics of their use such as frequency and source location (e.g., a given user may be permitted as a Center User but not as a Mobile User). This function maintains the certificate-managed application permissions that a user is permitted. However, certificates granted by non-Core authorities will not always include information describing the permissions the user is entitled to when interacting with the Core, so the Core needs its own records associating user and permissions.

Associated Information Objects:

- **User Apps** is sent to the Application Permission Registry. It includes the User ID of a user or class of users and a specification of what applications that user should have access to.
- **User ID, Apps** is received from Provide Special Permissions. It includes the User ID of a user or class of users a request to determine the applications the user is entitled to special permissions to use.
- **User Info Changes** is a request from Modify User Permissions to change the permissions associated with a specific user.
- **User ID, function** is a query received from Check User Permissions to determine if the user associated with the ID is permitted access to the given *function*.
- **User Perm, function** is the response to Check User Permissions, stating whether the *user* is permitted access to the previously identified *function*.

4.2.6.3.16 User Trust Management Configuration

This data store maintains the configuration of User Trust Management services and includes geographic scope and content of services offered. It also maintains the list of ESS and other Cores that provide credentials and distribute CRLs. This store responds to queries from Operator-controlled and other functions that require understanding of operational scope.

Associated Information Objects:

- **Credential Request and User** is a query received from Provide Credentials to determine whether the Core can provide the credentials requested by the System User.
- **Credential Source** is a response to Credential Request and User indicating if the Core can provide the requested credentials, and if not, the IP address of the credential provider.
- **Scope** is the response to Scope Query describing the operational scope of User Trust Management.
- **Scope Query** is a query received from a CRL or certificate distribution-related function concerning the operational scope of User Trust Management.

4.2.6.4 View Description

The Core System may act as Certificate Authority for users requiring digital certificates, except for those requiring IEEE 1609.2 certificates. Alternative architectures were developed where the Core would distribute IEEE 1609.2 certificates; these are documented in chapter 6 along with the rationale for why they are not in the architecture.

System Users requiring credentials interact with the Core System's Provide Credentials object. This object either works to obtain credentials for the System User, or if the Core does not provide the type of

credentials the user requires (e.g. IEEE 1609.2 certificates) provides them the IP address of another entity that does (e.g., an ESS DSRC RA).

The Core grants X.509 certificates to those users that require them. Typically this will be Center and Field Users, including RSE. The Core also distributes certificate revocation lists for the certificates that it grants.

The Core interacts with the ESS DSRC RA to provide information about special application permissions managed by the IEEE 1609.2 certificate that may be coordinated by the Core. It maintains these application permissions in the Application Permission Registry, and maintains the list of users that can use these applications in the User Permission Registry.

The Core stores CRLs and any personal information provided by System Users in a secure data store.

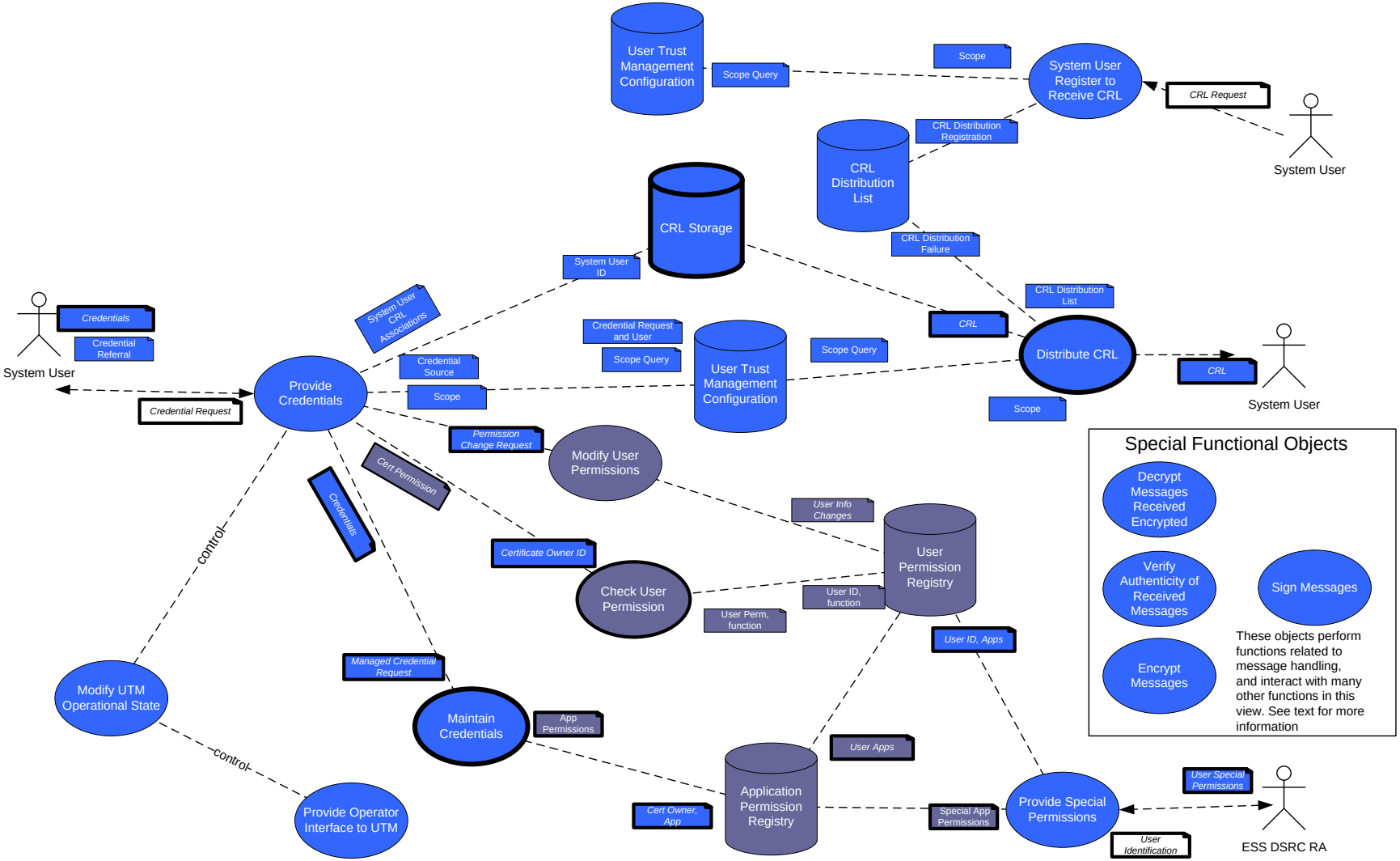


Figure 4-13: Functional View – Credential Management

4.2.6.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Security Credentials Distribution
Enterprise Viewpoint: Enterprise View – Governance
Enterprise Viewpoint: Enterprise View – Business Model Facilitation
Functional Viewpoint: Functional View – Functional View – Top Level
Functional Viewpoint: Functional View – Misbehavior Management
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Information Viewpoint: Information View – Top Level External Objects
Information Viewpoint: Information View – Top Level Internal Objects

4.2.7 Functional View – Misbehavior Management

4.2.7.1 Introduction

In order to ensure trust between System Users, the Core needs to identify System Users that are not acting properly. The Core then needs to act to ensure that the actions of this misbehaving System Users do not negatively affect other System Users. The Core also needs to ensure its own integrity and the integrity of the information it stores and passes. This requires that the Core monitor Operators, and identify and act on when they operate in such a way as to jeopardize the Core or the information it passes and stores.

This view illustrates the Core's role in the detecting and reacting to the identification of System User and Operator misbehavior.

4.2.7.2 Concerns Addressed by this View

Interfaces	How does the Core System enable control of the services it provides?
Functionality	How does the Core System monitor the services it provides? How does the Core System support the coordination of resources between different Cores? How does the Core System function internally? How do the Core System's components work together? How does the Core System transition between operational modes?
Security	What functional elements are involved in the distribution and revocation of digital certificates, and what roles do those entities have? What functional elements are involved in the detection of misbehavior by System Users, and what roles do those entities have?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?

4.2.7.3 Object Definitions and Roles

4.2.7.3.1 Actors

4.2.7.3.1.1 **External CA**

The External CA actor represents External Support Systems that function as Certificate Authorities. This includes the ESS DSRC Certificate Authority and ESS X.509 Certificate Authority.

4.2.7.3.1.2 **Other Core**

This actor represents another Core System

4.2.7.3.1.3 System User

This actor represents a System User, including Center, Mobile and Field.

4.2.7.3.2 Special Functional Objects

These special objects interact with various other functions to perform operations on messages sent or received. They are not shown interacting on the view description diagram for the sake of clarity.

4.2.7.3.2.1 Decrypt Messages Received Encrypted

This function accepts an encrypted message intended for the Core System and decrypts it using the Core's private key.

4.2.7.3.2.2 Encrypt Messages

This function encrypts a message using the public key of the intended recipient.

4.2.7.3.2.3 Sign Messages

This function attaches the Core System's digital signature to a message by encrypting a hash of the message content with the Core's private key.

4.2.7.3.2.4 Verify Authenticity of Received Messages

This function verifies that the message is not a duplicate, has a valid digital signature, that it meets formatting rules and that its contents are within prescribed limits for the message type.

4.2.7.3.3 Ask CA to Revoke

This object accepts receives misbehaving user IDs, determines if this Core manages the certificates associated with that user ID, and if not determines who does manage those certificates and sends a request to that ESS or Core to revoke the certificates of the misbehaving user.

Associated Information Objects:

- **Credential Source** is the response to User Type, indicating the responsible entity: either this Core, another Core, or an ESS.
- **Misbehaving User ID** contains the ID and type of misbehavior the System User has committed. Depending on the method of identification, the contents could include a user ID, a certificate ID, or a pseudo-ID. This message is received from the identify Misbehaving System Users function, and sent to the External CA.
- **User Type** is a query sent to the User Trust Management Configuration store, asking who is able to revoke the certificate for the given user type.

4.2.7.3.4 Check User Permission

This object accepts a SUID or Operator ID, along with a type of operation that the user is attempting to access, and responds with whether or not the user is permitted that action. With regard to certificate distribution User Permissions maintains a listing of what each user is permitted to do; consequently it provides the information necessary to establish the permissions section of an identity certificate.

Associated Information Objects:

- **Internal Misbehavior Report** is a report of misbehavior sent to the Receive Internal Misbehavior Reports function. It includes an identification of the reporting function, ID or pseudo-ID of the misbehaving user, and characterization of the type of misbehavior.
- **Operator ID, function** is the identity of the operator sent to the Check User Permission function to verify whether the operator is permitted a given **function**.

- **Operator Permission** is the response to Identify Misbehaving Operators or Revoke Operator Permissions, describing the Operator's current permissions.
- **SUID, Function Permission** is a response sent to Identify Misbehaving System Users indicating whether or not the System User is permitted to use the **function**.
- **System User ID, function** is a request received from Identify Misbehaving System Users to determine whether the System User is permitted to carry out **function**.
- **User ID** is a unique representation of a user sent to the User Permission Registry as a query to determine the user's permissions and/or characteristics
- **User Info** describes the characteristics of a given **User** and is received from the User Permission Registry.

4.2.7.3.5 CRL Storage

This object maintains Certificate Revocation Lists for all certificates, including both CRLs maintained by this Core and CRLs maintained by other Cores or ESS and received by this Core. It responds to queries about associations between System User IDs with entries in the CRL.

Associated Information Objects:

- **Complete CRL** contains the IDs of all certificates the sending Core has revoked. This is sent to or received from the Exchange CRLs with other Cores function.
- **CRL Changes** contains changes to the CRL received from the Manage CRL function. These changes are used to update the CRL maintained by this Core.
- **CRL Deltas** describes the changes to the CRL since the last CRL Delta or Complete CRL was sent or received.
- **External CRL** is a Certificate Revocation List received from an external CA through the Obtain CRLs function.

4.2.7.3.6 Disable Misbehaving Geo-Cast Device

This object receives a message from Identify Misbehaving System Users indicating that a specified geo-cast device needs to be disabled so that further geo-cast messages do not use the device. This function interacts with the Geo-Cast Device Catalog to disable the device.

Associated Information Objects

- **Geo-cast Info Changes** describes changes to the geo-cast info of geo-cast supporting devices; this includes performance and operating characteristics, and the permissions System Users must have to access these devices. This message is sent to the Geo-Cast Device Catalog store.
- **Misbehaving Geo-Cast Device** describes a geo-cast device that must be removed from use. It is received from the Identify Misbehaving System Users function.

4.2.7.3.7 Exchange CRLs with other Cores

This object allows the Core to exchange CRLs with other Cores. This allows each Core to maintain a complete CRL, so that System Users need receive a CRL from only one source.

Associated Information Objects:

- **Complete CRL** contains the IDs of all certificates the sending Core has revoked. This is sent by the Core or received from another Core. The sending Core obtains the Complete CRL from CRL Storage. Received CRLs are provided to CRL Storage.

- **CRL Deltas** describes the changes to the CRL since the last CRL Delta or Complete CRL was sent to the Core that is receiving this message. The sending Core obtains CRL Deltas from CRL Storage. The receiving Core forwards CRL changes to its own CRL Storage.

4.2.7.3.8 Exchange Misbehavior Reports with other Cores

Cores need to share misbehavior reports to establish a System User's pattern of behavior. Establishing a pattern of misbehavior provides the Core with sufficient reason to withdraw System User permissions. This Functional Object allows Cores to share misbehavior reports.

Associated Information Objects:

- **C2C Misbehavior Report** contains a certificate ID, time, and type of misbehavior identified, sent to and received from other Cores.
- **Misbehavior Report** is a misbehavior report received from the Misbehavior Reports Log
- **Operational Changes** describe how this function needs to alter operations. If in normal mode, this will include restrictions on operations. If already restricted, this will be a change to restrictions including possibly removing all restrictions and returning to normal mode.
- **Other Core Misbehavior Report** is a misbehavior report sent to the Misbehavior Reports Log, originally received as part of C2C Misbehavior Report.

4.2.7.3.9 Geo-cast Device Catalog

This data store keeps track of the information necessary for System Users to perform geo-casting. This includes locations, addresses, and ranges of devices that accept geo-cast messages. It accepts messages to configure, change and in this case disable access to geo-cast devices.

Associated Information Objects:

- **Geo-cast Info Changes** describes changes to the geo-cast info of geo-cast supporting devices; this includes performance and operating characteristics, and the permissions System Users must have to access these devices. This message is received from the Disable Misbehaving Geo-cast Device function.

4.2.7.3.10 Identify Misbehaving Operators

This object monitors Operator actions and determines whether those actions constitute misbehavior.

Associated Information Objects:

- **Operator ID, misbehavior** is sent to the Revoke Operator Permissions function, sent when the Identify Misbehaving Operators function identifies misbehavior sufficient to justify revoking Operator permissions. The message includes the Operator's ID and an indication of the types of permissions to revoke.
- **Operator ID, function** is the identity of the operator sent to the Check User Permission function to verify whether the operator is permitted a given **function**.
- **Operator Permission** is the response from the Check User Permission function describing whether or not the **User** (System User, Core or Operator, depending on the original request) is permitted this action.
- **Misbehaving Operator Alert** is sent to the Provide Operator Interface to Misbehavior Management (MM) function to inform other Operators that an Operator may be engaged in misbehavior.

4.2.7.3.11 Identify Misbehaving System Users

This object examines System User misbehavior reports, and identifies when users are malfunctioning or appearing to act maliciously. It also accepts input from the Provide Operation Interface to MM, and thus from the Operator, of a misbehaving System User and his misbehavior. This allows the Operator to manually identify misbehaving users.

Associated Information Objects:

- **Misbehaving Geo-Cast Device** describes a geo-cast device that must be removed from use. It is sent to the Geo-cast device catalog function.
- **Misbehavior Reports for Analysis** is received from the Misbehavior Reports Log; they contain the misbehavior reports requested by Misbehavior Reports Request.
- **Misbehavior Report Request** is a request for misbehavior reports sent to the Misbehavior Reports Log. It includes a set of criteria for report selection including time interval, spatial restriction, System User type or even specific System User.
- **Misbehaving User Alert** is sent to the Provide Operator Interface to MM function to inform Operators that a System User may be engaged in misbehavior. The message indicates the type of misbehavior and type and/or identity of the System User as applicable.
- **System User ID, function** is a request sent to Check User Permissions to determine whether the System User is permitted to carry out **function**.
- **SUID, Function Permission** is a response from Check User Permissions indicating whether or not the System User is permitted to use the **function**.
- **Misbehaving User ID** contains the ID and type of misbehavior the System User has committed. Depending on the method of identification, the contents could include a user ID, a certificate ID or a pseudo-ID. This message is provided to the Manage CRL or Ask CA to Revoke functions, depending on the scope of the Core. It is also sent to the Revoke User Permissions function, so that any Core permissions that user has may be revoked.
- **Scope** describes the operational scope of User Trust Management. It is received from the User Trust Management Configuration store.

4.2.7.3.12 Manage CRL

This object manages additions to the CRL associated with certificates granted by the Core. It receives identity of misbehaving users and revokes their certificates.

Associated Information Objects:

- **Misbehaving User ID** contains the ID and type of misbehavior the System User has committed. Depending on the method of identification, the contents could include a user ID, a certificate ID or a pseudo-ID. This message is received from the Identify Misbehaving System Users function.
- **CRL Changes** are changes to the CRL sent to CRL Storage, based on the Misbehaving User ID.

4.2.7.3.13 Misbehavior Reports Log

This object maintains the misbehavior reports received by the Core, including those submitted by System Users, and those from other Cores and those generated by internal Core monitoring processes. It accepts queries on those reports and provides the reports matching that query to requesting functions.

Associated Information Objects:

- **System User Misbehavior Reports** is a report of misbehavior from the Receive System User Misbehavior Reports function. It includes an identification of the reporting System User, ID or pseudo-ID of the misbehaving user, and characterization of the type of misbehavior.
- **Internal Misbehavior Reports** is a report of misbehavior from the Receive Internal Misbehavior Reports function. It includes an identification of the reporting function, ID or pseudo-ID of the misbehaving user, and characterization of the type of misbehavior.
- **Misbehavior Report Request** is a request for misbehavior reports from the Identify Misbehaving System Users function. It includes a set of criteria for report selection including time interval, spatial restriction, System User type or even specific System User.
- **Misbehavior Reports for Analysis** is sent to the Identify Misbehaving System Users function contains the misbehavior reports requested by that function.
- **Misbehavior Report** is a misbehavior report sent to the Exchange Misbehavior Reports with other Cores function.
- **Other Core Misbehavior Report** is a misbehavior report received from the Exchange Misbehavior Reports with other Cores function.

4.2.7.3.14 Modify C2C Operational State

This object is a controlling function that instructs various Core2Core functions to change the way they operate; this can include adding or deleting instantiations of an object or by commanding functions to enter into a different state or mode.

4.2.7.3.15 Modify MM Operational State

This object is a controlling function that instructs various Misbehavior Management functions to change the way they operate; this can include adding or deleting instantiations of an object or by commanding functions to enter into a different state or mode.

4.2.7.3.16 Modify UTM Operational State

This object is a controlling function that instructs various User Trust Management functions to change the way they operate; this can include adding or deleting instantiations of an object or by commanding functions to enter into a different state or mode.

4.2.7.3.17 Modify User Permissions

This object enables a System User to register for Core System use. This function establishes the certificate-managed application permissions that a user is permitted. This function allows a System User to request additional permissions, which may be granted depending on User Permissions configuration maintained by the User Permission Registry.

Associated Information Objects:

- **Operator Permission** is a response to Operator ID, Revoke function sent to the Revoke Operator Permissions function.
- **Operator ID, Revoke function** is a request received from Revoke Operator Permissions to revoke the Operator's ability to carry out the specified **function**.
- **System User Permission** is a response from Modify User Permissions identifying the System User's new permissions.

- **System User ID, Revoke function** is a response to the System User ID, Revoke function sent to the Revoke Operator Permissions function.
- **User Info Changes** are changes to a User's permissions sent to the User Permission Registry.

4.2.7.3.18 Obtain CRLs

This function acquires CRLs from ESS providing CA functions.

Associated Information Objects:

- **External CRL** is a Certificate Revocation List received from and provided to CRL Storage.
- **CRL Sources** is received from the User Trust Management Configuration store, and indicates the sources for CRLs.

4.2.7.3.19 Provide Operator Interface to C2C

This object provides an interface to the Operator allowing him access to Core2Core functions.

4.2.7.3.20 Provide Operator Interface to MM

This object provides an interface to the Operator, allowing him access to Misbehavior Management functions.

Associated Information Objects:

- **Misbehaving Operator Alert** is received from the Identify Misbehaving Operator function to inform other Operators that an Operator may be engaged in misbehavior.
- **Misbehaving User Alert** is received from the Identify Misbehaving System Users function to inform the Operator of a System User identified as misbehaving.

4.2.7.3.21 Receive Internal Misbehavior Reports

This object accepts misbehavior reports from other functions within the Core. It validates reports and passes them to the Misbehavior Reports Log.

Associated Information Objects:

- **Internal Misbehavior Report** is a report of misbehavior received from the Check User Permissions function and sent to the Misbehavior Reports Log. It includes an identification of the reporting function, ID or pseudo-ID of the misbehaving user, and characterization of the type of misbehavior.
- **Decryption Error Message** describes a decryption failure and includes a copy of the encrypted message.
- **Authenticity Error Message** describes an authenticity verification failure, and includes a copy of the message that failed authenticity verification.

4.2.7.3.22 Receive System User Misbehavior Reports

This object accepts a misbehavior reports from System Users describing alleged misbehavior of other System Users. It validates reports and passes them to the Misbehavior Reports Log.

Associated Information Objects:

- **System User Misbehavior Report** is a report of misbehavior received from System Users and subsequently sent to the Misbehavior Reports Log. It includes an identification of

the reporting System User, ID or pseudo-ID of the misbehaving user, and characterization of the type of misbehavior.

- **Operational Changes** describe how this function needs to alter operations. If in normal mode, this will include restrictions on operations. If already restricted, this will be a change to restrictions, including possibly removing all restrictions and returning to normal mode.

4.2.7.3.23 Revoke Operator Permissions

This object accepts an Operator ID and an identification of misbehavior the Operator is engaged in and subsequently revokes his permissions through the Check User Permissions function.

Associated Information Objects:

- **Operator ID, misbehavior** is received from the Identify Misbehaving Operators function. The message includes the Operator's ID and an indication of the types of permissions to revoke.
- **Operator ID Revoke function** is a request sent to Modify User Permissions to revoke the Operator's permissions to carry out the indicated function.
- **Operator Permission** is a response from Modify User Permissions identifying the Operator's new permissions.

4.2.7.3.24 Revoke User Permissions

This object accepts a System User ID and an identification of misbehavior the System User is engaged in and subsequently revokes his permissions through the Check User Permissions function.

Associated Information Objects:

- **System User ID, misbehavior** is received from the Identify Misbehaving System Users function. The message includes the System User ID, and an indication of the types of permissions to revoke.
- **System User ID, function** is a request sent to Check User Permissions to revoke the System User's permissions to carry out the indicated function.
- **System User Permission** is a response from Modify User Permissions identifying the System User's new permissions.

4.2.7.3.25 User Permission Registry

This data store maintains a registry of System User, Operator, and Other Core permissions. It allows for the update and creation of System Users, Operators and Other Cores and how they are allowed to interact with the Core System. This includes functions they may exercise and characteristics of their use such as frequency and source location (e.g., a given user may be permitted as a Center User but not as a Mobile User). This function maintains the certificate-managed application permissions that a user is permitted.

Associated Information Objects:

- **User Info** describes the characteristics of a given **User** and is provided to the Manually Modify User Permissions and Modify User Permissions functions.
- **User Info Changes** are changes to a User's permissions received from the Modify User Permissions function.
- **User ID** is a unique representation of a user received from the Check User Permission, Modify Application Permissions, Modify User Permissions and Manually Modify User

Permissions functions as a query to determine the user's permissions and/or characteristics

4.2.7.3.26 User Trust Management Configuration

This data store maintains the configuration of User Trust Management services and includes geographic scope and content of services offered. It also maintains the list of ESS and other Cores that provide credentials and distribute CRLs. This store responds to queries from Operator-controlled and other functions that require understanding of operational scope.

Associated Information Objects:

- **Credential Source** is the response to User Type, indicating the responsible entity: either this Core, another Core or an ESS.
- **CRL Sources** is sent to the Obtain CRLs function, and indicates the sources for CRLs.
- **Scope** describes the operational scope of User Trust Management. It is provided to the Identify Misbehaving System Users function.
- **User Type** is a query from the Ask CA to Revoke function, asking who is able to revoke the certificate for the given user type.

4.2.7.4 View Description

This view is concerned exclusively with the detection of misbehavior by System Users and Operators.

The Misbehavior Reports Log is the repository of all information used as the basis for misbehavior analysis. It is populated with misbehavior reports from functions in this view (e.g. Exchange Misbehavior Reports with other Cores or Receive Internal Misbehavior Reports) and functions in other views (e.g. Manage Data Provision Requests).

Misbehavior analysis is performed by the Identify Misbehaving System Users and Identify Misbehaving System Operators functions. It consists of examining actions, looking for patterns and determining if an entity (either System User or Operator) is behaving in a manner that warrants action by the Core System. Once that determination is made, the appropriate function will be notified. For Operators this means withdrawing permissions managed by User Permissions. For System Users, this may include withdrawing permissions managed by User Permissions, disabling a geo-cast device, revoking a certificate issued by the Core, or asking another Core or external CA to revoke a certificate.

Misbehavior Management actions can be monitored and controlled by the Operator from a user interface. Control of misbehavior identification criteria is handled by the Misbehavior Management Configuration store that holds the subsystem configuration, documented in Functional View – System Configuration.

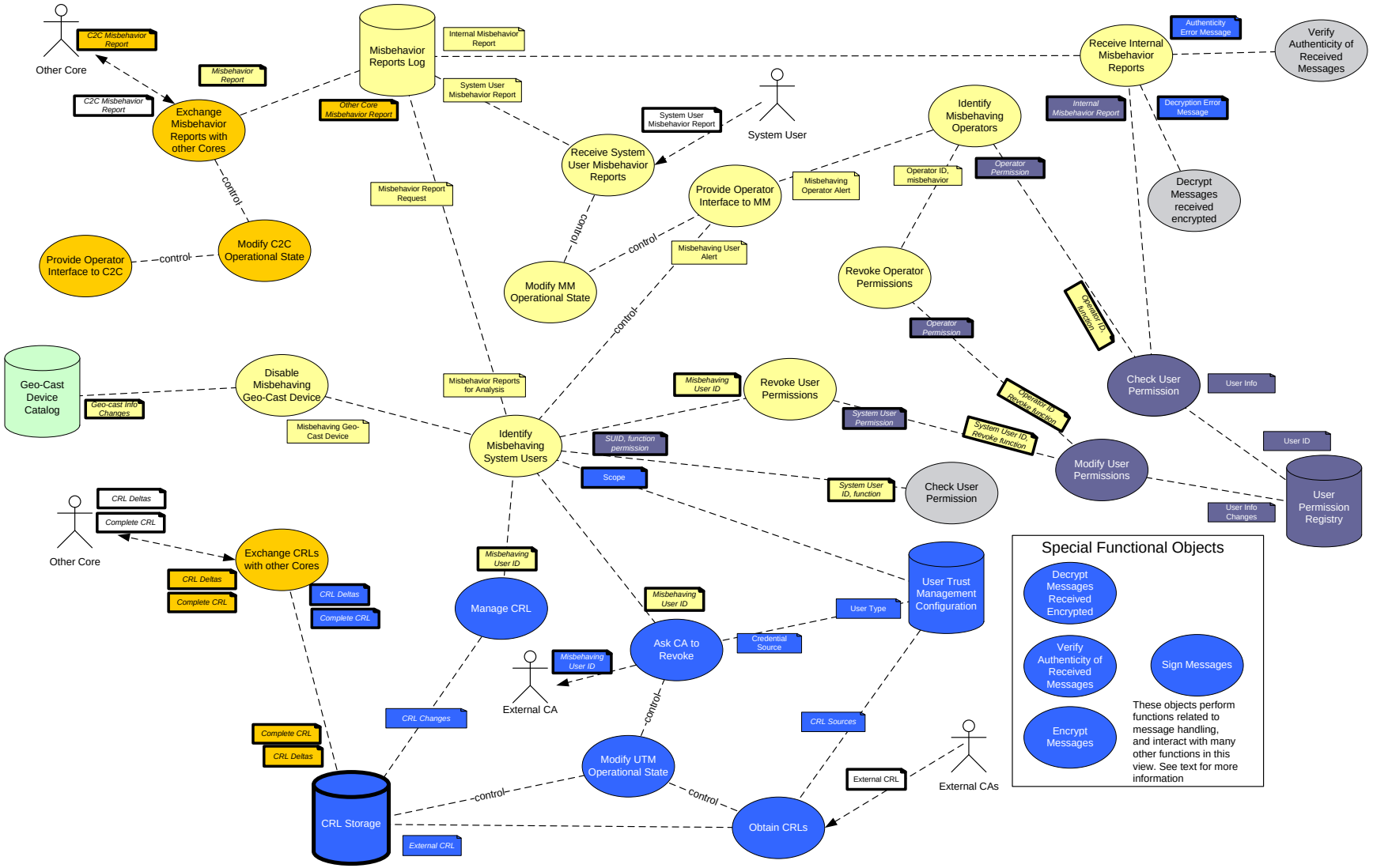


Figure 4-14: Functional View – Misbehavior Management

4.2.7.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Security Credentials Distribution
Enterprise Viewpoint: Enterprise View – Governance
Functional Viewpoint: Functional View – Top Level
Functional Viewpoint: Functional View – User Configuration
Functional Viewpoint: Functional View – Credentials Distribution
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Information Viewpoint: Information View – Top Level External Objects
Information Viewpoint: Information View – Top Level Internal Objects

4.2.8 Functional View – Core Decryption

4.2.8.1 Introduction

Encrypted messages meant for the Core must be decrypted in order for the Core to act on their contents. Decryption using PKI schemes requires the storage of a private key at the receiver. If the Core is to be scalable, it must be able to operate across multiple nodes (see the Connectivity View) which implies the storage of that same private key in multiple places. This makes security more difficult, as there would be multiple points that, if compromised, could surrender the private key. This view documents a mechanism that mitigates this problem.

Message Decryption

This view applies only to encrypted messages that are meant for the Core System. Encrypted messages that come to the Core but are addressed to System Users are **not** decrypted by the Core.

4.2.8.2 Concerns Addressed by this View

Security	How does the Core System secure System Users' personal information?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?

4.2.8.3 Object Definitions and Roles

4.2.8.3.1 Special Functional Objects

These special objects interact with various other functions to perform operations on messages sent or received. They are not shown interacting on the view description diagram for the sake of clarity.

4.2.8.3.1.1 **Sign Messages**

This function attaches the Core System's digital signature to a message by encrypting a hash of the message content with the Core's private key.

4.2.8.3.1.2 **Verify Authenticity of Received Messages**

This function verifies that the message is not a duplicate, has a valid digital signature, that it meets formatting rules and that its contents are within prescribed limits for the message type.

4.2.8.3.2 Service Component

The Service Component object is representative of any other function requiring decryption of a remotely received encrypted message.

Associated Information Objects:

- **Decrypted Message** is the decrypted form of the Remotely Encrypted Message.
- **Remotely Encrypted Message** is an encrypted message intended for the Application Component.

4.2.8.3.3 Decrypt Message Received Encrypted

This object has access to the Core's secret key. It uses that key to decrypt messages and provides that decrypted message to the Encrypt Message Using Core Local Key function.

Associated Information Objects:

- **Decrypted Message** is the decrypted version of the Remotely Encrypted Message.

4.2.8.3.4 Encrypt Message Using Core Local Key

This object encrypts the now-decrypted message with a local encryption mechanism known to all Core subsystems.

Associated Information Objects:

- **Decrypted Message** is the decrypted form of the Remotely Encrypted Message.
- **Locally Encrypted Message** is the locally encrypted form of the decrypted Remotely Encrypted Message.

4.2.8.3.5 Decrypt Locally Encrypted Message

This object decrypts the locally encrypted form of the Remotely Encrypted Message.

Associated Information Objects:

- **Decrypted Message** is the decrypted form of the Remotely Encrypted Message.
- **Locally Encrypted Message** is the locally encrypted form of the decrypted Remotely Encrypted Message.

4.2.8.3.6 Maintain Core Local Key

This object stores the key or keys used to implement the local encryption methods used to securely exchange data between a Service Component and the Core Decryptor.

Associated Information Objects:

- **Local Key** is the key used to implement the local encryption algorithm. This key could be different for different Service Components. It could change periodically or according to Operator action. It is managed by the Configure User Trust Management function (see Functional View – System Configuration).

4.2.8.3.7 Maintain Core Private Key

This object stores the Core's private key.

Associated Information Objects:

- **Private Key** is the Core's private key.

4.2.8.3.8 Notify Misbehavior of Failed Decryption

In the event that a message cannot be decrypted even though it appeared to be intended for the Core, this object will forward this event to Misbehavior Management.

Associated Information Objects:

- **Decryption Error Message** describes the decryption failure and includes a copy of the encrypted message. This message is provided by the Decrypt Message Received Encrypted function and is forwarded to the Receive Internal Misbehavior Reports function.

4.2.8.3.9 Provide Encrypted Message to Decryptor

This object provides the Remotely Encrypted Message to the Decrypt Message Received Encrypted function.

Associated Information Objects:

- **Remotely Encrypted Message** is an encrypted message intended for the Application Component.

4.2.8.3.10 Receive Internal Misbehavior Reports

This object collects misbehavior reports from Core subsystems.

Associated Information Objects:

- **Decryption Error Message** describes the decryption failure and includes a copy of the encrypted message. This message is received from the Notify Misbehavior of Failed Decryption function.

4.2.8.4 View Description

The view below shows the functional decomposition of the functional object Decrypt Message. This decomposition identifies two components dedicated to solving the decryption problem and three functions related to the Core function application that requires the message to be decrypted.

When an Application Component of any service has received a message addressed to the Core System that it needs to have decrypted with the Core's private key, it forwards that message to a message passing component. This message passing component is separated from the Service Component to decouple the service from its local communications processing. The message passing component (Provide Encrypted Message to Decryptor) passes the message to the Decrypt Messages received encrypted function. This function has access to the Core's private key. While not shown on this view, Connectivity View – Core System Functional Allocation makes clear that this function exists on only one node, thus requiring only one location to store this key.

The decrypted message is then passed to the encryption function, which encrypts the message using an encryption algorithm known by the Decrypt Locally Encrypted Message component. For example this could be a symmetric algorithm using pre-shared keys. Once encrypted, the message is passed to a receiver that can decrypt the locally encrypted message and pass the now-decrypted message to the Service Component that needs its content.

This approach was chosen to help ensure the privacy of secure data intended for the Core. By forcing all encrypted data to use the same mechanism for decryption, the Core is better able to secure this data. Data will then only be unencrypted on the Nodes (see Connectivity Views) where it is used, and not throughout the Core's internal network.

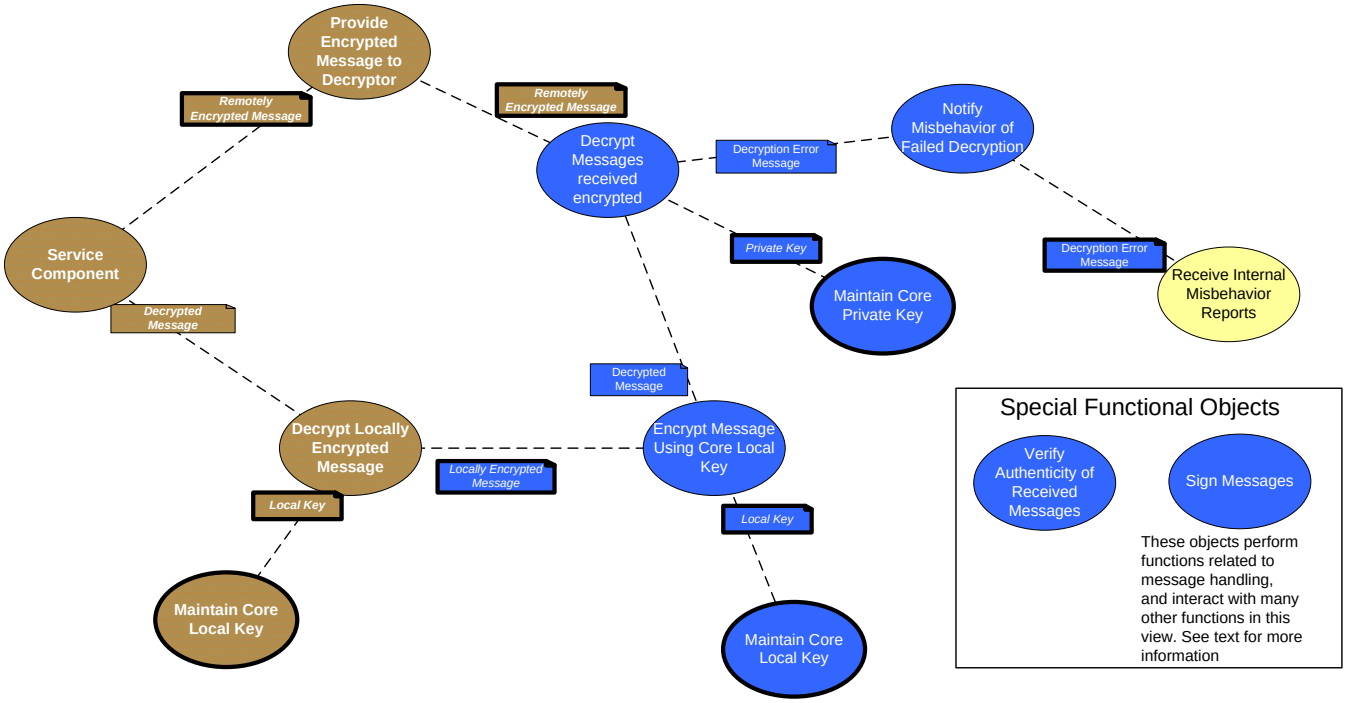


Figure 4-15: Functional View – Core Decryption

4.2.8.5 Configuration Information

The following views must be considered when changing this view:

Functional Viewpoint: Functional View – Top Level
Connectivity Viewpoint: Connectivity View – High Level
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Communications Viewpoint: Communications View – Mobile DSRC Device and Core
Communications Viewpoint: Communications View – Mobile Wide-Area Wireless User and Core
Communications Viewpoint: Communications View – Fixed Point Center/Field User and Core, Core2Core
Information Viewpoint: Information View – Top Level External Objects
Information Viewpoint: Information View – Top Level Internal Objects

4.2.9 Functional View – Networking

4.2.9.1 Introduction

The Core System must be connected to the Internet in order to provide services to System Users. This view explores the functionality required to maintain security and provide communications for the Core.

4.2.9.2 Concerns Addressed by this View

Interfaces	How does the Core System enable control of the services it provides?
Functionality	How does the Core System monitor the services it provides? How does the Core System support the coordination of resources between different Cores? How does the Core System function internally? How do the Core System's components work together? How does the Core System transition between operational modes?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?
Evolvability	How easily can the Core's functionality be expanded to cover new needs if they arise?

4.2.9.3 Object Definitions and Roles

Special Information Objects:

- **All Data In** represents data coming into the Core System from some external source. For the purposes of this view, data is considered "Data In" as long as it originates from an external source. All Data In may be encrypted. All Data In may require acknowledgement.
- **All Data Out** represents data originating from the Core System and destined for some external entity. For the purposes of this view, data is considered "Data Out" as long as it originates from a Core function (e.g. Manage Data Provision Requests). All Data Out may be encrypted. All Data Out may require acknowledgement.

4.2.9.3.1 Actors

4.2.9.3.1.1 **All External**

This actor represents all systems that the Core communicates with, including System Users and External Support Systems.

4.2.9.3.1.2 **Operator**

The Operator is the day-to-day administrator of the Core System. The Operator interacts with the Core through various "Provide Operator Interface to…" Functional Objects.

4.2.9.3.1.3 Private Core, ESS, Field, Center

This represents other Cores, ESS, Field Users and Center Users to which the Core has a private, dedicated network connection.

4.2.9.3.2 Special Functional Objects

These special objects interact with various other functions to perform operations on messages sent or received. They are not shown interacting on the view description diagram for the sake of clarity.

4.2.9.3.2.1 Decrypt Messages Received Encrypted

This function accepts an encrypted message intended for the Core System and decrypts it using the Core's private key.

4.2.9.3.2.2 Encrypt Messages

This function encrypts a message using the public key of the intended recipient.

4.2.9.3.2.3 Sign Messages

This function attaches the Core System's digital signature to a message by encrypting a hash of the message content with the Core's private key.

4.2.9.3.2.4 Verify Authenticity of Received Messages

This function verifies that the message is not a duplicate, has a valid digital signature, that it meets formatting rules and that its contents are within prescribed limits for the message type.

4.2.9.3.3 Generic Service Component

The Generic Service Component object is representative of any other Functional Object.

4.2.9.3.4 Intrusion Detection

This object identifies and reports malicious network and system activity. All data originating from the Core passes through this function. Data Out that is not known to be associated with malicious activity is forwarded to the Route Data Between Networks function. Data that may be associated with malicious activity is not forwarded, but an Intrusion Alert is logged and the Operator is notified.

Associated Information Objects:

- **Intrusion Alert** is sent to the Misbehavior Reports Log and the Provide Operator Interface to NS function. It identifies a malicious action that was detected by Intrusion Detection.

4.2.9.3.5 Intrusion Prevention

This object identifies, reports, and blocks malicious activity. All data intended for a function in the Core System passes through this function. Data In that is not known to be associated with malicious activity is forwarded to the Route Data/Request function. Data that may be associated with malicious activity is not forwarded, but an Intrusion Alert is logged and the Operator is notified.

Associated Information Objects:

- **Intrusion Alert** is sent to the Misbehavior Reports Log and the Provide Operator Interface to NS function. It documents an identified malicious action and/or the status of Intrusion Prevention's response to the action.

4.2.9.3.6 Misbehavior Reports Log

This object maintains the misbehavior reports received by the Core, including those submitted by System Users, and those from other Cores and those generated by internal Core monitoring processes. It accepts queries on those reports and provides the reports matching that query to requesting functions.

Associated Information Objects:

- **Intrusion Alert** is received from the Intrusion Detection and Intrusion Prevention functions; it documents an identified malicious action and/or the status of Intrusion Prevention's response to the action.

4.2.9.3.7 Modify NS Operational State

This object is a controlling function that instructs various functions to change the way they operate; this can include adding or deleting instantiations of an object or by commanding functions to enter into a different state or mode.

4.2.9.3.8 Monitor Service Control Node Performance

This object monitors the performance of application components on Service Control Nodes, and provides summary information to Route Data/Request.

Associated Information Objects:

- **Service Control Node Performance** describes the performance and loading of application components operating on Service Control Nodes.

4.2.9.3.9 Provide Internet Connectivity

This object provides the Core System's connection to the Internet. All data exchanged between the Core and an external entity using the Internet passes through this function.

4.2.9.3.10 Provide Private Network Connectivity

This object provides the Core System's connection to an external entity with which the Core has a private network connection. All data exchanged between the Core and this external entity passes through this function.

4.2.9.3.11 Provide Operator Interface to NS

This object provides the Operator with a window into Network Services operations. It reports performance, status of components, and anomalies, and allows the Operator to control Network Services functions.

Associated Information Objects:

- **Intrusion Alert** is received from Intrusion Prevention or Intrusion Detection functions, it documents an identified malicious action and/or the status of Intrusion Prevention's response to the action.

4.2.9.3.12 Route Data between Networks

This function routes data between private networks the Core System has established with external entities, the Core and the Internet. All combinations of source and destination will be routed:

- Core->Private,
- Private->Core
- Private->Private

- Private->Internet
- Internet->Private
- Core->Internet
- Internet->Core

4.2.9.3.13 Route Data/Request

This function receives data (which may be a request for service, data intended for data distribution, or other data intended for the Core) and routes the data to the appropriate Service Control Node Generic Service Component. It performs this routing based on the destination required for the data and the performance data it receives concerning Service Control Node performance. This function balances Service Control Node performance given available resources.

Associated Information Objects:

- **Service Control Node Performance** describes the performance and loading of application components operating on Service Control Nodes.

4.2.9.4 View Description

The view below shows the basic functionality required of the Core Access Node and Service Router Node (see Connectivity Views for more information about these objects). This addresses the Core's connectivity to private networks and the Internet, and the defense against attack through those networks.

One of the disadvantages of this approach is that all network traffic must be passed through the Intrusion Prevention System (IPS). This creates a potential bottleneck, as the Core's ability to provide services is constrained by the throughput of the IPS. It also introduces additional latency into the data stream. While no Core functions have such low latency requirements that this will have a significant effect, external applications that rely on data provided by the Core may have to account for this.

On the other hand, by maintaining a single point of control over inputs and outputs, adding new services to the Core is simplified: add the new service, reprogram the Route Data/Request function and publish the new interface to relevant developers.

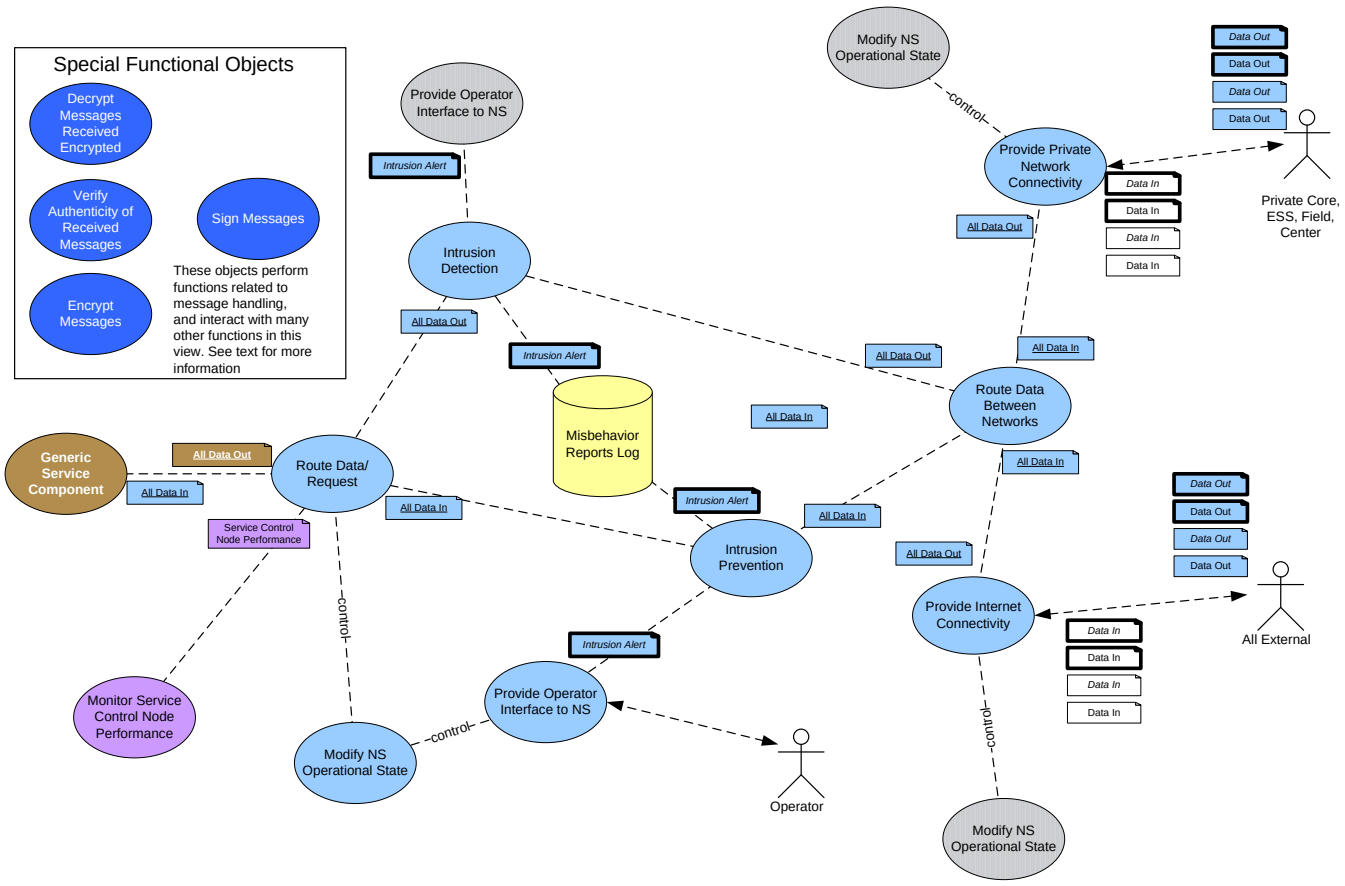


Figure 4-16: Functional View – Network Connectivity

4.2.9.5 Configuration Information

The following views must be considered when changing this view:

Connectivity Viewpoint: Connectivity View – High Level
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Communications Viewpoint: Communications View – Core Routing
Information Viewpoint: Information View – Top Level External Objects
Information Viewpoint: Information View – Top Level Internal Objects

4.2.10 Functional View – Core Backup

4.2.10.1 Introduction

Core Systems may provide backup functionality to one another. This includes backup of services, where one Core may provide services in behalf of another Core, and also backup of data, since data backup is required to implement service backup. This view addresses the functionality required to implement service and data backup.

4.2.10.2 Concerns Addressed by this View

Interfaces	How does the Core System enable control of the services it provides?
Functionality	How does the Core System monitor the services it provides? How does the Core System support the coordination of resources between different Cores? How does the Core System function internally? How do the Core System's components work together? How does the Core System transition between operational modes?
Security	How does the Core System secure System Users' personal information?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?

4.2.10.3 Object Definitions and Roles

4.2.10.3.1 Actors

4.2.10.3.1.1 **Other Core**

This actor represents another Core System.

4.2.10.3.1.2 **Operator**

The Operator is the day-to-day administrator of the Core System. The Operator interacts with the Core through various "Provide Operator Interface to…" Functional Objects.

4.2.10.3.2 Special Functional Objects

These special objects interact with various other functions to perform operations on messages sent or received. They are not shown interacting on the view description diagram for the sake of clarity.

4.2.10.3.2.1 **Decrypt Messages Received Encrypted**

This function accepts an encrypted message intended for the Core System and decrypts it using the Core's private key.

4.2.10.3.2.2 **Encrypt Messages**

This function encrypts a message using the public key of the intended recipient.

4.2.10.3.2.3 **Sign Messages**

This function attaches the Core System's digital signature to a message by encrypting a hash of the message content with the Core's private key.

4.2.10.3.2.4 **Verify Authenticity of Received Messages**

This function verifies that the message is not a duplicate, has a valid digital signature, that it meets formatting rules and that its contents are within prescribed limits for the message type.

4.2.10.3.3 Backup Other Core Data

This object receives requests for data backup from other Cores, provides accepted formats, and accepts data for backup. It stores data in Other Core Data Backups stores.

Associated Information Objects:

- **Backup Data** is data that has been extracted from the other Core's data store and sent to this Core in a format that this Core will accept.
- **Data Backup Request** is the request received from another Core to provide data backup.
- **Other Core Backup Data** is the Backup Data formatted for storage in the Other Core Data Backup.

4.2.10.3.4 Check User Permission

This object accepts a System User ID or Operator ID, along with a type of operation that the user is attempting to access, and responds with whether or not the user is permitted that action.

Associated Information Objects:

- **Operator ID, function** is the identity of the operator which is received from the relevant subsystem Operator Interface function to verify whether the operator is permitted a given **function**.
- **Permission** is the response from the Check User Permission function describing whether or not the **User** (System User, Core or Operator, depending on the original request) is permitted this action.

4.2.10.3.5 Configure Geo-cast Device Information

This object allows access to the Geo-Cast Device catalog.

Associated Information Objects:

- **Geo-cast Info Changes** describes changes to the geo-cast info of geo-cast supporting devices; this includes performance and operating characteristics, and the permissions System Users must have to access these devices.

4.2.10.3.6 Core Takeover

This object receives requests to change service scope from other Cores, reconfigures the Core and notifies Operator of changes. This may include automated changes to any subsystem's configuration, and also changes to user data subscription, data acceptance and geo-cast device stores of the Data Distribution subsystem. Configuration changes and changes to data distribution stores will take place according to pre-assigned settings maintained by each subsystem.

Associated Information Objects:

- **Other Core Takeover Request** is the request received from another Core for this Core to change its service scope and take over some services from the other Core.

4.2.10.3.7 Data Acceptance Catalog

This data store maintains a catalog of data types and sources that are used by the Core's data distribution function. It also maintains a list of the data types it does not accept but some other facility does. It allows changes to the catalog to update existing and add new data types, sources and combinations of types and sources. This function responds to queries about data types telling whether or not that type/source combination is accepted for data distribution. It also analyzes its data distribution coverage area, time, and source/type combinations against those reported by other Cores to determine if there are any overlaps (where multiple Cores provide the same services in the same area) or conflicts (where Cores provide different and conflicting information for the same area, for instance if one Core says that a given data type must go to external sink A, but another Core says that it accepts that data type).

Associated Information Objects:

- **Data Acceptance Changes** are received from the Modify Data Acceptance Catalogs function. This includes the types of data and their source or source type, and whether the Core should accept or not accept that data for distribution.

4.2.10.3.8 Data Subscription Catalog

This data store maintains a catalog of data types and sources and the System Users that want to receive that data. It may allow specification of aggregation period and sampling rate for each System User. Allowable sampling rates and aggregation specifications may be restricted based on configuration and performance of this subsystem. Allows changes to the catalog to update existing and add new subscribers and responds to queries about data subscribers and their subscriptions.

Associated Information Objects:

- **Data Subscription Changes** is received from Modify System User Data Subscriptions. It specifies a System User and changes (additions if new) to the user's subscription.

4.2.10.3.9 Generic Configure Subsystem

This object represents all the Configure [subsystem] objects. It allows the operator to configure all subsystem functions. Configuration information is provided to the relevant subsystem Configuration store. Includes:

- Configure Core2Core
- Configure Data Distribution
- Configure Misbehavior Management
- Configure Network Services
- Configure Service Monitor
- Configure Time Synchronization
- Configure User Permissions
- Configure User Trust Management

4.2.10.3.10 Generic Core Data Store

This data store is representative of all Core data stores, including configuration data stores but also other operational stores, such as Data Acceptance Catalog or User Permissions Catalog.

Associated Information Objects:

- **Data to be Backed Up** is data from the Core data stores that will be backed up, sent to the Provide Data to be Backed Up function.
- **Restore Data** is data from another Core to be used to restore a Core data store.

4.2.10.3.11 Generic Provide Operator Interface

This object represents all Provide Operator Interface to [subsystem] functions. These functions provide a user interface for all subsystems to the Operator, allowing him to interact with them. This function interacts with many other monitoring, configuration and settings functions to enable the Operator to manage the Core. It provides a display of current operational status, access to historical performance and access to functions used to configure Core services.

Associated Information Objects:

- **Operator ID, function** is the identity of the operator which is sent to the Check User Permission function to verify whether the operator is permitted a given **function**.
- **Other Core Service Configuration Data** is configuration data extracted from the relevant Other Core Data Backup.
- **Permission** is the response from the Check User Permission function describing whether or not the **User** (System User, Core or Operator, depending on the original request) is permitted this action.

Includes:

- Provide Operator Interface to C2C
- Provide Operator Interface to DD
- Provide Operator Interface to MM
- Provide Operator Interface to NS
- Provide Operator Interface to SM
- Provide Operator Interface to TS
- Provide Operator Interface to UP
- Provide Operator Interface to UTM

4.2.10.3.12 Generic Subsystem Configuration

These stores maintain configuration data for subsystem functions, including startup and operating parameters. Configuration data is made available to all subsystem functions.

Includes:

- Core2Core Configuration
- Data Distribution Configuration
- Misbehavior Management Configuration
- Network Services Configuration
- Service Monitor Configuration
- Time Synchronization Configuration
- User Permissions Configuration
- User Trust Management Configuration

Generic Subsystem Configuration refers to all of the Configuration stores. It is drawn separately to simplify the diagram.

4.2.10.3.13 Geo-cast Device Catalog

This data store keeps track of the information necessary for System Users to perform geo-casting. This includes locations, addresses, and ranges of devices that accept geo-cast messages.

Associated Information Objects:

- **Geo-cast Info Changes** describes changes to the geo-cast info of geo-cast supporting devices; this includes performance and operating characteristics, and the permissions System Users must have to access these devices. This message is received from the Configure Geo-Cast Device Information function.

4.2.10.3.14 Get Backed Up Data

This object retrieves data backed up by another Core and provides it to the appropriate data Store.

Associated Information Objects:

- **Restore Data** is data from another Core to be used to restore a Core data store.
- **Data Request** is a request sent to another Core to retrieve data previously backed up. Includes specification of the data set that was backed up.

4.2.10.3.15 Modify C2C Operational State

This object is a controlling function that instructs various Core2Core functions to change the way they operate; this can include adding or deleting instantiations of an object or by commanding functions to enter into a different state or mode.

4.2.10.3.16 Modify Data Acceptance Catalog

This object modifies the Data Acceptance Catalog indicating what types/sources of data the Core accepts and what types are handled directly by external data sinks. In the context of Core Backup, this may include a change in the scope of data acceptance: geographic area and/or source types.

Associated Information Objects:

- **Data Acceptance Changes** are sent to the Data Acceptance Catalog. This includes the types of data and their source or source type, and whether the Core should accept or not accept that data for distribution.

4.2.10.3.17 Modify System User Data Subscriptions

This object modifies existing or creates new System User data subscriptions, according to input from Other Core Data Backups.

Associated Information Objects:

- **Other Core Subscriptions** is received from Other Core Data Backups, and includes subscription information for System Users from the Core for which this Core is providing backup services.
- **Data Subscription Changes** is a request sent to the Data Subscriptions Catalog. It specifies a System User and changes (additions if new) to the user's subscription.

4.2.10.3.18 Monitor Core Services Performance

This object monitors the performance of all Core System functions and interfaces. This includes measures of throughput, buffer levels, and resource usage. This information is provided to the Request Core Takeover function.

Associated Information Objects:

- **Detailed Service Status** is detailed information describing the performance of all Core functions and interfaces provided to the Log System State, Provide Service Status to Other Cores and Performance and Provide Operator Interface functions.

4.2.10.3.19 Other Core Data Backups

This data store holds backup data provided by other Cores. It responds to backup and restore requests.

Associated Information Objects:

- **Other Core Backup Data** is accepted, stored and distributed upon request.
- **Other Core Service Configuration Data** is provided to Generic Configure Subsystem on request. It includes configuration data relevant to the subsystem in question.

4.2.10.3.20 Provide Other Core Data

This function provides backup data to a Core that backed up the data to this Core.

Associated Information Objects:

- **Other Core Backup Data** is data originally provided by another Core, stored in the Other Core Data Backups, sent back to the original Core
- **Restore Data** is data from another Core to be used to restore a Core data store.
- **Data Request** is a request received from another Core to retrieve data previously backed up. Includes specification of the data set that was backed up.

4.2.10.3.21 Provide Data to be Backed Up

This object requests data backup services from another Core. It extracts data from Core data stores and provides that data to the other Core.

Associated Information Objects:

- **Backup Data** is data that has been extracted from a Core data store and put into a format that the other Core will accept, and then sent to that Core.
- **Data Backup Request** is the request made of another Core to provide data backup.
- **Data to be Backed Up** is data from Core data stores that will be backed up.

4.2.10.3.22 Request Core Takeover

This object sends a request to another Core for that Core to change service scope. This may be a request to increase or decrease service area. An increase will be requested in response to this Core's need to decrease scope as a result of a performance issue, in which case the request is prompted by the performance data this function receives from Service Monitor. A decrease will be requested when this Core is prepared to resume its nominal services. Either request may also be Operator controlled. When a Core Takeover is acknowledged by another Core, this Core would alter the performance of the services it requested by taken over.

Associated Information Objects:

- **Detailed Service Status** is detailed information describing the performance of all Core functions and interfaces received from the Monitor Core Services Performance function.
- **Other Core Takeover Request** is the request sent to another Core for that Core to change its service scope.

4.2.10.4 View Description

Core Backup addresses two functions:

- Expansion of Core System services in response to another Core's request and subsequent reduction when that other Core is ready to resume services.
- Backup of data used to maintain operations where the data backup facility is provided by another Core System.

Data backup applies to any Core System data store. In order to perform backup, the Core asking for backup service (Source Core) must negotiate with the Core providing the backup service (Backup Core). The Backup Core will indicate whether it can provide backup, in what format it requires data. The Source Core can then transmit data to be backed up to the Backup Core. Any data that is stored in encrypted form on one Core will be stored in that same encrypted form on the Backup Core. Enterprise relationships necessary to implement backup are discussed in Enterprise View – Operations.

Similar procedures are used for restoration of Source Core data: the Source Core queries for the backup data, the Backup Core responds with what data is available, the Source Core selects the backup data it wishes and the Backup Core provides it.

Data Backup also allows distribution of Core operational data to other Cores that may use that data to provide services when the Source Core requires it.

When a Core System needs to enter a maintenance state (to repair hardware, patch software, etc.) it may need to take a service offline. It can ask for another Core to provide that service when it is in a maintenance state. This Core Takeover function may require changes to the configuration of any Core subsystem, but in particular Data Distribution.

In order for Core Takeover to work, the Core System requesting service expansion (Maintenance Core) must provide the Core that will provide expanded services (Takeover Core) with relevant configuration information. This could happen using Core Backup, or manually through the Operator. In either case, the relevant configuration information must be present in the Takeover Core for Takeover to work.

Once the Maintenance Core can return to normal operations, it asks the Takeover Core to change its service scope back to what it was prior to Takeover, and the Maintenance Core can then resume services.

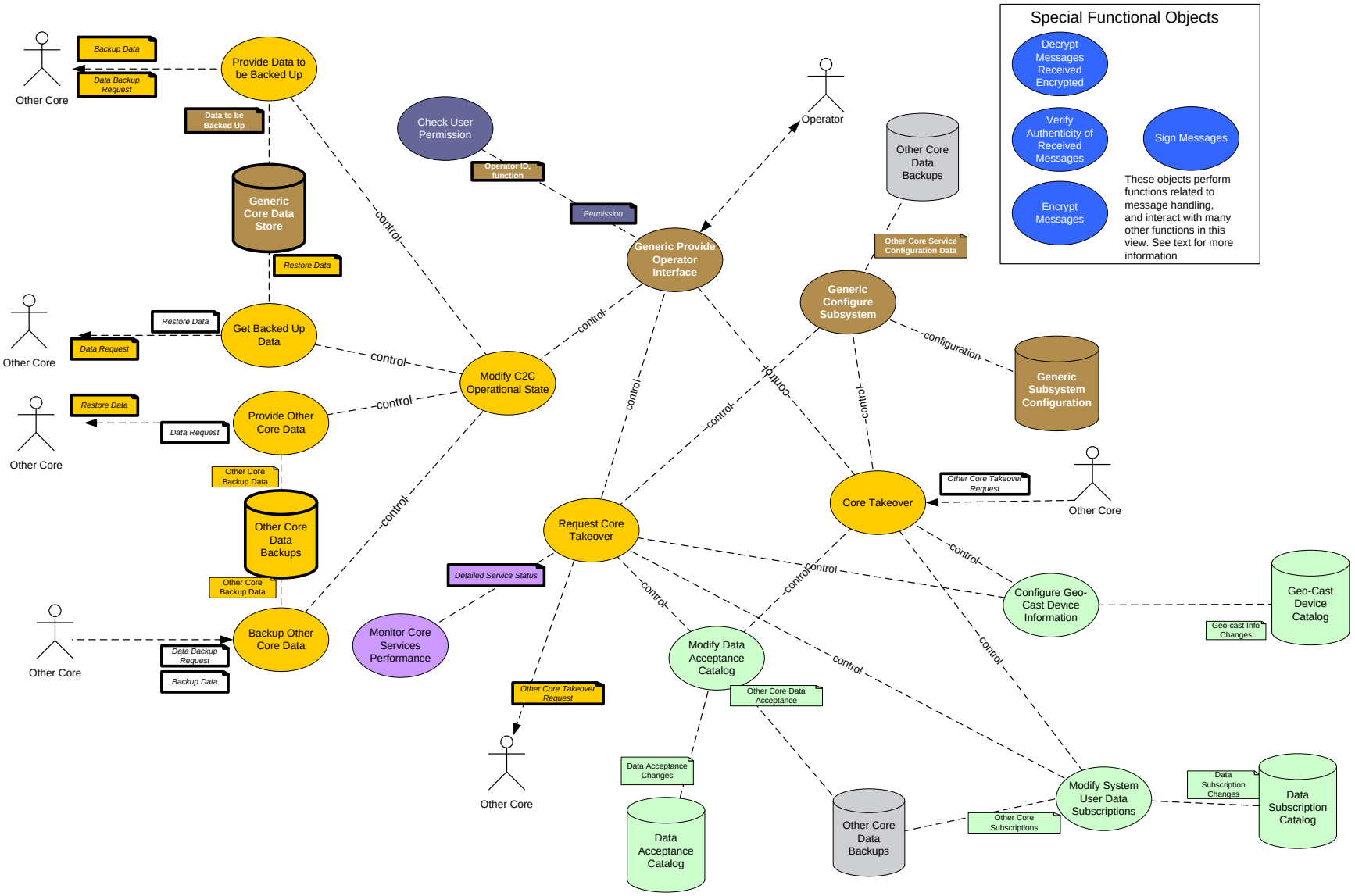


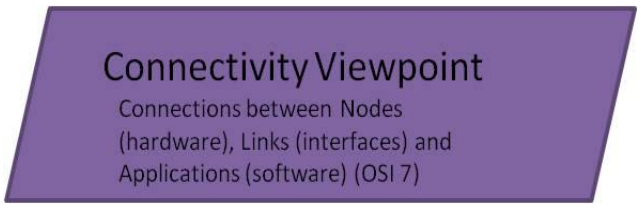
Figure 4-17: Functional View – Core Backup

4.2.10.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Security Credentials Distribution
Enterprise Viewpoint: Enterprise View – Configuration and Maintenance
Enterprise Viewpoint: Enterprise View – Operations
Enterprise Viewpoint: Enterprise View – Governance
Enterprise Viewpoint: Enterprise View – Business Model Facilitation
Functional Viewpoint: Functional View – Top Level
Functional Viewpoint: Functional View – System Configuration
Functional Viewpoint: Functional View – Credentials Distribution
Information Viewpoint: Information View – Top Level External Objects
Information Viewpoint: Information View – Top Level Internal Objects

4.3 Connectivity Viewpoint



The first Connectivity View illustrates the relationships between the Nodes that implement the Core System. Following that is a view that takes the Core System Nodes and allocates functions to those Nodes. Functions are implemented by Engineering Objects (mostly software). Several different representative allocation implementations are shown. Lastly, a view is presented that discusses mode transitions.

The three Connectivity Views presented are:

- High Level connectivity
- Allocation of functional to software and hardware objects
- State and mode transitions

Table 4-3 shows which Connectivity Views are relevant to each stakeholder.

Table 4-3: Connectivity View Stakeholder Matrix

Stakeholder / Connectivity View	Mobile User	Field User	Center User	Operator	Acquirer	Maintainer	Developer	Manager	Tester	Policy Setter	Application Developer	Device Developer	Service Provider
High Level	■	■	■		■	■	■		■		■	■	■
Core System Function Allocation						■	■		■				
State and Mode Transition				■	■	■	■	■	■	■			

The following Connectivity Views are relevant to each Stakeholder as shown in the Table above:

- Mobile Users: “High Level”
- Field Users: “High Level”
- Center Users: “High Level”
- Operators: “State and Mode Transition”
- Acquirers: “High Level” and “State and Mode Transition”
- Maintainers: “High Level,” “Core System Functional Allocation,” and “State and Mode Transition”
- Developers: “High Level,” “Core System Functional Allocation,” and “State and Mode Transition”
- Managers: “State and Mode Transition”
- Testers: “High Level,” “Core System Functional Allocation,” and “State and Mode Transition”
- Policy Setters: “State and Mode Transition”
- Application Developers: “High Level”
- Device Developers: “High Level”
- Service Providers: “High Level”

4.3.1 Connectivity View – High Level

4.3.1.1 Introduction

In order to promote scalability, it is necessary to allow deployment of the Core functionality across multiple nodes. This view shows the highest level view of the distributed Core and how it is connected to System Users.

4.3.1.2 Concerns Addressed by this View

Performance	Can the Core meet all of the performance requirements defined in the SyRS (e.g., availability, reliability, capacity, and other quantitative measures)?
Security	How are the Core System components secured from network attack? How are the Core System components physically secured?
Feasibility	Are Core System services feasible to develop given current technology and resources?
Risks	Is the Core System's hardware and software architecture susceptible to failure, and if so under what circumstances? What are the characteristics of this failure?
Evolvability	Is the structure of the Core System sufficiently flexible and scalable to deploy and to enable changes to cover new needs if they arise?
Maintainability	Can the Core's functionality be sustained with acceptable levels of downtime as per the SyRS?

4.3.1.3 Object Definitions and Roles

Core Nodes: All Core Nodes are located in a fixed, environmentally controlled, and secured facility meeting facility requirements for a Tier 2 data center as specified in Appendix G of TIA-942. They are connected to one another by wired communications.

4.3.1.3.1 Core Access Node

This is the Core System's gateway to the Internet and private networks. All communications traffic between the Core and any external entity passes through the Core Access Node. Incoming messages that pass security checks are passed to the Service Router. Outgoing messages are received directly from Service Component Nodes, and if they pass security checks are forwarded to the appropriate network (Internet or private).

4.3.1.3.2 Core Decryptor

This node is responsible for providing decryption of encrypted messages intended for the Core, and providing those messages back to the Service Component Node that requires them, in a locally encrypted form that the receiving SEO on the Service Component Node can interpret.

4.3.1.3.3 Core Switch

This node is responsible for switching Core System internal network traffic between Core Nodes. There are two Core Switches; each is a network switching device providing connectivity between Core Nodes. One provides connectivity between Core Decryptor, Core Access Node, Service Router and Service Component Nodes. The second provides connectivity between the Core Decryptor and Service Component Nodes.

4.3.1.3.4 Service Component Node

The Service Component Node (SCN) is the service provider for the Core System. It routes all communications traffic to System Users through the Service Router and receives communications from System Users through the Service Router. There may be multiple SCNs in the Core.

Each SCN may operate SEOs that implement any combination of Functional Objects.

4.3.1.3.5 Service Router

The Service Router maintains an understanding of the loading of all Service Component Nodes, including what processes are running on each node and how much capacity remains for each process. It receives incoming data from the Core Access Node and forwards it to the appropriate Service Component Node.

User Nodes: These nodes represent the external or System Users of Core services.

4.3.1.3.6 Center User Node

This is the Center User. This user operates an application at fixed location and connects to the Core through the Internet or through a private connection to the Service Router.

4.3.1.3.7 Field User Node

This is the Field User. This user operates an application at a fixed location and connects to the Core through the Internet or through a private connection to the Service Router.

4.3.1.3.8 Mobile User Node

This is the Mobile User. This user operates an application on a mobile device. It connects wirelessly by 5.9 GHz DSRC to the DSRC Field Node, through cellular communications to the Cellular Node, or through other wireless technology to the Other Wireless Node.

Communications Nodes: These nodes provide communications between User Nodes and Core Nodes.

4.3.1.3.9 Cellular Node

This represents the cellular infrastructure, including 3G and 4G infrastructure, which provides a gateway to Core services through the Internet.

4.3.1.3.10 DSRC Field Node

This represents 5.9 GHz DSRC-based RSE that provide a gateway to Core services through the Internet or through a private connection to the Service Router.

4.3.1.3.11 Internet

This is the publically accessible Internet, through which users may communicate to gain access to the Core.

4.3.1.3.12 Other Wireless Access Node

This represents other forms of wireless technology that the Mobile User Node may have access to, including municipal Wi-Fi or home wireless networks.

4.3.1.3.13 Private Network Access Node

This is a device that provides access to a privately-controlled network, through which the Center or Field User can reach the Core System. Traffic between the Center or Field and the Core traverses this private network. For Field and Center Nodes, maintenance and management of this node is the responsibility of the Field or Center Node that uses this Node to connect to the Core (see Enterprise View – Configuration and Maintenance). For other Cores, the maintenance and management of this Node may be shared between Cores that use the Node.

4.3.1.3.14 Fixed Point Access Node

This represents a fixed access point with an Internet backhaul connection that the Field and Center User have access to. This includes T1, T3, fiber-optic and similar fixed point cabled technologies as well as municipal Wi-Fi or commercial or home wireless networks.

Links: Core Nodes are connected by physical, wired links. The Core Access Node is connected to the Internet by a physical, wired link. Communications Nodes are connected to one another by physical links that are most likely wired connections. Field, Center and Mobile User are connected to Communications Nodes by whatever means is at their disposal. For Center and Field Nodes, this is most likely a wired connection. Mobile Nodes connect through a wireless connection.

4.3.1.4 View Description

The Core Access Node provides the interface between the Internet and the Core System's Service Component Nodes. All communications traffic originating outside the Core also passes through the Service Router. The Service Router selects a Service Component Node to receive communications traffic depending on load.

A Core has a minimum of one Service Component Node but no maximum. Service Component Nodes run SEOs that provide Core functionality. SCNs may be organized by subsystem (e.g. User Permissions node) or function (e.g. Maintain DSRC Anonymous Certificates). This division into smaller systems or functions improves Core computational scalability to the extent that Core processing capacity is essentially unconstrained by the architecture. This also allows the core to evolve and support new capabilities without requiring substantial re-architecting.

A Core has a single Core Decryptor node. This node includes the functionality for decrypting messages intended for the Core, re-encrypting using a Core-specific encryption method and sending them to the SEO they were intended for.

Mobile User Nodes are connected wirelessly. All other nodes can be connected wirelessly or by wire.

All Nodes may access the Core System through the Internet. System Users may arrange for a private connection to the Core System, if the Core System Manager (see various Enterprise Views) is willing to allow a private connection from that user. Similarly, Cores may communicate using private connections if the Core System Managers agree to do so. For example, Cores that exchange large amounts of data, in particular Cores that provide backup relationships to one another, may consider a private connection more appropriate.

Reliance on one Node for decryption is a failure risk, but necessary to ensure the protection of private data. Reliability of this node may be enhanced by operating a redundant node that provides fault tolerance. The degree to which the Core Decryptor should be made fault tolerant will depend on the Core's deployment.

Reliance on a single Core Access Node for network access is another failure risk, but one that cannot be avoided. Each Core must defend itself against cyber-attack, and thus all incoming communications traffic must be analyzed at the point of entry to the system. This node can be made fault tolerant, but the configuration of redundant sub-nodes within the Core Access Node could be complex. Like the Core Decryptor, determination of this redundant arrangement will depend on the deployment; those Cores with no tolerance for failure may need to invest in a more complex, available Core Access Node.

Overall availability is specified in the SyRS as a system performance requirement. TIA-942 tier 2 data center guidance yields approximately 99.7% availability, which meets the SyRS availability requirement.

SCNs, Core Decryptor and Service Router could be implemented on Nodes that are not connected to the same Core Switch. Some or all links could be implemented using Virtual Private Networks (VPNs). This has the advantage of allowing more flexibility in implementation, particularly if the Core is hosted by a remote hosting operation. It has the disadvantage of removing the physical security associated with a dedicated network and relying on the security of the chosen VPN solution(s). This approach is viewed as an alternative, but not a replacement to the primary approach illustrated here.

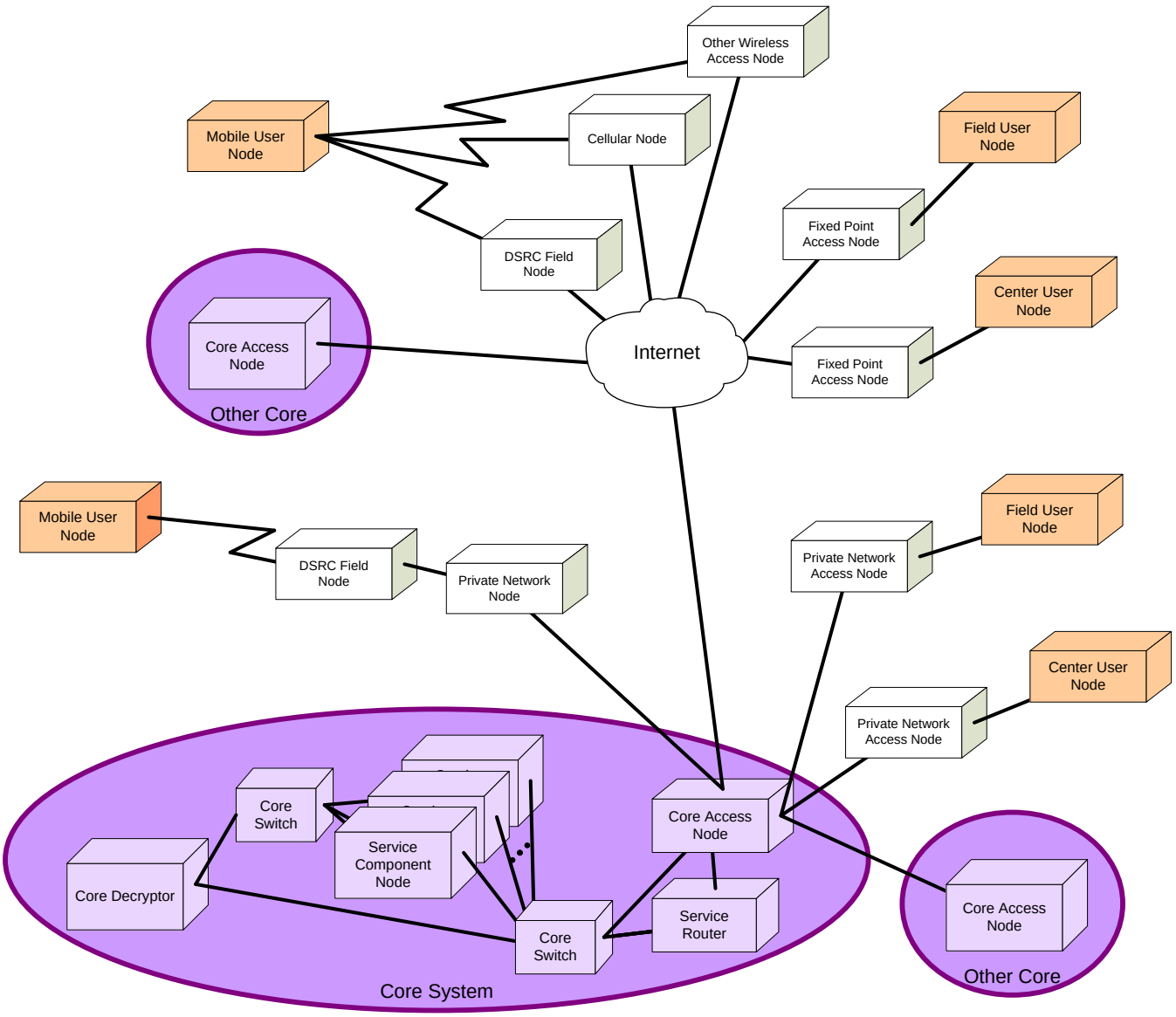


Figure 4-18: Connectivity View High Level

4.3.1.5 Configuration Information

The following views must be considered when changing this view:

Enterprise Viewpoint: Enterprise View – Configuration and Maintenance
Enterprise Viewpoint: Enterprise View – Governance
Enterprise Viewpoint: Enterprise View – Business Model Facilitation
Functional Viewpoint: Functional View – Core Decryption
Functional Viewpoint: Functional View – Networking
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Communications Viewpoint: Communications View – Mobile DSRC Device and Core
Communications Viewpoint: Communications View – Mobile Wide-Area Wireless User and Core

4.3.2 Connectivity View – Core System Functional Allocation

4.3.2.1 Introduction

This view illustrates the allocation of Core System Functional Objects to Engineering Objects and the assignment of Engineering Objects to component nodes. This includes identification of devices (Hardware Engineering Objects, aka Nodes) and SEOs. It includes an allocation of SEOs to Nodes, and a discussion of scalability.

4.3.2.2 Concerns Addressed by this View

Performance	Can the Core System provide services with sufficient responsiveness to enable System User applications? Can the Core meet all of the performance requirements defined in the SyRS (e.g., availability, reliability, capacity and other quantitative measures)?
Interfaces	Can the Core System meet the interface requirements defined in the SyRS?
Security	What physical elements are involved in the distribution and revocation of digital certificates, and what are their roles?
Feasibility	Are Core System services feasible to develop given current technology and resources?
Risks	Is the Core System's hardware and software architecture susceptible to failure, and if so under what circumstances? What are the characteristics of this failure?
Evolvability	Is the structure of the Core System sufficiently flexible and scalable to deploy and to enable changes to cover new needs if they arise?
Deployability	Are the Engineering Objects that make up the Core System practical to deploy in the projected deployment environment and in a timely manner?
Maintainability	Is the structure of the Core System maintainable with a reasonable allocation of resources for the entities that are likely to consider deployment?

4.3.2.3 Object Definitions and Roles

Engineering Objects defined in this section implement Functional Objects defined in the Functional Views. References are made to those Functional Objects to aid the reader in determining traceability between the Functional and Connectivity Views. Many Functional Objects appear in more than one View. To aid the developer, all references are included here. Developers should reference the Functional Object descriptions for further detail about the activities that a given SEO is responsible for.

Nodes:

4.3.2.3.1 Core Decryptor

This Node is responsible for providing decryption of encrypted messages intended for the Core, and providing those messages back to the Service Component Node that requires them in a locally encrypted form that the receiving SEO on the Service Component Node can interpret.

The Core Decryptor is physically connected to all Service Component Nodes. Any message requiring decryption is passed from a Service Component Node to the Core Decryptor which performs the decryption and returns the message in locally encrypted form.

Table 4-4: Core Decryptor Engineering Objects

Name	Description	Functional Objects Implemented	Interfaces
SEO-Authenticity-Verifier	This SEO verifies the authenticity of messages. While nominally a User Trust Management function, this is called out separately because it must reside on every SCN.	4.2.2.3.2.4 Verify Authenticity of Received Messages 4.2.3.3.2.4 Verify Authenticity of Received Messages 4.2.4.3.2.4 Verify Authenticity of Received Messages 4.2.5.3.2.4 Verify Authenticity of Received Messages 4.2.6.3.2.4 Verify Authenticity of Received Messages 4.2.7.3.2.4 Verify Authenticity of Received Messages 4.2.8.3.1.2 Verify Authenticity of Received Messages 4.2.9.3.2.4 Verify Authenticity of Received Messages 4.2.10.3.2.4 Verify Authenticity of Received Messages	SEO-Decryptor SEO-Core-Config SEO-System-Log
SEO-Decryptor	This SEO receives all messages from System Users that are encrypted and addressed to the Core. The Decryptor maintains the Core's private encryption keys. It uses those keys to decrypt the message, and passes it to a local encryptor, choosing the encryptor appropriate for the SEO that is to	4.2.2.3.2.1 Decrypt Messages Received Encrypted 4.2.3.3.2.1 Decrypt Messages Received Encrypted 4.2.4.3.2.1 Decrypt Messages Received Encrypted 4.2.5.3.2.1 Decrypt Messages Received Encrypted 4.2.6.3.2.1 Decrypt Messages Received Encrypted 4.2.7.3.2.1 Decrypt Messages Re-	SEO-Local-Encryptor SEO-Core-Config SEO-System-Log

Name	Description	Functional Objects Implemented	Interfaces
	eventually receive the message.	ceived Encrypted 4.2.9.3.2.1 Decrypt Messages Received Encrypted 4.2.10.3.2.1 Decrypt Messages Received Encrypted 4.2.8.3.3 Decrypt Message Received Encrypted 4.2.8.3.7 Maintain Core Private Key 4.2.8.3.8 Notify Misbehavior of Failed Decryption 4.2.8.3.9 Provide Encrypted Message to Decryptor	
SEO-Local-Encryptor	This SEO implements an encryption algorithm known to at least one other Core SEO. It receives messages from an SEO Operating on the same Node (in this case the Decryptor), encrypts those messages and then passes them to the SEO that the message is intended for. There are several Local Encryptors, one for each local encryption algorithm.	4.2.8.3.4 Encrypt Message Using Core Local Key 4.2.8.3.6 Maintain Core Local Key	SEO-Core-Cert SEO-Core-Config SEO-System-Log
SEO-Local-Signer	This SEO maintains a key and uses it to sign messages output from the SCN that it operates on. It obtains the key from SEO-Core-Cert.	4.2.8.3.1.1 Sign Messages	SEO-Core-Cert SEO-Core-Config SEO-System-Log
SEO-Core-Config	This SEO maintains the configuration of the Core, including startup and operating parameters, allocation of SEO's to HEO's and configuration of devices used by the Core, including Geo-	4.2.2.3.5 Data Acceptance Catalog 4.2.2.3.7 Geo-cast Device Catalog 4.2.3.3.5 [Subsystem] Configuration / Generic Subsystem Configuration 4.2.4.3.8 Data Acceptance Catalog 4.2.4.3.11 Geo-cast Device Catalog 4.2.6.3.16 User Trust Management	SEO-Core-Config-Mod SEO-Other-Core-Config SEO-User-Registry-Mod SEO-MR-Log

Name	Description	Functional Objects Implemented	Interfaces
	cast devices.	Configuration 4.2.7.3.26 User Trust Management Configuration 4.2.7.3.9 Geo-cast Device Catalog 4.2.10.3.7 Data Acceptance Catalog	SEO-System-Log
SEO-System-Log	This SEO maintains a record of Core performance, state changes and events, and the state of other Cores that it monitors. It provides a mechanism for extracting performance information for the Core Certification Authority.	4.2.4.3.7 Core Register to Receive Status from Other Core 4.2.5.3.4 Event Log 4.2.5.3.19 System State and Performance Log 4.2.5.3.13 Monitor Core Services Performance 4.2.5.3.14 Monitor Status of other Cores 4.2.5.3.15 Provide Record of System Performance	All other SEOs Core Certification Authority SEO-Core-Config SEO-System-Log
SEO-Health-Monitor	This SEO monitors the health and integrity of the Core and its operating environment, and takes action if the Core's performance is in jeopardy.	4.2.5.3.12 Monitor Core Health and Safety 4.2.5.3.20 Take Action in Response to Environment Issue 4.2.9.3.8 Monitor Service Control Node Performance	SEO-Core-Controller SEO-Core-Config SEO-System-Log

4.3.2.3.2 Service Component Node

This is the service provider for the Core System. It routes all communications traffic to System Users through the Service Router and receives communications from System Users through the Service Router. There may be multiple SCNs in the Core.

Each SCN may operate Software Engineering Objects that implement any combination of Functional Objects.

Service Component Nodes receive messages from System Users and other external entities (e.g., Cores, ESS) through the Service Router. All messages (whether they are signed and/or encrypted, no matter its purpose) reach Service Component Nodes only after passing through the Service Router. Messages sent from an SEO to a System User or other Core go directly to the Core Access Node.

Table 4-5: Service Component Node Engineering Objects

Name	Description	Functional Objects Implemented	Interfaces
SEO-Operator-Interface	This SEO provides the Operator with an interface to every subsystem, the ability to configure that subsystem through access to the SEO-Core-Config-Mod SEO, and to control the state and mode of objects within that subsystem.	4.2.2.3.15 Provide Operator Interface to DD 4.2.3.3.12 Provide Operator Interface to [Subsystem] / Generic Subsystem Operator Interface 4.2.4.3.10 Generic Provide Operator Interface 4.2.6.3.12 Provide Operator Interface to UTM 4.2.10.3.11 Generic Provide Operator Interface	Operator SEO-Core-Config-Mod SEO-Core-Controller SEO-Core-Config SEO-Other-Core-Config SEO-System-Log
SEO-Core-Controller	This SEO provides control over the operating state and mode of all other Core SEOs.	4.2.2.3.12 Modify DD Operational State 4.2.4.3.18 Modify DD Operational State 4.2.4.3.19 Modify UP Operational State 4.2.5.3.5 Generic Modify Subsystem Operational State 4.2.5.3.10 Modify SM Operational State 4.2.5.3.11 Modify TS Operational State 4.2.6.3.10 Modify UTM Operational State 4.2.7.3.14 Modify C2C Operational State 4.2.7.3.15 Modify MM Operational State 4.2.7.3.16 Modify UTM Operational State 4.2.9.3.7 Modify NS Operational State 4.2.10.3.6 Core Takeover 4.2.10.3.15 Modify C2C Operational State 4.2.10.3.22 Request Core Takeover	SEO-Operator-Interface SEO-Health-Monitor SEO-Core-Config-Mod SEO-Core-Config SEO-System-Log

Name	Description	Functional Objects Implemented	Interfaces
SEO-User-Registry	This SEO maintains the permissions of users of the Core System, including Operators, Other Cores and System Users.	4.2.2.3.4 Check User Permission 4.2.3.3.3 Check User Permission 4.2.3.3.14 User Permission Registry 4.2.4.3.3 Application Permission Registry 4.2.4.3.4 Check User Permission 4.2.5.3.3 Check User Permission 4.2.5.3.21 User Permission Registry 4.2.6.3.3 Application Permission Registry 4.2.6.3.4 Check User Permission 4.2.6.3.15 User Permission Registry 4.2.7.3.4 Check User Permission 4.2.7.3.25 User Permission Registry 4.2.10.3.4 Check User Permission	SEO-User-Registry-Mod SEO-Operator-Interface SEO-Core-Config SEO-System-Log SEO-Core-Config-Mod SEO-System-Log
SEO-Core-Config	This SEO maintains the configuration of the Core, including startup and operating parameters, allocation of SEO's to HEO's, and configuration of devices used by the Core, including Geo-cast devices.	4.2.2.3.5 Data Acceptance Catalog 4.2.2.3.7 Geo-cast Device Catalog 4.2.3.3.5 [Subsystem] Configuration / Generic Subsystem Configuration 4.2.4.3.8 Data Acceptance Catalog 4.2.4.3.11 Geo-cast Device Catalog 4.2.6.3.16 User Trust Management Configuration 4.2.7.3.9 Geo-cast Device Catalog 4.2.7.3.26 User Trust Management Configuration 4.2.10.3.7 Data Acceptance Catalog	SEO-Core-Config-Mod SEO-Other-Core-Config SEO-User-Registry-Mod SEO-MR-Log SEO-System-Log

Name	Description	Functional Objects Implemented	Interfaces
SEO-Other-Core-Config	This SEO exchanges configuration information with other Cores, examines it against this Core's configuration, and identifies conflicts. It also receives conflict information provided by other Cores.	4.2.3.3.6 Exchange Configuration Info with Other Cores 4.2.3.3.8 Identify Core Conflicts 4.2.3.3.11 Other Core Configs 4.2.3.3.13 Receive Core Conflict Info 4.2.4.3.21 Other Core Configs	Other Core SEO-Operator-Interface SEO-Core-Config SEO-System-Log
SEO-Core-Cert	This SEO obtains the Core's X.509 certificate and CRLs published by the certificate provider and other CAs. It also obtains and maintains the keys used by the Core to encrypt and sign messages. While all encryption of remotely received messages is handled at the Core Decryptor, signing of messages produced by the Core is done at every SCN; this object distributes the keys necessary to sign messages, but holds the keys related to encryption.	4.2.3.3.9 Maintain Core X.509 Certificate 4.2.7.3.18 Obtain CRLs	ESS X.509 CA SEO-CRL SEO-Core-Config SEO-System-Log SEO-Local-Signer

Name	Description	Functional Objects Implemented	Interfaces
SEO-Local-Signer	This SEO maintains a key and uses it to sign messages output from the SCN that it operates on. It obtains the key from SEO-Core-Cert.	4.2.2.3.2.3 Sign Messages 4.2.3.3.2.3 Sign Messages 4.2.4.3.2.3 Sign Messages 4.2.5.3.2.3 Sign Messages 4.2.6.3.2.3 Sign Messages 4.2.7.3.2.3Sign Messages 4.2.9.3.2.3 Sign Messages 4.2.10.3.2.3 Sign Messages	SEO-Core-Cert SEO-Core-Config SEO-System-Log
SEO-Core-Config-Mod	Enables the modification of Core configuration objects.	4.2.3.3.4 Configure [Subsystem] Objects 4.2.3.3.10 Manually Modify Other Core Configs 4.2.4.3.5 Configure Geo-cast Device Information 4.2.4.3.14 Manually Modify User Permissions 4.2.4.3.13 Manage System User Data Subscriptions 4.2.4.3.17 Modify Data Acceptance Catalog 4.2.7.3.6 Disable Misbehaving Geo-Cast Device 4.2.10.3.5 Configure Geo-cast Device Information 4.2.10.3.9 Generic Configure Subsystem 4.2.10.3.12 Generic Subsystem Configuration 4.2.10.3.16 Modify Data Acceptance Catalog	SEO-Operator-Interface SEO-Core-Controller SEO-Core-Config SEO-User-Registry Field Node Owner Operator SEO-Backup-Store SEO-System-Log SEO-Backup-to-Other
SEO-User-Registry-Mod	This SEO accepts requests from the System User and End User Application Deployer for modifications to the SEO-User-Registry. It also accepts requests from the SEO-MR-Log to change permissions based on misbehavior report analysis.	4.2.4.3.20 Modify User Permissions 4.2.4.3.16 Modify Application Permissions 4.2.6.3.9 Modify User Permissions 4.2.7.3.23 Revoke Operator Permissions 4.2.7.3.24 Revoke User Permissions	SEO-User-Registry System User End User Application Deployer SEO-MR-Log SEO-Core-Config SEO-System-Log

Name	Description	Functional Objects Implemented	Interfaces
SEO-MR-Log	This SEO receives misbehavior reports from System Users and internal Core objects, maintains logs of those reports and performs analysis on those logs. It also requests revocation of credentials and/or permissions for offending System Users and Operators from SEO-User-Registry-Mod, SEO-CRL and external CAs.	4.2.2.3.11 Misbehavior Reports Log 4.2.4.3.15 Misbehavior Reports Log 4.2.7.3.3 Ask CA to Revoke 4.2.7.3.8 Exchange Misbehavior Reports with other Cores 4.2.7.3.13 Misbehavior Reports Log 4.2.7.3.21 Receive Internal Misbehavior Reports 4.2.7.3.22 Receive System User Misbehavior Reports 4.2.7.3.10 Identify Misbehaving Operators 4.2.7.3.11 Identify Misbehaving System Users 4.2.8.3.10 Receive Internal Misbehavior Reports 4.2.9.3.6 Misbehavior Reports Log	SEO-User-Registry-Mod SEO-CRL SEO-Core-Config SEO-System-Log Other Cores External CA
SEO-System-Log	This SEO maintains a record of Core performance, state changes and events, and the state of other Cores that it monitors. It provides a mechanism for extracting performance information for the Core Certification Authority.	4.2.4.3.7 Core Register to Receive Status from Other Core 4.2.5.3.4 Event Log 4.2.5.3.19 System State and Performance Log 4.2.5.3.13 Monitor Core Services Performance 4.2.5.3.14 Monitor Status of other Cores 4.2.5.3.15 Provide Record of System Performance	All other SEOs Core Certification Authority SEO-Core-Config SEO-System-Log
SEO-Health-Monitor	This SEO monitors the health and integrity of the Core and its operating environment, and takes action if the Core's performance is in jeopardy.	4.2.5.3.12 Monitor Core Health and Safety 4.2.5.3.20 Take Action in Response to Environment Issue 4.2.9.3.8 Monitor Service Control Node Performance	SEO-Core-Controller SEO-Core-Config SEO-System-Log

Name	Description	Functional Objects Implemented	Interfaces
SEO-Time	This SEO acquires time sync from an external source and provides it to all Core SEOs and HEOs.	4.2.5.3.8 Get Time from External Available Source 4.2.5.3.9 Make Time Available to All Subsystems 4.2.5.3.7 Generic Service Component	All other SEOs SEO-Core-Config SEO-System-Log
SEO-Service-Status-Dist	This SEO maintains a list of recipients of service status information, provides an interface for System Users and Cores to request service status information and distributes service status information.	4.2.4.3.6 Core Register to Receive Status 4.2.4.3.23 System User Register to Receive Status 4.2.4.3.22 Service Status Distribution Catalog 4.2.5.3.16 Provide Service Status to other Cores 4.2.5.3.17 Provide Service Status to System Users 4.2.5.3.18 Service Status Distribution Catalog	Other Core System User SEO-Core-Config SEO-System-Log
SEO-Data-Provision	This SEO provides an interface to System Users for them to discover what data this Core, and other Cores and 3rd parties this Core knows about, handles.	4.2.4.3.12 Manage Data Provision Requests	System User SEO-Core-Config SEO-MR-Log SEO-System-Log
SEO-Data-Acquirer	This SEO receives data from System Users and passes it on to the SEO-Parser or SEO-Scheduler, as appropriate.	4.2.2.3.16 Receive Data from System Users 4.2.2.3.2.4 Verify Authenticity of Received Messages	System User SEO-MR-Log SEO-Parser SEO-Scheduler SEO-Core-Config SEO-System-Log SEO-Data-SubLog

Name	Description	Functional Objects Implemented	Interfaces
SEO-Data-Scheduler	This SEO accepts messages for Geo-Cast and maintains a schedule of when to provide data to SEO-Data-Matcher	4.2.2.3.9 Manage Geo-cast Messages	SEO-Data-Acquirer SEO-Data-GeoLog SEO-Data-Matcher SEO-Core-Config SEO-System-Log
SEO-Data-GeoLog	This SEO logs all geo-cast messages.	4.2.2.3.8 Geo-cast Message Log	SEO-Data-Scheduler SEO-Core-Config SEO-System-Log
SEO-Data-SubLog	This SEO maintains the list of System Users and the data they are subscribed to, and also provides a means for changing subscriptions, both by System Users and by the Operator.	4.2.2.3.6 Data Subscription Catalog 4.2.4.3.9 Data Subscription Catalog 4.2.4.3.13 Manage System User Data Subscriptions 4.2.10.3.17 Modify System User Data Subscriptions 4.2.10.3.8 Data Subscription Catalog	SEO-Data-Acquirer SEO-Data-Sampler SEO-Data-Parser SEO-Data-Repackager SEO-Aggregator SEO-Health-Monitor SEO-Data-Matcher SEO-Core-Config SEO-System-Log System User Operator
SEO-Data-Sampler	Implements the Sample Data Functional Object.	4.2.2.3.18 Sample Data	SEO-Data-SubLog SEO-Data-Parser SEO-Data-Matcher SEO-Core-Config SEO-System-Log

Name	Description	Functional Objects Implemented	Interfaces
SEO-Data-Parser	Implements the Parse Data Functional Object.	4.2.2.3.13 Parse Data	SEO-Data-Acquirer SEO-Data-SubLog SEO-Data-Sampler SEO-Data-Aggregator SEO-Data-Repackager SEO-Core-Config SEO-System-Log
SEO-Data-Repackager	Implements the Repackage Data Functional Object.	4.2.2.3.17 Repackage Data	SEO-Data-Parser SEO-Data-SubLog SEO-Data-Aggregator SEO-Core-Config SEO-System-Log
SEO-Data-Aggregator	Implements the Aggregate Data Functional Object.	4.2.2.3.3 Aggregate Data	SEO-Data-SubLog SEO-Data-Parser SEO-Data-Repackager SEO-Core-Config SEO-System-Log
SEO-Data-Matcher	Provides data to System Users that have subscribed to it.	4.2.2.3.10 Match Data to Data Subscribers 4.2.2.3.14 Provide Data to Subscribing System Users	SEO-Data-SubLog SEO-Data-Sampler SEO-Data-Scheduler SEO-Core-Config SEO-System-Log System User

Name	Description	Functional Objects Implemented	Interfaces
SEO-Cert-Store	Provides an interface to System Users to acquire a digital certificate, either by granting it or telling the System User who does grant certificates for them. Also informs the ESS DSRC RA of special permissions that System Users are entitled to.	4.2.6.3.11 Provide Credentials 4.2.6.3.8 Maintain Credentials 4.2.6.3.13 Provide Special Permissions	System User SEO-User-Registry SEO-Core-Config SEO-System-Log
SEO-CRL	Maintains and distributes CRLs.	4.2.6.3.6 CRL Storage 4.2.6.3.7 Distribute CRL 4.2.7.3.5 CRL Storage 4.2.7.3.12 Manage CRL 4.2.7.3.7 Exchange CRLs with other Cores	SEO-CRL-Dist-List SEO-Core-Config SEO-Core-Cert SEO-MR-Log SEO-System-Log System User Other Core
SEO-CRL-Dist-List	Provides means for System Users to register to receive CRLs.	4.2.6.3.5 CRL Distribution List 4.2.6.3.14 System User Register to Receive CRL	System User SEO-Core-Config SEO-CRL SEO-System-Log
SEO-Backup-Store	Maintains backups of configuration data from other Cores.	4.2.10.3.3 Backup Other Core Data 4.2.10.3.19 Other Core Data Backups 4.2.10.3.20 Provide Other Core Data	SEO-Core-Config-Mod SEO-Core-Config SEO-System-Log

Name	Description	Functional Objects Implemented	Interfaces
SEO-Backup-to-other	Provides data to other Cores for backup purposes.	4.2.10.3.10 Generic Core Data Store 4.2.10.3.14 Get Backed Up Data 4.2.10.3.21 Provide Data to be Backed Up	SEO-Core-Config-Mod SEO-Core-Config SEO-System-Log
Local Decryptor	This SEO receives messages encrypted by the Decryptor SEO. It uses the local encryption algorithm to decrypt the message and pass it to the appropriate Service Component.	4.2.8.3.5 Decrypt Locally Encrypted Message 4.2.8.3.6 Maintain Core Local Key	SEO-Local-Encryptor SEO-Core-Config SEO-System-Log
Authenticity Verifier	This SEO verifies the authenticity of messages. While nominally a User Trust Management function, this is called out separately because it must reside on every SCN.	4.2.2.3.2.4 Verify Authenticity of Received Messages 4.2.3.3.2.4 Verify Authenticity of Received Messages 4.2.4.3.2.4 Verify Authenticity of Received Messages 4.2.5.3.2.4 Verify Authenticity of Received Messages 4.2.6.3.2.4 Verify Authenticity of Received Messages 4.2.7.3.2.4 Verify Authenticity of Received Messages 4.2.8.3.1.2 Verify Authenticity of Received Messages 4.2.9.3.2.4 Verify Authenticity of Received Messages 4.2.10.3.2.4 Verify Authenticity of Received Messages	SEO-Core-Config SEO-System-Log

Name	Description	Functional Objects Implemented	Interfaces
SEO-Local Encryptor	This SEO implements an encryption algorithm known to at least one other Core SEO. It receives messages from an SEO Operating on the same Node, encrypts those messages and then passes them to the SEO that the message is intended for. There are several Local Encryptors, one for each local encryption algorithm.	4.2.2.3.2.2 Encrypt Messages 4.2.3.3.2.2 Encrypt Messages 4.2.4.3.2.2 Encrypt Messages 4.2.5.3.2.2 Encrypt Messages 4.2.6.3.2.2 Encrypt Messages 4.2.7.3.2.2 Encrypt Messages 4.2.9.3.2.2 Encrypt Messages 4.2.10.3.2.2 Encrypt Messages	SEO-Core-Config SEO-System-Log
SEO-Encryptor	This SEO encrypts data intended for an external actor using the recipients public-key.	4.2.2.3.2.2 Encrypt Messages 4.2.3.3.2.2 Encrypt Messages 4.2.4.3.2.2 Encrypt Messages 4.2.5.3.2.2 Encrypt Messages 4.2.6.3.2.2 Encrypt Messages 4.2.7.3.2.2 Encrypt Messages 4.2.9.3.2.2 Encrypt Messages 4.2.10.3.2.2 Encrypt Messages	SEO-Core-Config SEO-System-Log

4.3.2.3.3 Service Router

This is the device that scans all incoming messages and determines what SCN to provide the message to, depending on the loading of the various SCNs.

Table 4-6: Service Router Node Engineering Objects

Name	Description	Functional Objects Implemented	Interfaces
SEO-Service-Router	Provides routing between Service Router and SEOs on SCNs.	4.2.9.3.13 Route Data/Request	SEO-Health-Monitor All SEOs with external interfaces SEO-Intrusion-Prevention SEO-Core-Config
SEO-Core-Config	This SEO maintains the configuration of the Core, including startup and operating parameters, allocation of SEO's to HEO's, and configuration of devices used by the Core, including Geo-cast devices.	4.2.2.3.5 Data Acceptance Catalog 4.2.2.3.7 Geo-cast Device Catalog 4.2.3.3.5 [Subsystem] Configuration / Generic Subsystem Configuration 4.2.4.3.8 Data Acceptance Catalog 4.2.4.3.11 Geo-cast Device Catalog 4.2.6.3.16 User Trust Management Configuration 4.2.7.3.9 Geo-cast Device Catalog 4.2.7.3.26 User Trust Management Configuration 4.2.10.3.7 Data Acceptance Catalog	SEO-Core-Config-Mod SEO-Other-Core-Config SEO-User-Registry-Mod SEO-MR-Log SEO-System-Log
SEO-System-Log	This SEO maintains a record of Core performance, state changes and events, and the state of other Cores that it monitors. It provides a mechanism for extracting performance information for the Core Certification Authority.	4.2.4.3.7 Core Register to Receive Status from Other Core 4.2.5.3.4 Event Log 4.2.5.3.12 Monitor Core Health and Safety 4.2.5.3.13 Monitor Core Services Performance 4.2.5.3.14 Monitor Status of other Cores 4.2.5.3.15 Provide Record of System Performance	All other SEOs Core Certification Authority SEO-Core-Config SEO-System-Log

Name	Description	Functional Objects Implemented	Interfaces
SEO-Health-Monitor	This SEO monitors the health and integrity of the Core and its operating environment, and takes action if the Core's performance is in jeopardy.	4.2.5.3.19 System State and Performance Log 4.2.5.3.20 Take Action in Response to Environment Issue 4.2.9.3.8 Monitor Service Control Node Performance	SEO-Core-Controller SEO-Core-Config SEO-System-Log

4.3.2.3.4 Core Access Node

This is the gateway for the Core System. In addition to connectivity, the Core Access Node includes an Intrusion Prevention/Detection System, Firewall and Router.

Table 4-7: Core Access Node Engineering Objects

Name	Description	Functional Objects Implemented	Interfaces
SEO-Conn	Provides network connectivity	4.2.9.3.9 Provide Internet Connectivity 4.2.9.3.10 Provide Private Network Connectivity 4.2.9.3.12 Route Data between Networks	All external SEO-Core-Config
SEO-Intrusion-Prevention	Provides Intrusion Prevention and detection functionality.	4.2.9.3.4 Intrusion Detection 4.2.9.3.5 Intrusion Prevention	SEO-Conn SEO-Core-Config
SEO-Access-Health-Perf	This SEO monitors the health and performance of the Core Access Node and the SEOs operating on it. It is noted separately from SEO-Health-Monitor and SEO-System-Log because the Core Access Node is a specialized networking device and not a general purpose computer.	4.2.5.3.12 Monitor Core Health and Safety 4.2.5.3.13 Monitor Core Services Performance	SEO-Operator-Interface SEO-System-Log SEO-Core-Config

4.3.2.3.5 Core Switch

There are two Core Switches; each is a network switching device providing connectivity between Core Nodes. One provides connectivity between Core Access Node, Service Router and Service Component Nodes. The second provides connectivity between the Core Decryptor and Service Component Nodes.

Table 4-8: Core Switch Engineering Objects

Name	Description	Functional Objects Implemented	Interfaces
SEO-Switch-Health-Perf	This SEO monitors the health and performance of the Core Switch. It is noted separately from SEO-Health-Monitor and SEO-System-Log because the Core Switch is a specialized networking device and not a general purpose computer.	4.2.5.3.12 Monitor Core Health and Safety 4.2.5.3.13 Monitor Core Services Performance	SEO-Operator-Interface SEO-System-Log SEO-Core-Config

Links:

Connections between Service Component Node(s), Service Router and Core Access Node are physical network connections on the same Local Area Network (LAN). Connections between SCNs and the Core Decryptor are through a separate LAN. Performance of these network connections will depend on the scope of the Core System, chiefly the amount of data handled by Data Distribution SEOs.

4.3.2.4 View Description

Figure 4-19 shows the base Core Function Allocation Connectivity View. The illustrated case assumes that a single Service Component Node is sufficient to operate the majority of Core SEOs. The 'SEO' prefix in the previous tables is dropped in the illustration to save space. Information Objects do not appear on this view because including them would make the diagram hard to read due to clutter. The Information Objects passed between SEOs are the sum of the Information Objects passed by the Functional Objects implemented by those SEOs.

All data coming to and sent by the Core System passes through the Core Access Node. This is a commercially available router with integrated IPS and management modules. The Core Access Node sends all data that passes the IPS to the Service Router through a Core Switch. The Service Router forwards data to the appropriate Service Component Node. In Figure 4-19 there is only one option, but Figure 4-20 illustrates an implementation with a second SCN, where the first SCN operates most of the housekeeping and low-overhead SEOs, while a second SCN operates all Data Distribution SEOs except for Data Provision.

Figure 4-21 takes this a step further with an implementation where the Data-Acquirer and Data-Parser SEOs are partitioned into another SCN, offloading the processing of the Parser process from the Data Distribution Node.

Figure 4-22 shows a variation where three SCNs are dedicated to operating Data-Acquirer and Data-Parser SEOs. In this variation the Service-Router would apportion incoming data between the three SCNs. Additional variations are possible, separating other high impact SEOs (e.g., Data-Aggregator, Data-Sampler) onto separate Nodes. This allows the Core to scale to support ever larger amounts of data input. It also allows the Core to increase throughput and/or decrease response time to support the demands of System User applications. An analysis of application performance requirements cannot be completed at this time, as such requirements have not been published. The intent of this architecture is to provide sufficient flexibility, scalability and evolvability that when those requirements are provided, a Core System could be specified that would satisfy reasonable application requirements.

The ability to place SEOs on different SCNSs not only helps scalability, but failure mitigation. If one Node fails, another can be brought online to replace it. A similar approach to redundancy could be taken with regard to the Core Decryptor, Housekeeping SCN and Service Router, where one backup Node could be ready to take over for any of these should a failure be detected.

The one area where the Core is at risk is with regard to its networking components. The Core Access Node and Core Switches are individual single points of failure. The Core Switches are compact but potentially expensive devices for Cores distributing large amounts of data. Spares could be kept ready in case of failure, which has a significant capital cost but costs little in maintenance. The Core Access Node will be more sophisticated and expensive; maintaining a spare of this device could be a significant expense. Short turn-around onsite maintenance is an alternative.

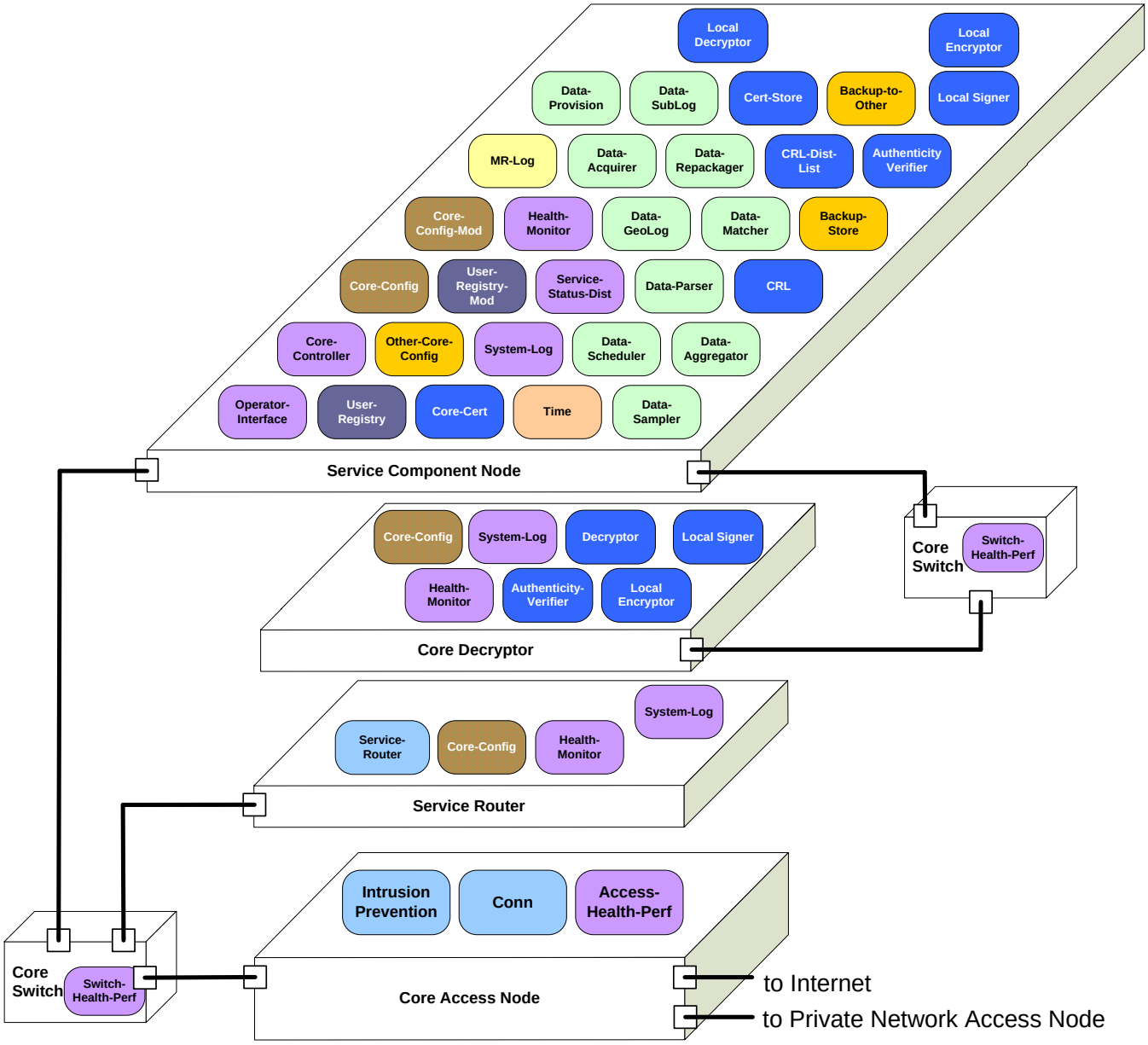


Figure 4-19: Connectivity View – Node Functional Allocation, one SCN

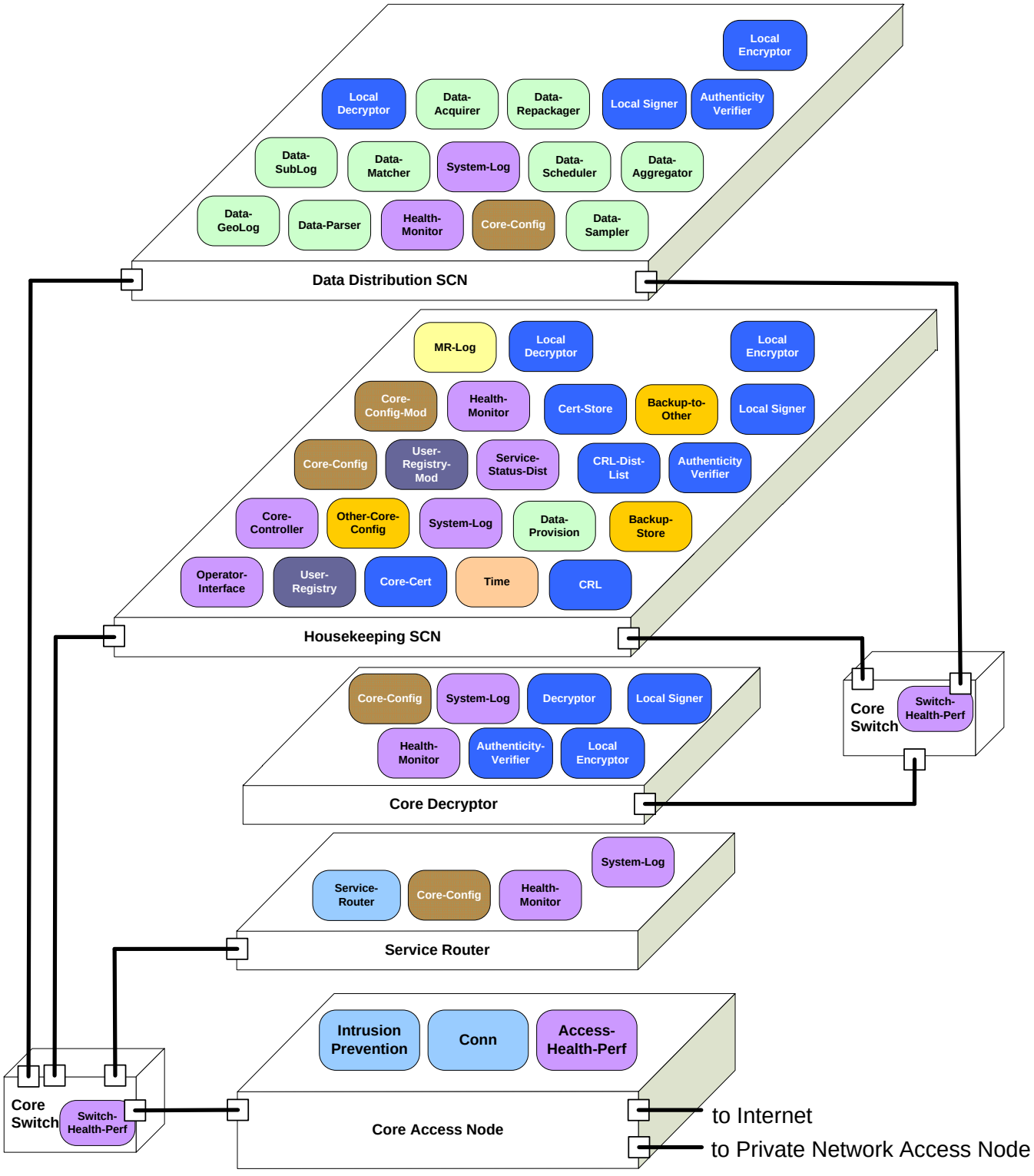


Figure 4-20: Connectivity View – Node Functional Allocation, Separate DD SCN

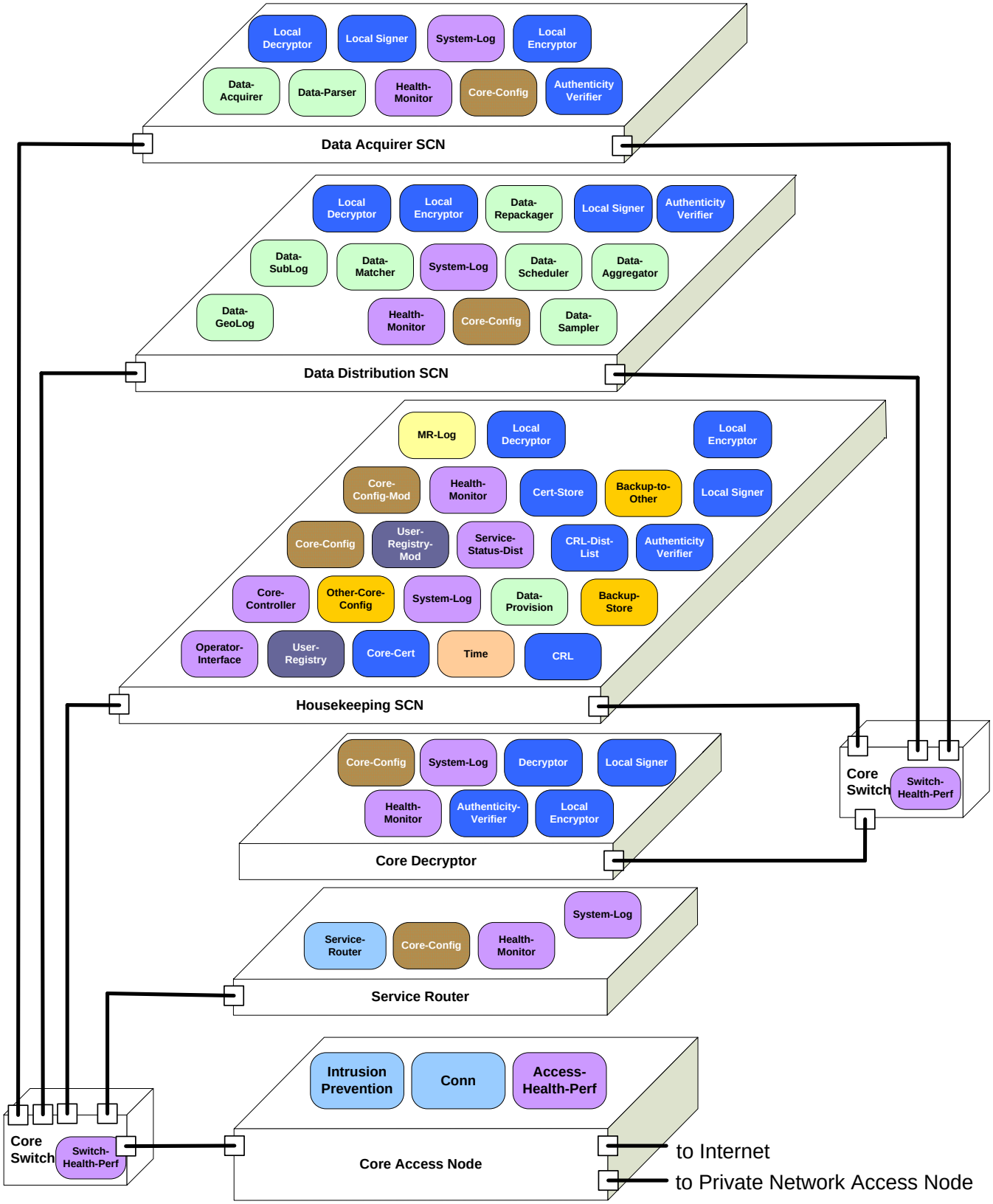


Figure 4-21: Connectivity View – Node Functional Allocation, Separate DD, Acquirer SCN

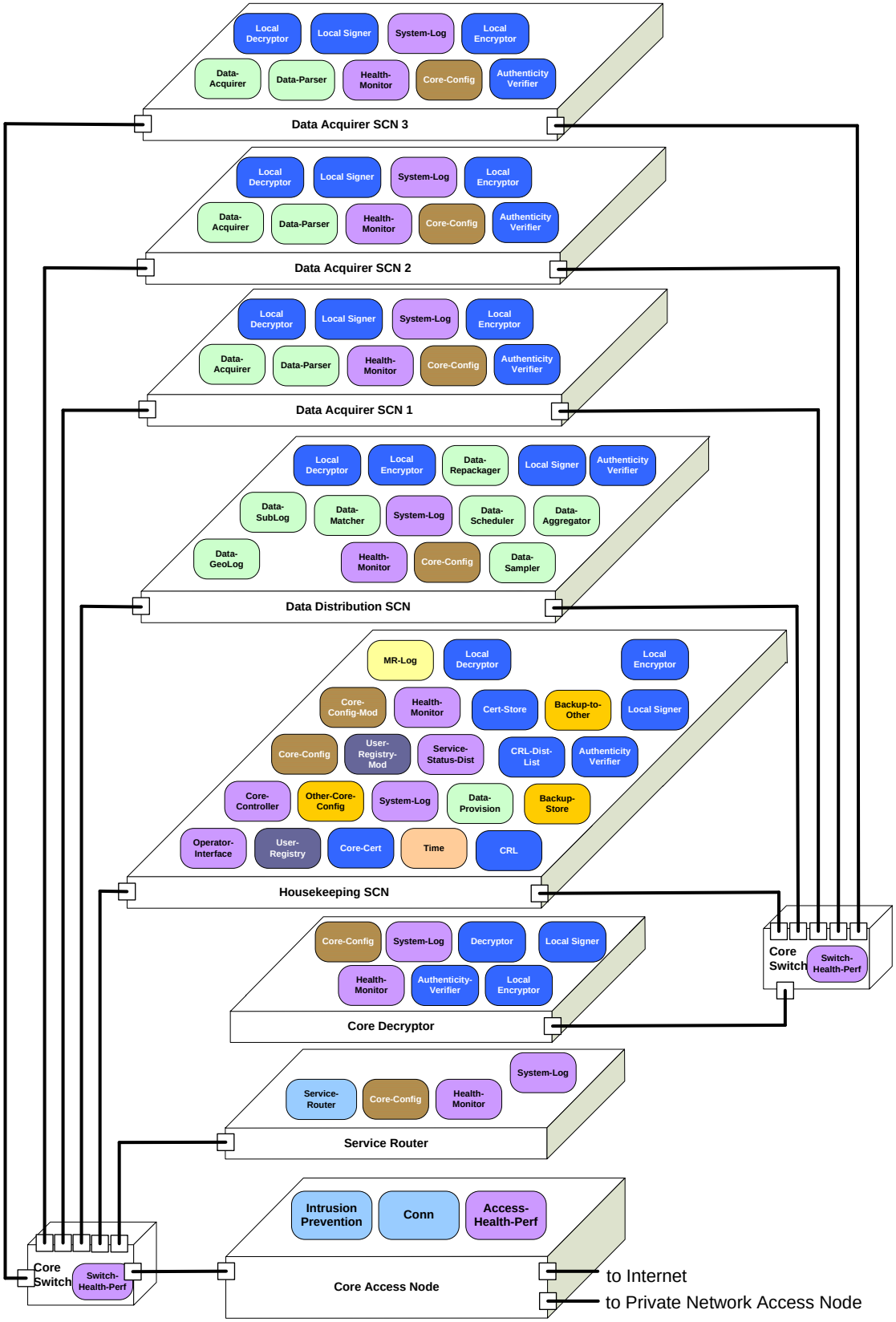


Figure 4-22: Connectivity View – Core System Function Allocation, Multiple Data Acquirer SCNs

4.3.2.5 Configuration Information

The following views must be considered when making changes to this view:

Functional Viewpoint: Functional View – Data Distribution
Functional Viewpoint: Functional View – System Configuration
Functional Viewpoint: Functional View – User Configuration
Functional Viewpoint: Functional View – System Monitor and Control
Functional Viewpoint: Functional View – Credentials Distribution
Functional Viewpoint: Functional View – Misbehavior Management
Functional Viewpoint: Functional View – Core Decryption
Functional Viewpoint: Functional View – Networking
Functional Viewpoint: Functional View – Core Backup
Connectivity Viewpoint: Connectivity View – High Level
Communications Viewpoint: Communications View – Mobile DSRC Device and Core
Communications Viewpoint: Communications View – Mobile Wide-Area Wireless User and Core
Communications Viewpoint: Communications View – Fixed Point Center/Field User and Core, Core2Core
Communications Viewpoint: Communications View – Core Routing

4.3.3 Connectivity View – State and Mode Transitions

4.3.3.1 Introduction

While states and state transitions were discussed in the Functional View – Top Level, the Core will not be implemented as a set of Functional Objects. It will be implemented using HEOs and SEOs. The states and modes those HEOs and SEOs operate in and transition between are the subject of this view.

4.3.3.2 Concerns Addressed by this View

Performance	Can the Core meet all of the performance requirements defined in the SyRS (e.g., availability, reliability, capacity and other quantitative measures)?
Risks	Is the Core System's hardware and software architecture susceptible to failure, and if so under what circumstances? What are the characteristics of this failure?
Maintainability	Is the structure of the Core System maintainable with a reasonable allocation of resources for the entities that are likely to consider deployment? Can the Core's functionality be sustained with acceptable levels of downtime as per the SyRS?

4.3.3.3 Object Definitions and Roles

SEOs and HEOs are discussed as groups. No specific objects are referenced in this view.

4.3.3.4 View Description

While within the Standby and Operational states, an Engineering Object may be in one of five modes, as illustrated in Figure 4-23:

- Normal mode: In the normal mode, there is little or no functional or performance impacts on the ability of the object to provide its services.
- Degraded mode: In the degraded mode, the object is impaired to a significant extent: its ability to provide services is greatly reduced or eliminated completely. Degraded mode is a reflection of conditions; it is not a mode that is voluntarily entered. An HEO may enter degraded mode if it starts to overheat and most reduce performance to manage the heat load for example. An SEO may enter degraded mode if the HEO it is operating on enters degraded mode, or if a software bug decreases performance.
- Restricted mode: In the restricted mode, the ob-

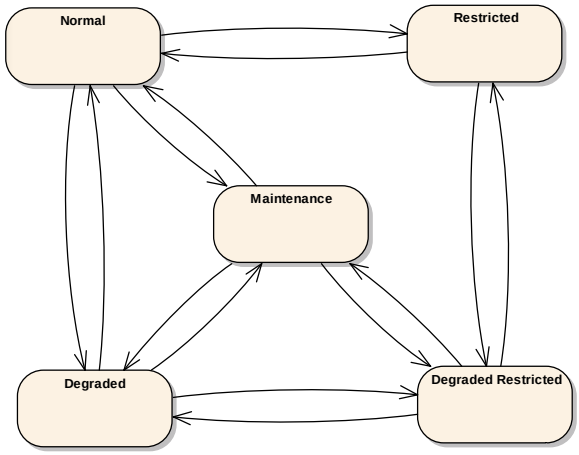


Figure 4-23: Standby and Operational Modes

ject is capable of performing as expected; however certain services or features are disabled to support a specific event such as an evacuation. The form of the restriction is determined by the Developer, System Manager and System Deployer, and implemented as part of the configuration of the object. Restricted Mode is entered when the Operator commands the object to enter it or when object performance reaches a pre-defined threshold. In a restricted mode, an object could curtail the use of particular functions to privileged users. For instance, SEO-Service-Status-Dist might cease responding to System User requests for status updates, but still respond to other Cores. Specification of what changes each object will undergo when in restricted mode is left for the design phase.

- Degraded/Restricted Mode: If during the course of operating in a restricted mode there is a loss of functionality, or if while in degraded mode there is a need to enter restricted mode, the object may enter the degraded/restricted mode. This mode is a combination of the restricted and degraded modes.
- Maintenance Mode: The Operator may place an object in maintenance mode to replace an impaired object, to upgrade an object or certify the object. Depending on the nature of maintenance planned, the impact on the object's ability to provide services may be impacted. Also, its ability to manage itself and provide visibility into how it is performing may be impacted.

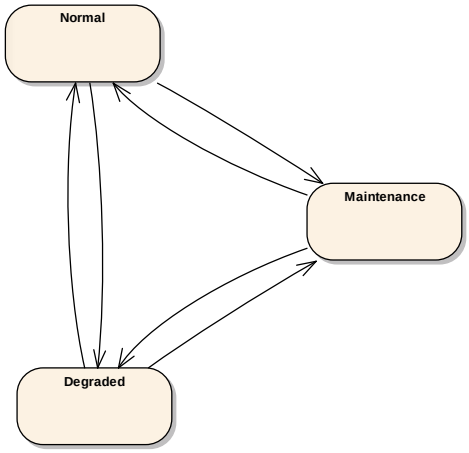


Figure 4-24: Training and Installation Modes

While within the Training and Installation states, each object may be in one of three modes, as illustrated in Figure 4-24. Definitions of these modes are the same as those defined above under Standby and Operational.

In order to quantify the availability performance measure, there must be a definition of what constitutes an available Core System. *A Core System is considered available if it delivers all of the services it offers*. So for example, if a Core offers all services described in this SAD, but the SEO-Parser fails, the Core is considered to be not available, even though all other objects continue to operate. If for example the SEO-Parser fails but another instantiation is started, then the Core was unavailable only during the time between the failure and the start of the new process.

The impact of maintenance mode on the availability metric will depend on the object. When an HEO is sent into maintenance mode it may not be able to offer any services; in this case another HEO offering identical services could be put online before the HEO was put into maintenance mode. In this way the Core would continue to be available.

4.3.3.5 Configuration Information

The following views must be considered when changing this view:

Functional Viewpoint: Functional View – Top Level

4.4 Communications Viewpoint

The Communications Views defined here address communications between Core and Mobile, Field and Center users as well as other Cores.

Four Communications Views are presented:

- Communications between the Core System and Mobile DSRC-equipped devices
- Communications between the Core System and Wide-Area-Wireless devices
- Communications between the Core System and Fixed-point devices
- Routing between devices connected to the Core by private networks

Table 4-9 shows which Communications Views are relevant to each stakeholder.

Table 4-9: Communications View Stakeholder Matrix

Communications View \ Stakeholder	Mobile User	Field User	Center User	Operator	Acquirer	Maintainer	Developer	Manager	Tester	Policy Setter	Application Developer	Device Developer	Service Provider
Mobile DSRC Device and Core	■	■		■	■	■	■	■	■	■	■	■	■
Mobile Wide-Area Wireless User and Core	■			■	■	■	■	■	■	■	■	■	■
Fixed Point Center/Field User and Core, Core2Core		■	■	■	■	■	■	■	■	■	■	■	■
Core Routing	■	■	■	■	■	■	■	■	■	■	■	■	■

The following Communications Views are relevant to each Stakeholder as shown in the Table above:

- Mobile Users: “Mobile DSRC Device and Core,” “Mobile Wide-Area Wireless User and Core,” and “Core Routing”
- Field Users: “Mobile DSRC Device and Core,” “Fixed Point Center/Field User and Core, Core2Core,” and “Core Routing”
- Center Users: “Fixed Point Center/Field User and Core, Core2Core,” and “Core Routing”
- Operators: all
- Acquirers: all
- Maintainers: all
- Developers: all
- Managers: all
- Testers: all
- Policy Setters: all
- Application Developers: all
- Device Developers: all
- Service Providers: all

4.4.1 Communications View – Mobile DSRC Device and Core

4.4.1.1 Introduction

In order for Mobile Users to communicate with the Core System, they must use some form of wireless communications. This is the first of two views exploring Mobile-Core communications; this view focuses on Mobile Users using DSRC. Since DSRC communication is by definition short range, an intermediary must accept the DSRC message and provide it to the Core. Current concepts for the *connected vehicle* environment focus on DSRC at 5.9 GHz as defined in the IEEE 1609.x and 802.11p standards; development of new DSRC technologies may require re-examination of this view.

4.4.1.2 Concerns Addressed by this View

Performance	Do the communications protocols allow the Core to meet the performance requirements defined in the SyRS?
Interfaces	Do communications protocols allow the Core to meet the interface requirements defined in the SyRS?
Functionality	What functionality exists in the communications protocols used by the Core? What reliability features are included in the communication protocols?
Security	What provisions for ensuring the privacy of communications by System Users are included in the communications protocols? How do the communications protocols provide non-repudiation of messages sent to and from the Core System? How do the communications protocols protect the integrity of messages sent to and from the Core System.
Organization/Resources	What resources are required to develop, deploy, operate and maintain the communications protocols that the Core System needs?
Appropriateness	Do the communications protocols fulfill the needs set forth in the ConOps and support applications that meet the overall *connected vehicle* goals?
Feasibility	Are the communications protocols required by the Core System feasible to specify, develop, and deploy?
Risks	If communications protocols chosen are developed by independent bodies are there any risks associated with changes to those standards that may impact the applicability of those standards? If communications protocols chosen are developed by independent bodies are there any risks associated with the timely publication of those standards?
Evolvability	Are the communications protocols chosen scalable to support foreseeable demands from System Users?

Deployability	Are the communications protocols required by the Core practical to deploy from a capital and human resource perspective?

4.4.1.3 Object Definitions and Roles

4.4.1.3.1 Mobile DSRC Device

This node is the Mobile User node equipped with a DSRC radio that needs to communicate with the Core System.

Application Mobile Component: This is the SEO that wants to communicate with the Core.
Session Layer Security: A session layer security protocol such as Transport Layer Security (TLS) or Datagram Transport Layer Security (DTLS).
Internet Protocols: TCP, UDP and IPv6
IEEE 1609.3: Defines WAVE Short Messages (WSM) the Wave Short Message Protocol (WSMP)
IEEE 1609.2: Defines security services for use over DSRC.
IEEE 802.2: Defines the Logical Link Control (LLC) layer
IEEE 1609.4: Defines the WAVE MAC layer
IEEE 802.11p: Defines the physical layer

The Mobile DSRC Device and DSRC Field Node communicate across a wireless medium.

4.4.1.3.2 DSRC Field Node

This field node provides DSRC connectivity to the Mobile User with backhaul capable of accessing the Core System. This could be through the Internet or through a private connection to the Core.

Field Mobile Component: This SEO is responsible for managing the communications exchange between the Mobile and the Core System. When the Mobile uses WSM, this object is responsible for the translation between WSM and Internet Protocols. Application architecture may dictate further responsibilities for this object. This object could be limited to protocol translation, but could also have higher layer application related functionality (e.g., data aggregation). Responsibility for development of this object is split between the Field Node Developer and End User Application Developer, depending on the Field Node architecture and policies and application architecture. (Field Node Developers may enable End User Application Developers to develop and install software on Field Nodes.)
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Internet Protocols: TCP, UDP and IPv6
IEEE 1609.3: Defines WAVE Short Messages WSM and WSMP
IEEE 1609.2: Defines security services for use over DSRC.
IEEE 802.2: Defines the LLC layer
IEEE 1609.4: Defines the WAVE MAC layer
IEEE 802.11p: Defines the physical layer
Low Level WAN: A variety of Wide Area Network (WAN) communications protocols using a variety of physical media are possible, including cellular, Worldwide Interoperability for Microwave Access (WiMAX), T1/T3

4.4.1.3.3 Core Access Node

This is the Core System Node that receives all System User communications, scans it for intrusion protection, and routes it to the appropriate destination.

Conn: The SEO responsible for providing connectivity between the Core and other networks.
Intrusion Prevention: The SEO responsible for implementing the Intrusion Prevention and Detection functions.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gigabit-per-second (Gbps).

4.4.1.3.4 Service Router

The Service Router receives all System User communications that are intended for a Core service and routes data to the appropriate Core SEO.

Service Router: The SEO responsible for implementing Service Router functions.
Intrusion Prevention and Detection: The SEO responsible for implementing the Intrusion Prevention and Detection function.
Intrusion Detection: The SEO responsible for implementing the Intrusion Detection function.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gbps.

4.4.1.3.5 Service Component Node

This is the Core System node that provides service functionality. Most Core SEOs operate on a Service Component Node.

Core Service Component: Representative of any Core System SEO that communicates with a Mobile User.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gbps.

4.4.1.3.6 Decryption Node

This is the Core System node that provides decryption of messages intended for the Core, re-encrypts them using local encryption keys, and sends the message data back to the appropriate SEO.

Core Local Encryptor: The SEO responsible for implementing the Core Local Encryption function.
Core Decryptor: The SEO responsible for implementing the Core Decryption function.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gbps.

4.4.1.4 View Description

Communications from a Mobile User using DSRC start with a Mobile User – Field Node interaction. If the communication is encrypted, a session layer of the protocol stack (e.g. DTLS) will be used, otherwise it will be bypassed (this is not shown on the diagram for clarity).

The Mobile User communicates with the Field Node using IP or WSM protocols. The Field Node communicates with the Core System using IP protocols. If the Mobile User communication was encrypted the Field Node will use a session layer encryption protocol to maintain a similar level of security. The Field Node is not as latency-concerned when communicating with the Core, so it could use TCP and thus TLS. Addressing the Core may start with a well-known hostname that is resolved by the Field Node's Domain Name Service server, or it may be stored locally on the Field Node.

All messages reach the Core through the Core Access Node. Messages are scanned by Intrusion Protection and Detection SEOs as they pass up the protocol stack, and again as they descend. Messages that are not blocked by Intrusion Protection are passed to the Service Router.

A message that arrives at the Core System's Service Router will be passed to the relevant SEO on an available Service Component Node. If the message is encrypted it may need to be passed to the Core Decryption Node, decrypted, re-encrypted and then returned.

The Core System uses IPv6 as the middle-layer protocols for all internal and external network interfaces. TCP offers acknowledgement and retransmit that improves overall reliability. IP includes unique message identification that can assist with non-repudiation detection. Replay attacks still require authenticity verification and the Intrusion Prevention SEO. Developers may opt to use UDP for some Core interfaces, sacrificing TCP's reliability for simplicity and potentially increased throughput. IPv6 includes more useful multicast features than IPv4; some interfaces may be built to leverage this. Multicast should decrease the processing requirements for Core SEOs that can use it.

Security, including both the protection of data and assurance of data integrity, is handled at multiple layers. Session-layer protocols such as TLS and DTLS can be used to secure message exchanges. IPsec can be used at the Internet layer. These protocols require significant processing and therefore more resources at source and destination. Individual messages can be encrypted at the application layer (the SEO); like session layer protocols this is expensive in terms of compute resources. Lower layer security has been a topic of the IEEE 1609.x standards, and may offer another option.

Message non-repudiation is handled by the use of a signature, a cryptographic hash based on the public-key infrastructure concepts in IEEE 1609.2. Assurance that a message is genuine, and in particular not a replay, relies on this function as well, though it assumes that a reasonable timestamp is provided as part of the message content. If timestamps are not included in the message, then replay attacks are vulnerability.

Most of the protocols suggested for use in this view are available and usable without new research. MAC-layer security for DSRC may require additional work. IPv6 equipment and software is substantially more available than it was during the VII period; reports from the Proof-of-Concept about the difficulty of acquiring IPv6 equipment should not be a concern.

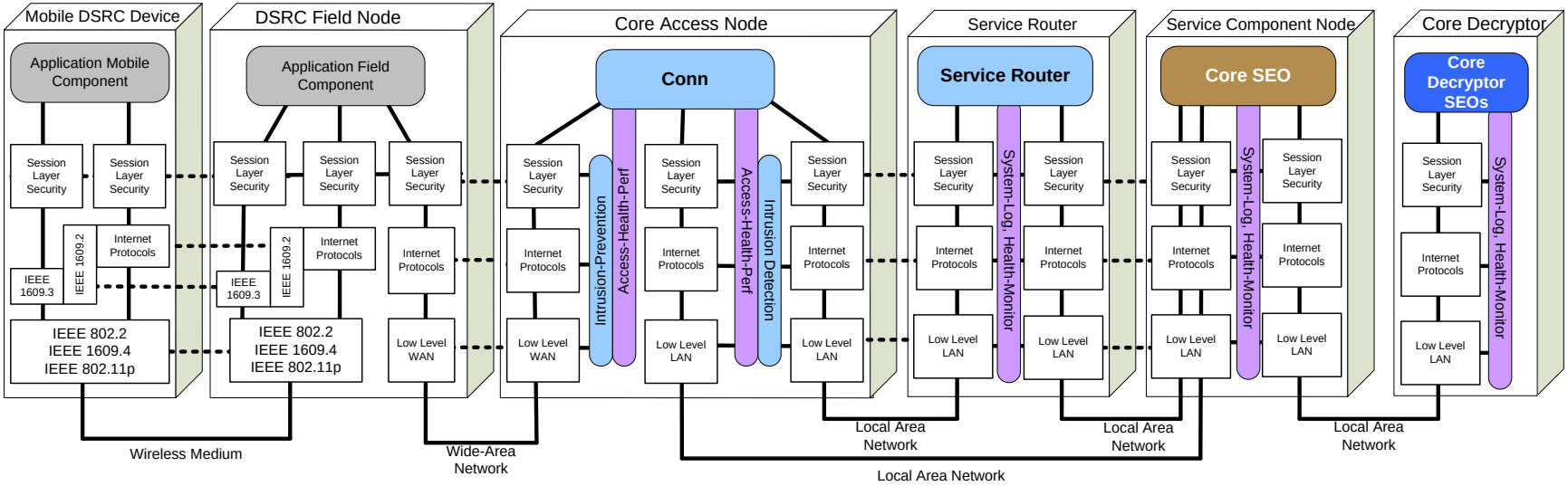


Figure 4-25: Communications View – DSRC Mobile to Core

Note that the Core Switch is not shown in this diagram. This is to reduce the number of objects on the drawing and make it easier to read. The Core Switch functions as part of the Local Area Network, and includes Low-Level LAN protocol boxes, as well as the Switch-Health-Perf SEO.

4.4.1.5 Configuration Information

The following views must be considered when making changes to this view:

Enterprise Viewpoint: Enterprise View – Business Model
Functional Viewpoint: Functional View – Core Decryption
Connectivity Viewpoint: Connectivity View – High Level
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Communications Viewpoint: Communications View – Mobile Wide-Area Wireless User and Core
Communications Viewpoint: Communications View – Fixed Point Center/Field User and Core, Core2Core
Communications Viewpoint: Communications View – Core Routing

4.4.2 Communications View – Mobile Wide-Area Wireless User and Core

4.4.2.1 Introduction

In order for Mobile Users to communicate with the Core System, they must use some form of wireless communications. This is the second of two views exploring Mobile-Core communications; this view focuses on Mobile Users using wide-area wireless communications. This includes Wi-Fi, cellular and other wireless communications.

Current cellular data communications are usually referred to as either 3G or 4G. 3G refers to 3rd generation mobile telecommunications meeting the International Mobile Telecommunications-2000 specifications. Current 3G systems typically provide peak bandwidth in the 200 kilobits per second (Kbps) to 1.5 megabits per second (Mbps) range.

Cellular systems marketed as 4G do not typically meet the 4G standard. 4G refers to 4th generation mobile telecommunications meeting the International Mobile Telecommunications Advanced (IMT-A) specifications. 4G requires 100 Mbps communications for mobile devices. Systems currently marketed as 4G are actually pre-4G as none of these systems (such as current WiMAX and Long Term Evolution (LTE) implementations) meet the 4G requirements. Supporting technologies continue to improve and it is expected that implementations of those technologies meeting 4G requirements will be available in a timeframe suitable for them to participate in the *connected vehicle* environment.

4.4.2.2 Concerns Addressed by this View

Performance	Do the communications protocols allow the Core to meet the performance requirements defined in the SyRS?
Interfaces	Do communications protocols allow the Core to meet the interface requirements defined in the SyRS?
Functionality	What functionality exists in the communications protocols used by the Core? What reliability features are included in the communication protocols?
Security	What provisions for ensuring the privacy of communications by System Users are included in the communications protocols? How do the communications protocols provide non-repudiation of messages sent to and from the Core System? How do the communications protocols protect the integrity of messages sent to and from the Core System.
Organization/Resources	What resources are required to develop, deploy, operate and maintain the communications protocols that the Core System needs?
Appropriateness	Do the communications protocols fulfill the needs set forth in the ConOps and support applications that meet the overall *connected vehicle* goals?

Feasibility	Are the communications protocols required by the Core System feasible to specify, develop, and deploy?
Risks	If communications protocols chosen are developed by independent bodies are there any risks associated with changes to those standards that may impact the applicability of those standards? If communications protocols chosen are developed by independent bodies are there any risks associated with the timely publication of those standards?
Evolvability	Are the communications protocols chosen scalable to support foreseeable demands from System Users?
Deployability	Are the communications protocols required by the Core practical to deploy from a capital and human resource perspective?

4.4.2.3 Object Definitions and Roles

4.4.2.3.1 Mobile Wide-Area Wireless Device

This node is the Mobile User node equipped with a wireless radio that needs to communicate with the Core System.

Application Mobile Component: This is the SEO that wants to communicate with the Core.
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Internet Protocols: TCP, UDP and IPv6
Data Link/Physical: Any of an assortment of protocols depending on the wide area wireless technology being used.

The Mobile Wide Area Wireless Device and Wide-Area Wireless Field Node communicate across a wireless medium.

4.4.2.3.2 Wide-Area Wireless Field Node

This is the field node providing wireless connectivity to the Mobile User. It has a backhaul providing connectivity to the Core System.

Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Internet Protocols: TCP, UDP and IPv6
Data Link/Physical: Any of an assortment of protocols depending on the wide area wireless technology being used.
Low Level WAN: A variety of wide area network communications media are possible, including 3G/4G cellular, WiMAX, T1/T3

The Mobile Wide Area Wireless Device and Wide-Area Wireless Field Node communicate across a wireless medium. The Wireless Field Node and the Core System (Service Router) communicate across a wired or wireless medium.

4.4.2.3.3 Core Access Node

This is the Core System Node that receives all System User communications, scans it for intrusion protection, and routes it to the appropriate destination.

Conn: The SEO responsible for providing connectivity between the Core and other networks.
Intrusion Prevention: The SEO responsible for implementing the Intrusion Prevention and Detection functions.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gbps.

4.4.2.3.4 Service Router

The Service Router receives all System User communications that are intended for a Core service and routes data to the appropriate Core SEO.

Service Router: The SEO responsible for implementing Service Router functions.
Intrusion Prevention and Detection: The SEO responsible for implementing the Intrusion Prevention and Detection function.
Intrusion Detection: The SEO responsible for implementing the Intrusion Detection function.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gbps.

4.4.2.3.5 Service Component Node

This is the Core System node that provides service functionality. Any Functional Object other than those assigned to decryption or service routing operates on a Service Component Node.

Core Service Component: Representative of any Core System SEO that communicates with a Mobile User.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gbps.

4.4.2.3.6 Decryption Node

The Core System node that provides decryption of messages intended for the core, re-encrypts them using local encryption keys, and sends the message data back to the Service Router.

Core Local Encryptor: The SEO responsible for implementing the Core Local Encryption function.
Core Decryptor: The SEO responsible for implementing the Core Decryption function.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gbps.

4.4.2.4 View Description

Communications from a Mobile User using wide-area wireless communications start with a Mobile User – Field Node interaction. If the communication is encrypted the session layer of the protocol stack will be used, otherwise it will be bypassed (this is not shown on the diagram for clarity). Addressing the Core may start with a well-known hostname that is resolved by the Field Node's Domain Name Service server, or it may be stored locally on the Mobile Device node.

The Field Node communicates with the Core System using IP protocols. If the Mobile User communication was encrypted the field node will use a session-layer encryption to maintain a similar level of security.

All messages reach the Core through the Core Access Node. Messages are scanned by Intrusion Protection and Detection SEOs as they pass up the protocol stack, and again as they descend. Messages that are not blocked by Intrusion Protection are passed to the Service Router.

A message that arrives at the Core System's Service Router will be passed to the relevant SEO on an available Service Component Node. If the message is encrypted it may need to be passed to the Core Decryption Node, decrypted, re-encrypted and then returned.

The Core System uses IPv6 as the middle-layer protocols for all internal and external network interfaces. TCP offers acknowledgement and retransmit that improves overall reliability. IP includes unique message identification that can assist with non-repudiation detection. Replay attacks still require authenticity verification and the Intrusion Prevention SEO. Developers may opt to use UDP for some Core interfaces, sacrificing TCP's reliability for simplicity and potentially increased throughput. IPv6 includes more useful multicast features than IPv4; some interfaces may be built to leverage this. Multicast should decrease the processing requirements for Core SEOs that can use it.

Security, including both the protection of data and assurance of data integrity, is handled at multiple layers. Session-layer protocols such as TLS and DTLS can be used to secure message exchanges. IPsec can be used at the Internet layer. All of these protocols require significant processing and therefore more resources at source and destination. Individual messages can be encrypted at the application layer (the SEO); like session and Internet layer protocols this is expensive in terms of compute resources.

Message non-repudiation is handled by the use of a signature, a cryptographic hash based on the public-key infrastructure concepts. Assurance that a message is genuine, and in particular not a replay, relies on this function as well, though it assumes that a reasonable timestamp is provided as part of the message content. If timestamps are not included in the message, then replay attacks are vulnerability.

Most of the protocols suggested for use in this view are available and usable without new research. IPv6 equipment and software is substantially more available than it was during the VII period; reports from the Proof-of-Concept about the difficulty of acquiring IPv6 equipment should not be a concern.

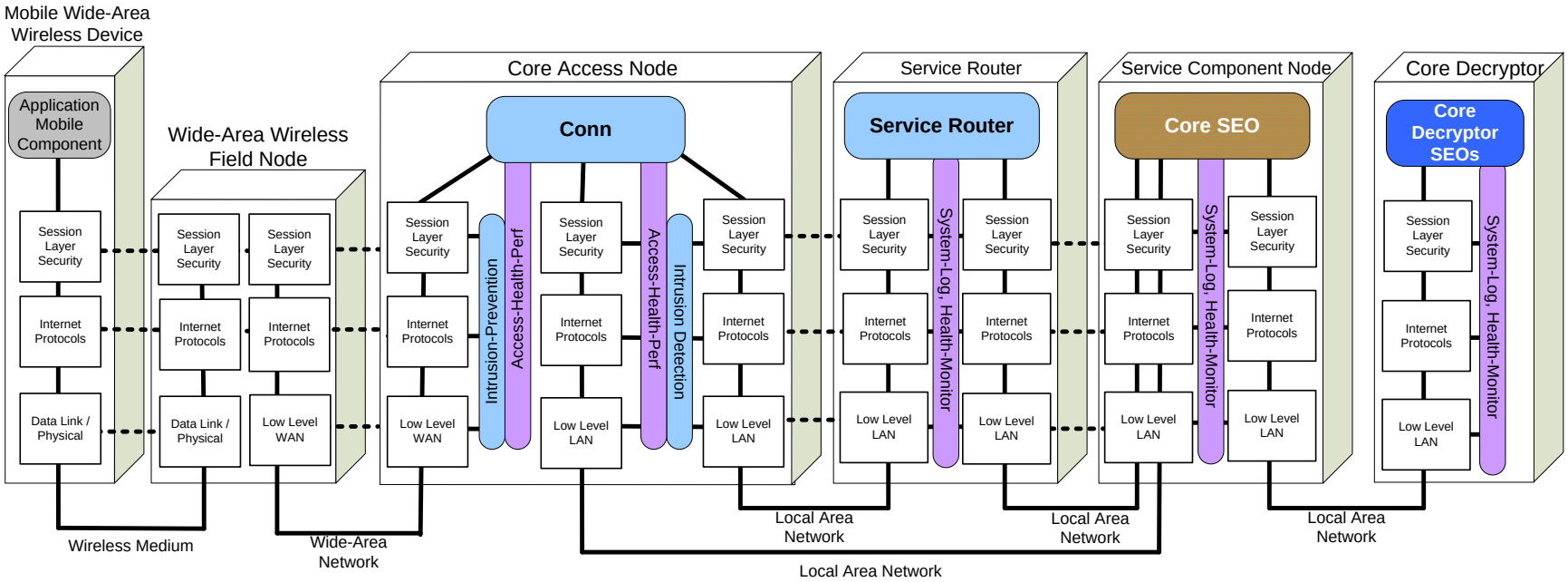


Figure 4-26: Communications View – Wide-Area Wireless and Core Communications

Note that the Core Switch is not shown in this diagram. This is to reduce the number of objects on the drawing and make it easier to read. The Core Switch functions as part of the Local Area Network, and includes Low-Level LAN protocol boxes, as well as the Switch-Health-Perf SEO.

4.4.2.5 Configuration Information

The following views must be considered when making changes to this view:

Enterprise Viewpoint: Enterprise View – Business Model
Functional Viewpoint: Functional View – Core Decryption
Connectivity Viewpoint: Connectivity View – High Level
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Communications Viewpoint: Communications View – Mobile DSRC Device and Core
Communications Viewpoint: Communications View – Fixed Point Center/Field User and Core, Core2Core
Communications Viewpoint: Communications View – Core Routing

4.4.3 Communications View – Fixed Point Center/Field User and Core, Core2Core

4.4.3.1 Introduction

Non-Mobile systems (Field, Center, External Support Systems and other Cores) that communicate with the Core System may do so using wired communications, either through the Internet or a private network. Non-Mobile systems may use wireless communications; for that case reference section 4.4.2.

4.4.3.2 Concerns Addressed by this View

Performance	Do the communications protocols allow the Core to meet the performance requirements defined in the SyRS?
Interfaces	Do communications protocols allow the Core to meet the interface requirements defined in the SyRS?
Functionality	What functionality exists in the communications protocols used by the Core? What reliability features are included in the communication protocols?
Security	What provisions for ensuring the privacy of communications by System Users are included in the communications protocols? How do the communications protocols provide non-repudiation of messages sent to and from the Core System? How do the communications protocols protect the integrity of messages sent to and from the Core System.
Organization/Resources	What resources are required to develop, deploy, operate and maintain the communications protocols that the Core System needs?
Appropriateness	Do the communications protocols fulfill the needs set forth in the ConOps and support applications that meet the overall *connected vehicle* goals?
Feasibility	Are the communications protocols required by the Core System feasible to specify, develop, and deploy?
Risks	If communications protocols chosen are developed by independent bodies are there any risks associated with changes to those standards that may impact the applicability of those standards? If communications protocols chosen are developed by independent bodies are there any risks associated with the timely publication of those standards?
Evolvability	Are the communications protocols chosen scalable to support foreseeable demands from System Users?
Deployability	Are the communications protocols required by the Core practical to deploy from a capital and human resource perspective?

4.4.3.3 Object Definitions and Roles

4.4.3.3.1 Fixed Device

This node is the Center User, Field User, other Core or External Support System node with a fixed point network connection. Typically this is a wired connection, though it could be a local wireless network (e.g. using 802.11b/g/n) that uses a wired backhaul. This connection to the Core could use a dedicated connection to the Core or could pass through the public Internet, as documented in Connectivity View – High Level.

Application Mobile Component: This is the Software Engineering Object that wants to communicate with the Core, most likely software.
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Internet Protocols: UDP/TCP and IPv6
Low Level WAN: A variety of wide area network communications media are possible, including 3G/4G cellular, WiMAX, T1/T3

4.4.3.3.2 Core Access Node

This is the Core System Node that receives all System User communications, scans it for intrusion protection, and routes it to the appropriate destination.

Conn: The SEO responsible for providing connectivity between the Core and other networks.
Intrusion Prevention: The SEO responsible for implementing the Intrusion Prevention and Detection functions.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gbps.

4.4.3.3.3 Service Router

The Service Router receives all System User communications that are intended for a Core service and routes data to the appropriate Core SEO.

Service Router: The SEO responsible for implementing Service Router functions.
Intrusion Prevention and Detection: The SEO responsible for implementing the Intrusion Prevention and Detection function.
Intrusion Detection: The SEO responsible for implementing the Intrusion Detection function.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gbps.

4.4.3.3.4 Service Component Node

This is the Core System node that provides service functionality. Any Functional Object other than those assigned to decryption or service routing operates on a Service Component Node.

Core Service Component: Representative of any Core System SEO that communicates with a Mobile User.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).

Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gbps.

4.4.3.3.5 Decryption Node

The Core System node that provides decryption of messages intended for the core, re-encrypts them using local encryption keys, and sends the message data back to the Service Router.

Core Local Encryptor: The SEO responsible for implementing the Core Local Encryption function.
Core Decryptor: The SEO responsible for implementing the Core Decryption function.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gbps.

4.4.3.4 View Description

Communications between a Fixed Device and the Core occur similarly to how that device accesses any other Internet-addressable device. It uses IP protocols to contact the Core, possibly using an encrypted session. Addressing the Core may start with a well-known hostname that is resolved by the Fixed Device's Domain Name Service server, or it may be stored locally on the Fixed Device node.

All messages reach the Core through the Core Access Node. Messages are scanned by Intrusion Protection and Detection SEOs as they pass up the protocol stack, and again as they descend. Messages that are not blocked by Intrusion Protection are passed to the Service Router.

A message that arrives at the Core System's Service Router will be passed to the relevant SEO on an available Service Component Node. If the message is encrypted it may need to be passed to the Core Decryption Node, decrypted, re-encrypted and then returned.

The Core System uses IPv6 as the middle-layer protocols for all internal and external network interfaces. TCP offers acknowledgement and retransmit that improves overall reliability. IP includes unique message identification that can assist with non-repudiation detection. Replay attacks still require authenticity verification and the Intrusion Prevention SEO. Developers may opt to use UDP for some Core interfaces, sacrificing TCP's reliability for simplicity and potentially increased throughput. IPv6 includes more useful multicast features than IPv4; some interfaces may be built to leverage this. Multicast should decrease the processing requirements for Core SEOs that can use it.

Security, including both the protection of data and assurance of data integrity, is handled at multiple layers. Session-layer protocols such as TLS and DTLS can be used to secure message exchanges. Internet Protocol Security (IPsec) can be used at the Internet layer. All of these protocols require significant processing and therefore more resources at source and destination. Individual messages can be encrypted at the application layer (the SEO); like session and Internet layer protocols this is expensive in terms of compute resources.

Message non-repudiation is handled by the use of a signature, a cryptographic hash based on the public-key infrastructure concepts. Assurance that a message is genuine, and in particular not a replay, relies on

this function as well, though it assumes that a reasonable timestamp is provided as part of the message content. If timestamps are not included in the message, then replay attacks are vulnerability.

Most of the protocols suggested for use in this view are available and usable without new research. IPv6 equipment and software is substantially more available than it was during the VII period; reports from the Proof-of-Concept about the difficult of acquiring IPv6 equipment should not be a concern.

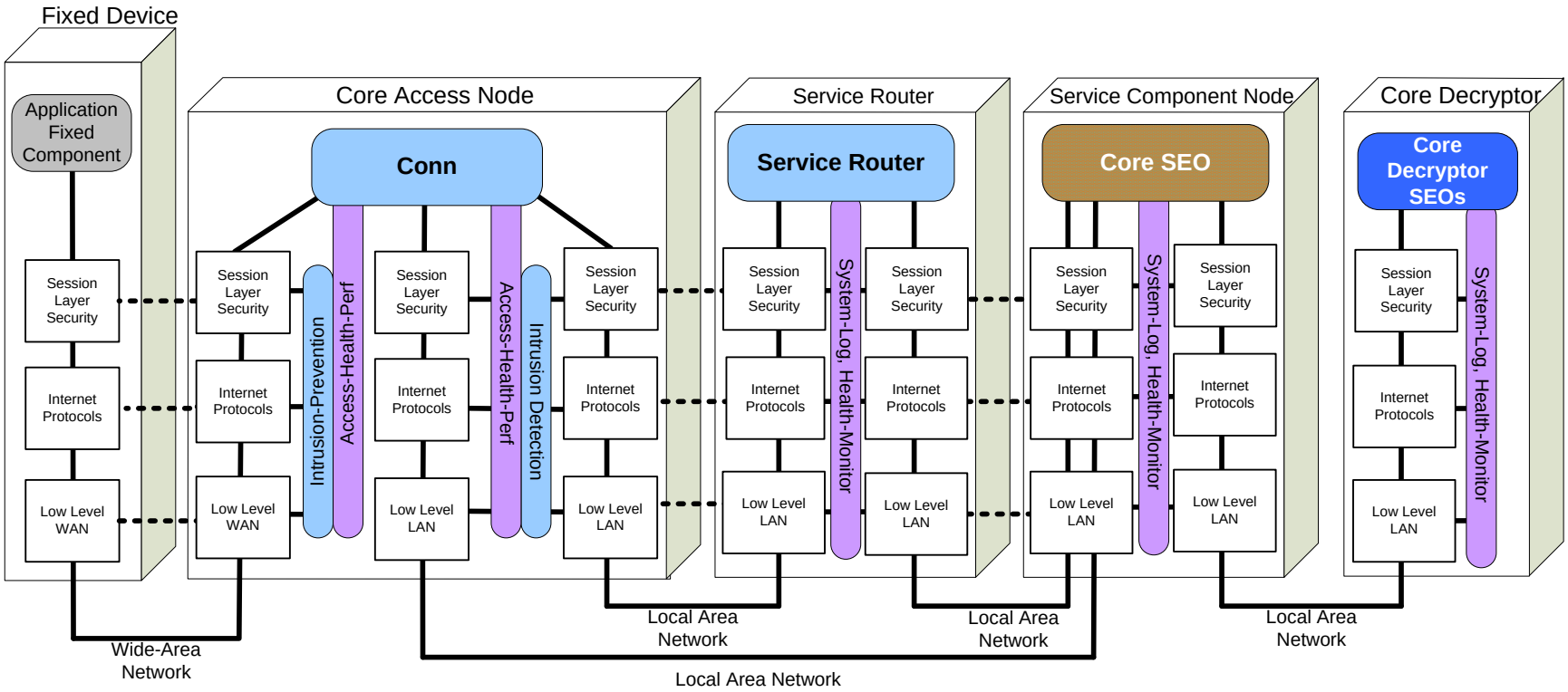


Figure 4-27: Communications View – Fixed Point-Core Communications

Note that the Core Switch is not shown in this diagram. This is to reduce the number of objects on the drawing and make it easier to read. The Core Switch functions as part of the Local Area Network, and includes Low-Level LAN protocol boxes, as well as the Switch-Health-Perf SEO.

4.4.3.5 Configuration Information

The following views must be considered when making changes to this view:

Enterprise Viewpoint: Enterprise View – Business Model
Functional Viewpoint: Functional View – Core Decryption
Connectivity Viewpoint: Connectivity View – High Level
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Communications Viewpoint: Communications View – Mobile DSRC Device and Core
Communications Viewpoint: Communications View – Mobile Wide-Area Wireless User and Core
Communications Viewpoint: Communications View – Core Routing

4.4.4 Communications View – Core Routing

4.4.4.1 Introduction

Field and Center Users and other Core Systems connected to the Core by private networks may interact with one another by passing communications traffic through the Core Access Node. Mobile Users that interact with Field Nodes connected to a Core through such a private network also can interact with other Centers and even other Cores. This view illustrates the Communications View of such interactions, summarizing the differences in the previous three views when the Core is used as a router.

4.4.4.2 Concerns Addressed by this View

Performance	Do the communications protocols allow the Core to meet the performance requirements defined in the SyRS?
Interfaces	Do communications protocols allow the Core to meet the interface requirements defined in the SyRS?
Functionality	What functionality exists in the communications protocols used by the Core? What reliability features are included in the communication protocols?
Security	What provisions for ensuring the privacy of communications by System Users are included in the communications protocols? How do the communications protocols provide non-repudiation of messages sent to and from the Core System? How do the communications protocols protect the integrity of messages sent to and from the Core System.
Organization/Resources	What resources are required to develop, deploy, operate and maintain the communications protocols that the Core System needs?
Appropriateness	Do the communications protocols fulfill the needs set forth in the ConOps and support applications that meet the overall *connected vehicle* goals?
Feasibility	Are the communications protocols required by the Core System feasible to specify, develop, and deploy?
Risks	If communications protocols chosen are developed by independent bodies are there any risks associated with changes to those standards that may impact the applicability of those standards? If communications protocols chosen are developed by independent bodies are there any risks associated with the timely publication of those standards?
Evolvability	Are the communications protocols chosen scalable to support foreseeable demands from System Users?

Deployability	Are the communications protocols required by the Core practical to deploy from a capital and human resource perspective?

4.4.4.3 Object Definitions and Roles

4.4.4.3.1 Mobile DSRC Device

This node is the Mobile User node equipped with a DSRC radio that needs to communicate with the Core System.

Application Mobile Component: This is the SEO that wants to communicate with the Core.
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Internet Protocols: TCP, UDP and IPv6
IEEE 1609.3: Defines WSM and the WSMP
IEEE 1609.2: Defines security services for use over DSRC.
IEEE 802.2: Defines the LLC layer
IEEE 1609.4: Defines the WAVE MAC layer
IEEE 802.11p: Defines the physical layer

The Mobile DSRC Device and DSRC Field Node communicate across a wireless medium.

4.4.4.3.2 DSRC Field Node

This field node provides DSRC connectivity to the Mobile User with backhaul capable of accessing the Core System. This could be through the Internet or through a private connection to the Core.

Field Mobile Component: This SEO is responsible for managing the communications exchange between the Mobile and the Core System. When the Mobile uses WSM, this object is responsible for the translation between WSM and Internet Protocols. Application architecture may dictate further responsibilities for this object. This object could be limited to protocol translation, but could also have higher layer application related functionality (e.g., data aggregation). Responsibility for development of this object is split between the Field Node Developer and End User Application Developer, depending on the Field Node architecture and policies and application architecture. (Field Node Developers may enable End User Application Developers to develop and install software on Field Nodes.)
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Internet Protocols: TCP, UDP and IPv6
IEEE 1609.3: Defines WSM and the WSMP
IEEE 1609.2: Defines security services for use over DSRC.
IEEE 802.2: Defines the LLC layer
IEEE 1609.4: Defines the WAVE MAC layer
IEEE 802.11p: Defines the physical layer
Low Level WAN: A variety of WAN communications protocols using a variety of physical media are possible, including cellular, WiMAX, T1/T3

4.4.4.3.3 Fixed Device

This node is the Center User, Field User, other Core or External Support System node with a fixed point network connection. Typically this is a wired connection, though it could be a local wireless network

(e.g. using 802.11b/g/n) that uses a wired backhaul. This connection to the Core could use a dedicated connection to the Core or could pass through the public Internet, as documented in Connectivity View – High Level.

Application Mobile Component: This is the Software Engineering Object that wants to communicate with the Core, most likely software.
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Internet Protocols: UDP/TCP and IPv6
Low Level WAN: A variety of wide area network communications media are possible, including 3G/4G cellular, WiMAX, T1/T3

4.4.4.3.4 Core Access Node

This is the Core System Node that receives all System User communications, scans it for intrusion protection, and routes it to the appropriate destination.

Conn: The SEO responsible for providing connectivity between the Core and other networks.
Intrusion Prevention: The SEO responsible for implementing the Intrusion Prevention and Detection functions.
Internet Protocols: UDP/TCP and IPv6
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Low Level LAN: A variety of local area network communications media are possible, but most likely switched Ethernet at 1 Gbps.

4.4.4.3.5 Mobile Wide-Area Wireless Device

This node is the Mobile User node equipped with a wireless radio that needs to communicate with the Core System.

Application Mobile Component: This is the SEO that wants to communicate with the Core.
Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Internet Protocols: TCP, UDP and IPv6
Data Link/Physical: Any of an assortment of protocols depending on the wide area wireless technology being used.

The Mobile Wide Area Wireless Device and Wide-Area Wireless Field Node communicate across a wireless medium.

4.4.4.3.6 Wide-Area Wireless Field Node

This is the field node providing wireless connectivity to the Mobile User. It has a backhaul providing connectivity to the Core System.

Session Layer Security: A session layer security protocol (e.g. TLS, DTLS).
Internet Protocols: TCP, UDP and IPv6
Data Link/Physical: Any of an assortment of protocols depending on the wide area wireless technology being used.
Low Level WAN: A variety of wide area network communications media are possible, including 3G/4G cellular, WiMAX, T1/T3

The Mobile Wide Area Wireless Device and Wide-Area Wireless Field Node communicate across a wireless medium. The Wireless Field Node and the Core System (Service Router) communicate across a wired or wireless medium.

4.4.4.4 View Description

Fixed Devices connected to the Core by a private network can use the Core Access Node to provide connectivity between them. When one or both parties in the communication are connected by private network to the Core, and the Core System Manager has established a relationship with the operator of the device (Field Node Owner/Operator or Center Owner/Operator), the Core Access Node may be configured to route traffic between the devices.

Since Mobile Users may interact with the Core through a Field Node, this impacts Mobile Users as well, effectively providing them connectivity to a Center User through a private network. The following figures illustrate the various scenarios. All concerns, issues and questions of protocols are identical to those discussed in the previous three views.

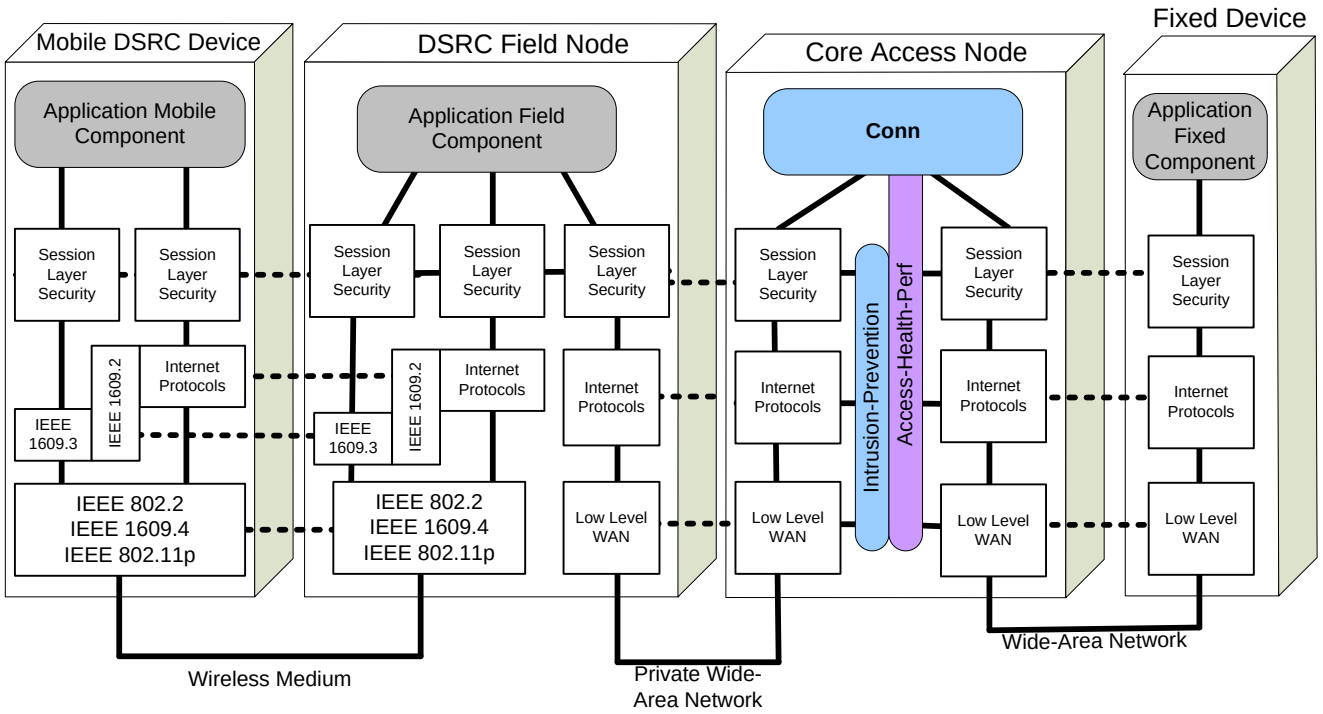


Figure 4-28: Communications View – DSRC Mobile over Private Network

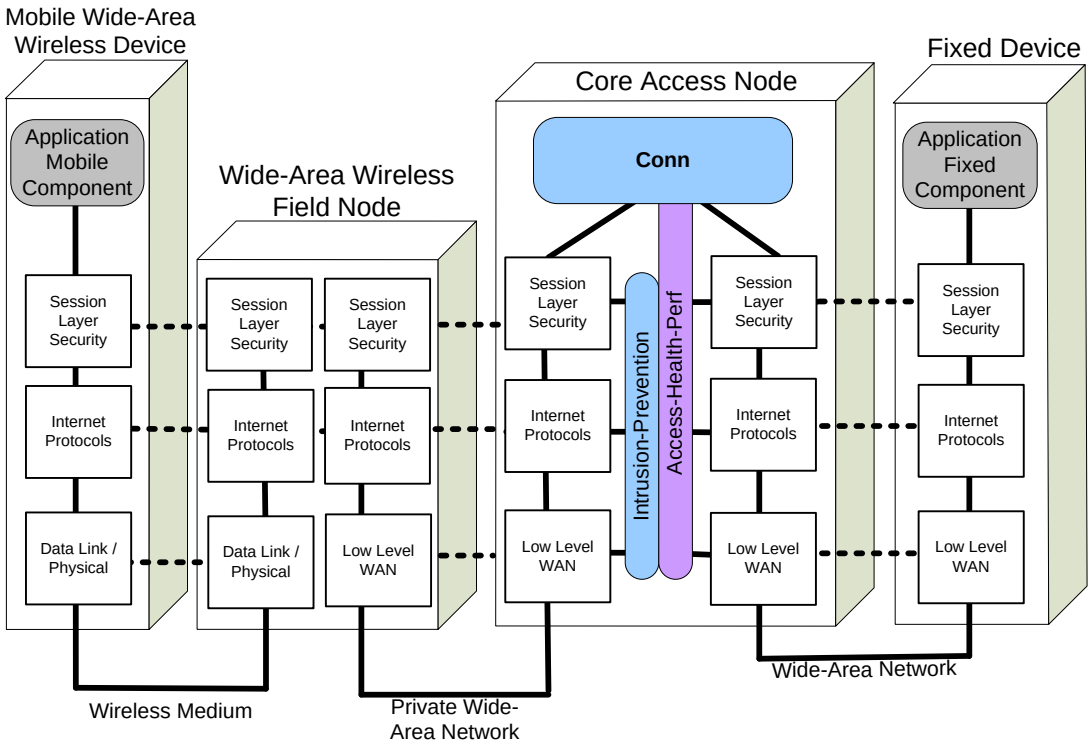


Figure 4-29: Communications View – Wide-Area Mobile over Private Network

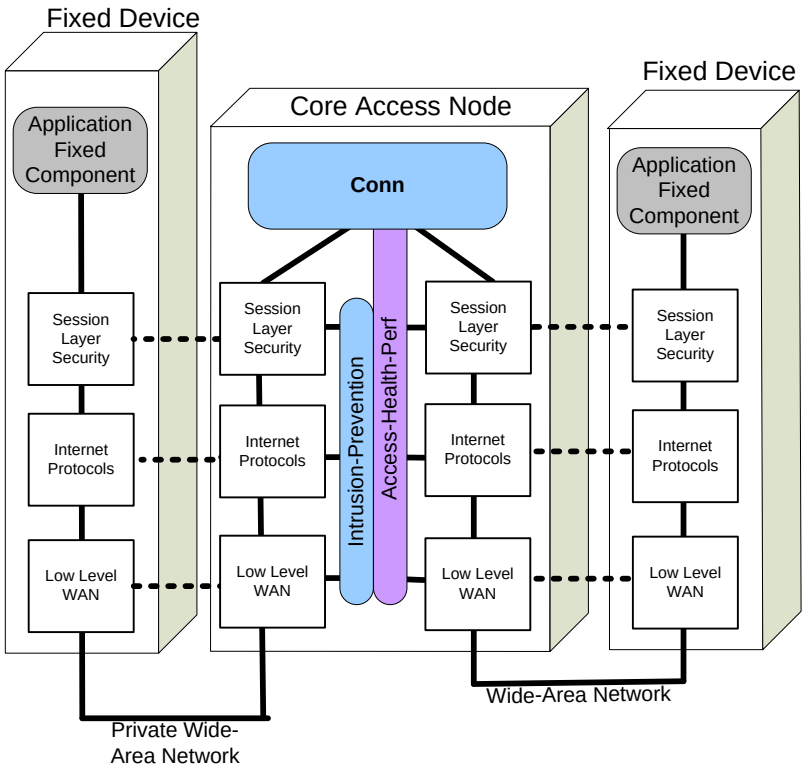


Figure 4-30: Communications View – Fixed Device over Private Network

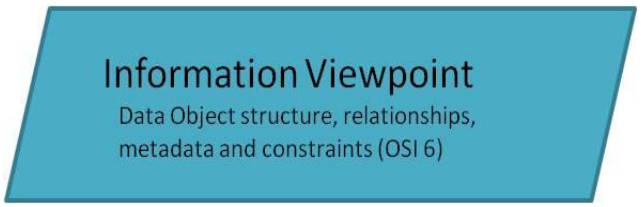
Note that the Core Switch is not shown on any of these diagrams. This is to reduce the number of objects on the drawings and make them easier to read. The Core Switch functions as part of the Local Area Network, and includes Low-Level LAN protocol boxes, as well as the Switch-Health-Perf SEO.

4.4.4.5 Configuration Information

The following views must be considered when making changes to this view:

Enterprise Viewpoint: Enterprise View – Business Model
Functional Viewpoint: Functional View – Core Decryption
Functional Viewpoint: Functional View – Networking
Connectivity Viewpoint: Connectivity View – High Level
Connectivity Viewpoint: Connectivity View – Core System Functional Allocation
Communications Viewpoint: Communications View – Mobile DSRC Device and Core
Communications Viewpoint: Communications View – Mobile Wide-Area Wireless User and Core
Communications Viewpoint: Communications View – Fixed Point Center/Field User and Core, Core2Core

4.5 Information Viewpoint



While much work has gone into the definition of message objects passed to and from vehicle-based DSRC-using Mobile Users (see the SAE J2735 standard), the Core System architecture will additionally define the messages that pass across the interfaces it provides to all System Users, the messages exchanged between Cores, the messages exchanged between the Core and ESS, and the messages various Core Software Engineering Objects exchange with one another internal to the Core.

Two Information Views are presented:

- External Objects
- Internal Objects

Table 4-10 shows which Information Views are relevant to each stakeholder.

Table 4-10: Information View Stakeholder Matrix

Information View \ Stakeholder	Mobile User	Field User	Center User	Operator	Acquirer	Maintainer	Developer	Manager	Tester	Policy Setter	Application Developer	Device Developer	Service Provider
Top Level External Objects	■	■	■	■	■		■		■	■	■	■	
Top Level Internal Objects				■		■	■		■				

The following Information Views are relevant to each Stakeholder as shown in the Table above:

- Mobile Users: Top Level External Objects
- Field Users: Top Level External Objects
- Center Users: Top Level External Objects
- Operators: both
- Acquirers: Top Level External Objects
- Maintainers: Top Level Internal Objects
- Developers: both
- Managers: neither
- Testers: both
- Policy Setters: Top Level External Objects
- Application Developers: Top Level External Objects
- Device Developers: Top Level External Objects
- Service Providers: neither

4.5.1 Information View – Top Level External Objects

4.5.1.1 Introduction

This view describes the objects that are exchanged, sent, and received over the Core System's external interfaces. This view does not define all messages, though it does define the content that must be included in some individual messages. Further decomposition will be needed to fully describe the messages and message dialogs necessary for an implementable interface. In some cases (e.g. Data Subscription Confirmation Info) an Information Object describes the acknowledgement portion of a communication. This occurs when there is a significant impact on Engineering Object definition because of that communication.

Many of these objects refer to relationships between Cores. These relationships are established by agreements such as a Memorandum of Understanding; they are not created automatically. Once those agreements are in place, then automated data exchanges through Core2Core may be configured. For a discussion of the types of agreements and relationships between Cores, see the Enterprise Viewpoint.

Also, the structure and/or use of many of these objects is subject to decisions yet to be made with regard to standardization or operational policies. These issues are captured in the 'policies' subsection under each object.

4.5.1.2 Concerns Addressed by this View

Security	Do any Information Objects potentially impact the privacy of System Users?
Interfaces	Are there any standards that define applicable message sets for Core System Information Objects? Which interfaces are candidates for standardization?
Appropriateness	Do the Information Objects convey information sufficient to realize Core System functionality as defined in the SyRS?

4.5.1.3 Object Definitions

4.5.1.3.1 Core2Core Objects

4.5.1.3.1.1 Backup Data

Directionality: Input and Output
Source/Destination: Other Core
Attributes: Digitally Signed and Secure, Acknowledgement required
Notes: Exchanged with Cores that have established relationships for backup and takeover.
Description: This is data extracted from a Core data store. Contents depend on the contents of that data store. Format is dependent on the receiving Core's formatting specification.

4.5.1.3.1.2 C2C Misbehavior Report

Directionality: Input and Output
Source/Destination: Other Core

Attributes: Digitally Signed and Secure, Acknowledgement required
Notes: Exchanged with Cores that have established relationships for exchange of misbehavior information. In practice, this will probably be limited to Cores that share geographic boundaries or have overlapping service areas.
Description: This includes the certificate ID associated with misbehavior, the type of misbehavior, time of the misbehavior, time of the misbehavior detection, certificate ID of the misbehavior report generator, and if available the identity of the misbehaving entity.

4.5.1.3.1.3 **Complete CRL**

Directionality: Input and Output
Source/Destination: Other Core
Attributes: Digitally Signed and Secure, Acknowledgement required
Notes: Exchanged with all other Cores that have a Trust Management subsystem.
Description: This is the list of all active digital certificates issued by the Core that are now invalid. Periodically regenerated, not instantaneously, so if a certificate has been revoked since its last generation, it will not be listed.

4.5.1.3.1.4 **Core Configuration Info**

Directionality: Input and Output
Source/Destination: Other Core
Attributes: Digitally Signed and Secure, Acknowledgement required
Notes: Exchanged with all other Cores that have a boundary or coverage area in common with or have a backup/takeover relationship with the Core System.
Description: This is a description of the services offered by the Core reporting the configuration info. Includes the area over which the service is offered, to whom the service is offered and a measure of the maximum expected performance of the Core's provision of those services (e.g., Data Distribution: max 500 subscribers and total data output < 10 Mb/s).

4.5.1.3.1.5 **Core Conflict Info**

Directionality: Input and Output
Source/Destination: Other Core
Attributes: Digitally Signed and Secure, Acknowledgement required
Notes: Exchanged with all other Cores that have a boundary or coverage area in common with the Core System.
Description: This is a description of the conflict in service provision between the two Cores. It includes the services that are in conflict and a description of the nature of the conflict, including area and System User types affected.

4.5.1.3.1.6 **Core Service Status Query**

Directionality: Input and Output
Source/Destination: Other Core
Attributes: Digitally Signed and Secure, Acknowledgement required
Notes: Exchanged with Core Systems that have a need to understand the status of services of one another. In practice this is likely limited to Cores that share geographic boundaries or have overlapping service areas.
Description: This requests the status of the targeted Core's services.

4.5.1.3.1.7 **Core Status Registration**

Directionality: Input and Output
Source/Destination: Other Core
Attributes: Digitally Signed and Secure, Acknowledgement required
Notes: Exchanged with Core Systems that have a need to understand the status of services of one another. In practice this is likely limited to Cores that share geographic boundaries or have overlapping service areas.
Description: This requests that the targeted Core provide the sending Core with status information on a periodic basis. The message includes the requestor's IP address, the services for which it requires status, the detail level of information to be provided and desired update frequency.

4.5.1.3.1.8 **CRL Deltas**

Directionality: Input and Output
Source/Destination: Other Core
Attributes: Digitally Signed and Secure, Acknowledgement required
Notes: Exchanged with all other Cores that have a Trust Management subsystem.
Description: This is the list of all active digital certificates issued by the Core that are invalid and are not included on the Complete CRL message. When combined with the Complete CRL includes all revoked certificate IDs.

4.5.1.3.1.9 **Data Backup Request**

Directionality: Input and Output
Source/Destination: Other Core
Attributes: Digitally Signed and Secure, Acknowledgement required
Notes: Exchanged with Cores that have established relationships for backup and takeover.
Description: This is the request sent to a Core that has a backup relationship with the sending Core. It includes the following characteristics of the data the Core wishes to back up: type of data store (e.g. Data Subscription Catalog), size of data to be backed up, desired backup start time, required backup completion time, length of time data must be backed up, projected restoration time (if any).

4.5.1.3.1.10 **Data Request**

Directionality: Input and Output
Source/Destination: Other Core
Attributes: Digitally Signed and Secure, Acknowledgement required
Notes: Exchanged with Cores that have established relationships for backup and takeover.
Description: This is the request sent to a Core that has previously backed up data. It includes the data store the Core wishes to restore, the time it needs the restore to start, the time it needs the restore to complete by

4.5.1.3.1.11 **Other Core Takeover Request**

Directionality: Input and Output
Source/Destination: Other Core
Attributes: Digitally Signed
Notes: Exchanged with Cores that have established relationships for service takeover. In practice, this will probably be limited to Cores that share geographic boundaries or have overlapping service areas.
Description: This includes the service, coverage area, start and expected end time and expected performance load that the Core wants the destination Core to provide service for.

4.5.1.3.1.12 **Restore Data**

Directionality: Input and Output
Source/Destination: Other Core
Attributes: Digitally Signed and Secure, Acknowledgement required
Notes: Exchanged with Cores that have established relationships for backup and takeover.
Description: This is formerly backed up data. The contents depend on the contents of that data store. Format is dependent on the sending Core's formatting specification.

4.5.1.3.1.13 **Service Status for Cores**

Directionality: Input and Output
Source/Destination: Other Core
Attributes: Digitally Signed and Secure, Acknowledgement required
Notes: Exchanged with Core Systems that have a need to understand the status of services of one another. In practice this is likely limited to Cores that share geographic boundaries or have overlapping service areas.
Description: This describes the status of all of the Core's services. For each service it includes the following data: state, mode, time of last mode transition, previous mode, projected time until next mode transition (if known), % load of maximum.

4.5.1.3.2 Data Distribution Objects:

4.5.1.3.2.1 **Data Acceptance Info**

Directionality: Output
Source/Destination: System User
Attributes: Digitally Signed
Notes: Specification of coverage areas need to be standardized, so that System Users and Cores all have a common frame of reference for when dealing with data acceptance information. For example, coverage areas could be organized by county, or by geometric area specified by polygon using lat/long coordinates as vertices. Similarly, Cores must operate with the same concepts of System User types (e.g., Transit Vehicles) so that they may properly coordinate activities that are limited to certain types of System Users.
Description: This is the response to the Data Provision Request. Indicates the service area over which the data in the Data Provision Request is accepted. This message may also be provided if a System User provides data that the Core does not accept.

4.5.1.3.2.2 **Data Provision Request**

Directionality: Input
Source/Destination: System User
Attributes: Digitally Signed, Acknowledgement Required
Notes: Specification of coverage areas need to be standardized, so that System Users and Cores all have a common frame of reference for when dealing with data acceptance information. For example, coverage areas could be organized by county, or a fraction of lat/long coordinates. Similarly, Cores must operate with the same concepts of System User types (e.g., Transit Vehicles) so that they may properly coordinate activities that are limited to certain types of System Users.
Description: This includes the type of data to be provided, source type and area over which the System User will provide data. For Mobile Users this is limited to SAE J2735 message types. For Field and

Center Users this may include messages in SAE J2735, but may include other messages. The specification of additional Field and Center data that can use this interface is TBD.

4.5.1.3.2.3 Data Subscription Confirmation Info

Directionality: Output
Source/Destination: System User
Attributes: Digitally Signed
Notes: Specification of coverage areas need to be standardized, so that System Users and Cores all have a common frame of reference for when dealing with data acceptance information. For example, coverage areas could be organized by county, or a fraction of lat/long coordinates. Similarly, Cores must operate with the same concepts of System User types (e.g., Transit Vehicles) so that they may properly coordinate activities that are limited to certain types of System Users.
Description: This is a response to the Data Subscription Request, describing the exact parameters of the System User's subscription.

4.5.1.3.2.4 Data Subscription Redirect Info

Directionality: Output
Source/Destination: System User
Attributes: Digitally Signed
Notes: Specification of coverage areas need to be standardized, so that System Users and Cores all have a common frame of reference for when dealing with data acceptance information. For example, coverage areas could be organized by county, or a fraction of lat/long coordinates. Similarly, Cores must operate with the same concepts of System User types (e.g., Transit Vehicles) so that they may properly coordinate activities that are limited to certain types of System Users.
Description: This is a response to the Data Subscription Request, rejecting the subscription. If the Core knows of another Core that may be able to satisfy the subscription, this message includes the IP address of that Core.

4.5.1.3.2.5 Data Subscription Request

Directionality: Input
Source/Destination: System User
Attributes: Digitally Signed
Notes: Specification of coverage areas need to be standardized, so that System Users and Cores all have a common frame of reference for when dealing with data acceptance information. For example, coverage areas could be organized by county, or a fraction of lat/long coordinates. Similarly, Cores must operate with the same concepts of System User types (e.g., Transit Vehicles) so that they may properly coordinate activities that are limited to certain types of System Users.
Description: This is the request to subscribe to data; includes the System User's Core account name and authorization and the following:

- Data the System User wishes to subscribe to (SAE J2735 messages if Mobile)
- Maximum data acceptance rate
- Aggregation time for the data (e.g., 1 hour would result in only 1 message per hour, with the message content the aggregation of the data elements over that one hour period)
- Sampling rate for the data (e.g., 1 in every 10 samples)
- Desired start time for subscription
- Desired end time for subscription

4.5.1.3.2.6 **Direct Data Distribution Info**

Directionality: Output
Source/Destination: System User
Attributes: Digitally Signed
Notes: Specification of coverage areas need to be standardized, so that System Users and Cores all have a common frame of reference for when dealing with data acceptance information. For example, coverage areas could be organized by county, or a fraction of lat/long coordinates. Similarly, Cores must operate with the same concepts of System User types (e.g., Transit Vehicles) so that they may properly coordinate activities that are limited to certain types of System Users.
Description: This is data describing 3rd parties that accept data that the Core does not accept. Includes the IP address and format expectations of the 3rd party.

4.5.1.3.2.7 **Field Node Configuration Information**

Directionality: Input
Source/Destination: System User
Attributes: Digitally Signed, Acknowledgement required
Notes: Specification of coverage areas need to be standardized, so that System Users and Cores all have a common frame of reference for when dealing with data acceptance information. For example, coverage areas could be organized by county, or a fraction of lat/long coordinates. Similarly, Cores must operate with the same concepts of System User types (e.g., Transit Vehicles) so that they may properly coordinate activities that are limited to certain types of System Users. Use of geo-cast messaging may be limited; criteria for establishing those limits must be defined.
Description: This specifies the location, IP address, communications range, bandwidth and constraints for use of Field Node Infrastructure. Constraints on the use of the Field Node may be imposed by the Field Node Owner/Operator, and also need to be specified here.

4.5.1.3.2.8 **Geo-Cast Message**

Directionality: Input
Source/Destination: System User
Attributes: Digitally Signed, Acknowledgement required
Notes: Specification of coverage areas need to be standardized, so that System Users and Cores all have a common frame of reference for when dealing with data acceptance information. For example, coverage areas could be organized by county, or a fraction of lat/long coordinates. Similarly, Cores must operate with the same concepts of System User types (e.g., Transit Vehicles) so that they may properly coordinate activities that are limited to certain types of System Users. Use of geo-cast messaging may be limited; criteria for establishing those limits must be defined.
Description: This is a message that the System User wishes to distribute over a specific area, either once or repeatedly over a period of time. The message includes the content to be distributed and information describing the desired distribution area and time over which the distribution should be made.

4.5.1.3.2.9 **Other Core Acceptance Info**

Directionality: Output
Source/Destination: System User
Attributes: Digitally Signed
Notes: Specification of coverage areas need to be standardized, so that System Users and Cores all have a common frame of reference for when dealing with data acceptance information. For example, coverage areas could be organized by county, or a fraction of lat/long coordinates. Similarly, Cores must op-

erate with the same concepts of System User types (e.g., Transit Vehicles) so that they may properly coordinate activities that are limited to certain types of System Users.
Description: This is the response to the Data Provision Request, sent when the Core does not accept the data the System User wishes to provide. It indicates the IP Address of another Core that services the area referenced in the Data Provision Request.

4.5.1.3.2.10 **Provided Data**

Directionality: Input
Source/Destination: System User
Attributes: Digitally Signed
Notes: Data from Mobile Users restricted to what is permitted by SAE J2735.
Description: This is data received from System User Data Providers intended for the publish/subscribe engine. Format of the data received from System Users is described by SAE J2735. Message formats for Field and Center Users are TBD.

4.5.1.3.2.11 **Repackaged, Addressed Data**

Directionality: Output
Source/Destination: System Users
Attributes: Digitally Signed
Notes: If the Core does not perform sampling, aggregation or selection of specific data fields, then there is no transformation or aggregation relationship with the incoming Data.
Description: This is data repackaged to match a specific subscriber's subscription criteria. This is data that was originally provided by other System Users (Data, above) that has been selected based on subscription criteria. If the subscription includes aggregation or sampling then the data will be aggregated or sampled as appropriate. If the subscription requested selection of specific data fields of incoming Data messages, then only those fields would be included.

4.5.1.3.3 Misbehavior Management Objects

4.5.1.3.3.1 **System User Misbehavior Report**

Directionality: Input
Source/Destination: System Users
Attributes: Digitally Signed
Notes: System Users will have to be characterized for reliability and believability in order to use misbehavior reports they provide.
Description: This is data describing misbehaving users. It Includes the certificate ID associated with misbehavior, the type of misbehavior, time of the misbehavior, time of the misbehavior detection, certificate ID of the misbehavior report generator, and if available the identity of the misbehaving entity.

4.5.1.3.4 Network Services Objects

4.5.1.3.4.1 **Data In**

Directionality: Input
Source/Destination: System Users, ESS, Other Cores
Attributes: Any combination of Digitally Signed, Secure and Acknowledgement Required
Notes: None
Description: This is any message received by the Core.

4.5.1.3.4.2 Data Out

Directionality: Output
Source/Destination: System Users, ESS, Other Cores
Attributes: Any combination of Digitally Signed, Secure and Acknowledgement Required
Notes: None
Description: This is any message sent by the Core.

4.5.1.3.5 Service Monitor Objects

4.5.1.3.5.1 Performance Records

Directionality: Output
Source/Destination: Core Certification Authority
Attributes: Digitally Signed, Secure, Acknowledgement Required
Notes: Makeup and operations of the Core Certification Authority may impact the information desired in this message.
Description: This is detailed information describing the long-term performance of Core services, provided to the Core Certification Authority. It includes of a record of availability for each system service since the last reporting period, performance loading records for services, and conflicts reported between this Core and other Cores.

4.5.1.3.5.2 Service Status

Directionality: Output
Source/Destination: System Users
Attributes: Digitally signed, optionally Acknowledgement Required
Notes: None.
Description: This is the status of the Core's subsystems. This may be limited to specific subsystems if in response to a query. For each service, this message indicates the current state and mode and if there is a known time for that state or mode to change.

4.5.1.3.5.3 Service Status Query

Directionality: Input
Source/Destination: System Users
Attributes: Digitally Signed, Acknowledgement Required
Notes: None.
Description: This is the request from System Users for Core service status information. Includes a listing of the services the System User desires status information about.

4.5.1.3.5.4 System User Status Registration

Directionality: Input
Source/Destination: System User
Attributes: Digitally Signed, Acknowledgement Required
Notes: None.
Description: This is a request from a System User to register for periodic reports of the Core's service status. It includes the System User's IP Address and desired update frequency

4.5.1.3.6 Time Synchronization Objects

4.5.1.3.6.1 Time

Directionality: Input
Source/Destination: External Stratum-2 time source
Attributes: None
Notes: None.
Description: This is time provided by an external Stratum-2 time source.

4.5.1.3.7 User Permissions Objects

4.5.1.3.7.1 Application Permission Request

Directionality: Input
Source/Destination: End User Application Developer
Attributes: Digitally Signed and Acknowledgement Required
Notes: Dependent on the Core's support for field applications.
Description: This is a submission of application permissions that must be managed using IEEE 1609.2 certificates. It includes the identification of the application, developer, version, and certificate permission information according to the formats in IEEE 1609.2.

4.5.1.3.7.2 User Identity and Permission Request

Directionality: Input and Output
Source/Destination: Other Cores
Attributes: Digitally Signed and Acknowledgement Required
Notes: Dependent on the Core's support for field applications. Since the identity credentials include PII, this message could impact the privacy of System Users.
Description: This is a submission of identity credentials and a request for permissions to use the Core System services. Optionally includes a request for application services managed by certificates distributed by the Core.

4.5.1.3.8 User Trust Management Objects

4.5.1.3.8.1 Credential Request

Directionality: Input
Source/Destination: System User
Attributes: Digitally Signed and Acknowledgement Required
Notes: Signed using the System User's currently valid digital certificate. If they have no such certificate, they must have a long term "base, low permissions" certificate that enables this request.
Description: This is request for credentials. It includes the System User's identity and the period of time the user needs the new certificate to be valid for.

4.5.1.3.8.2 Credential Referral

Directionality: Output
Source/Destination: System User
Attributes: Digitally Signed and Acknowledgement Required
Notes: None.

Description: This message is provided in response to a credential request that the Core cannot satisfy. It contains the IP address of another Core or ESS that provides the type of credentials the System User is requesting.

4.5.1.3.8.3 Credentials

Directionality: Output
Source/Destination: System User
Attributes: Digitally Signed and Acknowledgement Required
Notes: Since the identity credentials include PII, this message could impact the privacy of System Users.
Description: This message contains a new digital certificate for the System User. The certificate will be in the form of an X.509 digital certificate, and describe the actions the System User is allowed to perform when interacting with the Core and other System Users.

4.5.1.3.8.4 CRL

Directionality: Output
Source/Destination: System User
Attributes: Digitally Signed and Secure
Notes: None.
Description: This is the list of all active digital certificates issued by the Core that are now invalid. Periodically regenerated, not instantaneously, so if a certificate has been revoked since its last generation, it will not be listed.

4.5.1.3.8.5 CRL Request

Directionality: Input
Attributes: Digitally Signed, Secure and Acknowledgement Required
Source/Destination: System User
Notes: None.
Description: This request from a System User for that user to receive CRLs from the Core. It includes the type of CRL the System User wishes to acquire.

4.5.1.3.8.6 Ext X.509 Cert Request

Directionality: Output
Source/Destination: External Support System X.509 CA
Attributes: Digitally Signed and Secure
Notes: None.
Description: This is a request from the Core System to an external X.509 Certificate Authority, formatted according to that CA's requirements, requesting an X.509 certificate for the Core.

4.5.1.3.8.7 External CRL

Directionality: Input
Source/Destination: External Support System X.509 CA or External Support System DSRC CA.
Attributes: Digitally Signed
Notes: None.
Description: This a Certificate Revocation List received from an external CA.

4.5.1.3.8.8 Misbehaving User ID

Directionality: Output
Source/Destination: External Support System X.509 CA or External Support System DSRC CA.
Attributes: Digitally Signed, Secure and Acknowledgement Required

Notes: Standardized types of misbehavior will have to be defined. Interfaces to the ESS CAs will have to be defined. Feasibility of obtaining revocation from ESS X.509 CA is TBD.
Description: This contains the ID and type of misbehavior the System User has committed. Depending on the method of identification, the contents could include a user ID, a certificate ID or a pseudo-ID.

4.5.1.3.8.9 User Identification

Directionality: Input
Source/Destination: ESS DSRC Registration Authority
Attributes: Digitally Signed and Acknowledgement Required
Notes: Establishment of standard System User classes must be done in order to use the acquire permissions for classes of users without requiring identity. Since the message includes PII, this message could impact the privacy of System Users.
Description: This is a request for special permissions for a System User from the ESS DSRC RA. It includes an identification of the System User or the class that the System User is part of that the RA wishes to acquire permissions for.

4.5.1.3.8.10 User Special Permissions

Directionality: Output
Source/Destination: ESS DSRC Registration Authority
Attributes: Digitally Signed and Acknowledgement Required
Notes: Establishment of standard System User classes must be done in order to use the acquire permissions for classes of users without requiring identity. Since the identity referenced by this message could be PII, this message could impact the privacy of System Users.
Description: This is the response to the ESS DSRC RA. It includes an identification of the special permissions the user is entitled to.

4.5.1.3.8.11 X.509 Certificate

Directionality: Input
Source/Destination: External Support System X.509 CA
Attributes: Digitally Signed, Secure and Acknowledgement Required
Notes: None.
Description: This is an X.509 Certificate formatted according to ITU-T X.509, including the Core System's identity.

4.5.1.3.8.12 X.509 CRL

Directionality: Input
Source/Destination: External Support System X.509 CA
Attributes: Digitally Signed
Notes: None.
Description: This is a list of all activated X.509 certificates that are invalid.

4.5.1.4 View Description

External Information Objects are organized according to the subsystem that they originate from or terminate in, as shown below. Some of these objects relate to existing standards: X.509-related messages trace to the X.509 standard, and Provided Data includes (but is not limited to) the data descriptions in SAE J2735. The remaining messages will have to be defined. If Core Systems are to be widely deployed, these external interfaces are candidates for standardization.

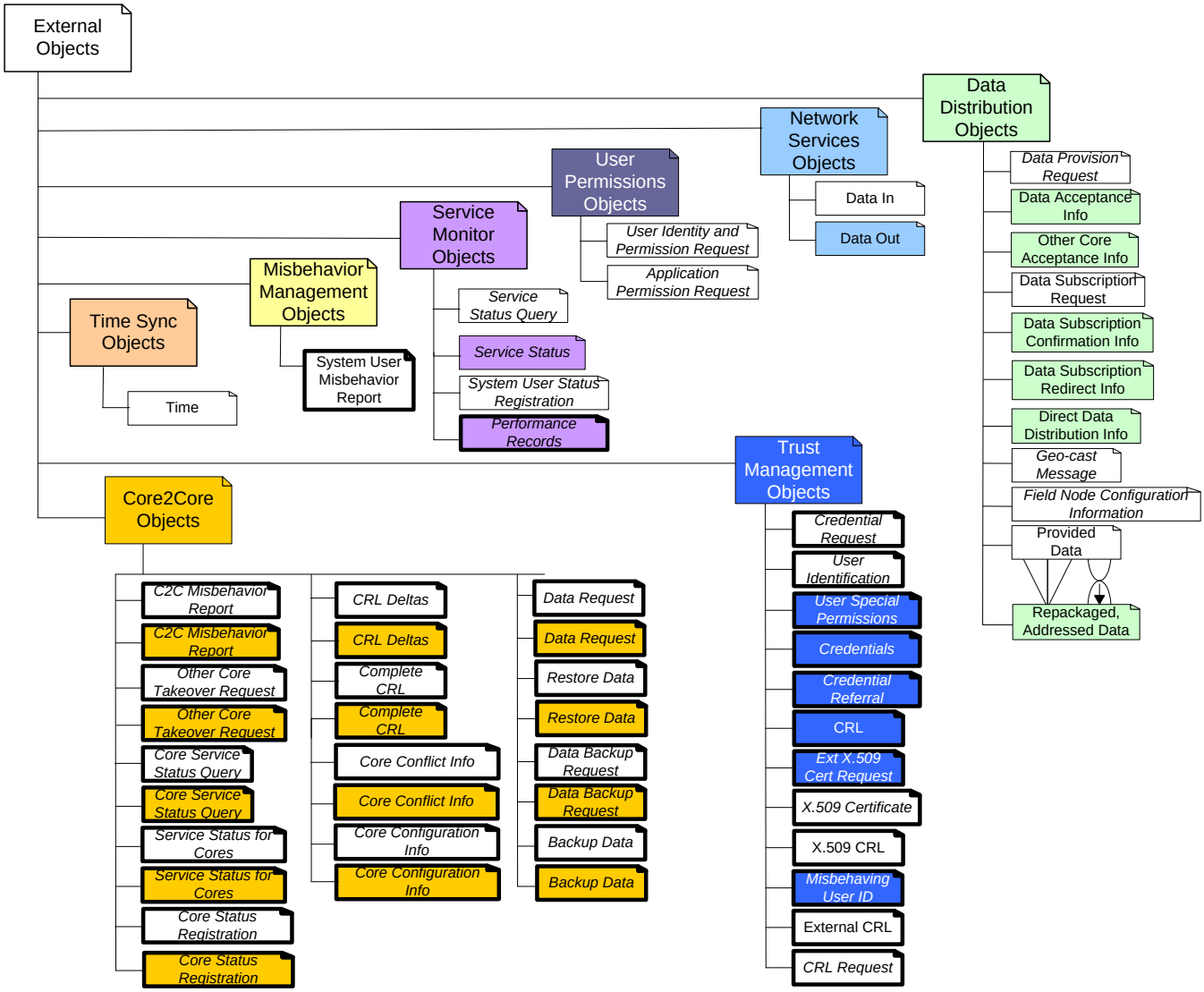


Figure 4-31: Information View – Top Level External Objects

4.5.1.5 Configuration Information

The following views must be considered when making changes to this view:

Functional Viewpoint: Functional View – Top Level
Functional Viewpoint: Functional View – Data Distribution
Functional Viewpoint: Functional View – System Configuration
Functional Viewpoint: Functional View – User Configuration
Functional Viewpoint: Functional View – System Monitor and Control
Functional Viewpoint: Functional View – Credentials Distribution
Functional Viewpoint: Functional View – Misbehavior Management
Functional Viewpoint: Functional View – Core Decryption
Functional Viewpoint: Functional View – Networking
Functional Viewpoint: Functional View – Core Backup
Information Viewpoint: Information View – Top Level Internal Objects

4.5.2 Information View – Top Level Internal Objects

4.5.2.1 Introduction

This view describes the objects that are exchanged, accepted and sent between subsystems over the Core System's internal interfaces.

4.5.2.2 Concerns Addressed by this View

Security	Do any Information Objects potentially impact the privacy of System Users?
Interfaces	Are there any standards that define applicable message sets for Core System Information Objects? Which interfaces are candidates for standardization?
Appropriateness	Do the Information Objects convey information sufficient to realize Core System functionality as defined in the SyRS?

4.5.2.3 Object Definitions

4.5.2.3.1 Objects Received by Core2Core

4.5.2.3.1.1 Core Config Info

Source: All subsystems
Destination: Core2Core
Attributes: Digitally Signed, Secure
Description: This message includes configuration data describing the services offered by the Core and the users it offers those services to.
Response Message Expected: None

4.5.2.3.1.2 Config Info for Other Cores

Source: All subsystems
Destination: Core2Core
Attributes: Digitally Signed, Secure
Description: This message includes configuration data describing the services offered by the Core and the users it offers those services to.
Response Message Expected: None

4.5.2.3.1.3 Data to be Backed Up

Source: All subsystems
Destination: Core2Core
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes data from the Core data stores that will be backed up by another Core.
Response Message Expected: None

4.5.2.3.1.4 Core Distribution List

Source: Service Monitor

Destination: Core2Core
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a list of Cores that have registered to receive automatic service status updates. This includes their IP address, the services they are to receive information about, the detail level of information they are to receive and their desired update rate.
Response Message Expected: None

4.5.2.3.1.5 Complete CRL

Source: User Trust Management
Destination: Core2Core
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a list of IDs of all certificates the Core has revoked.
Response Message Expected: None

4.5.2.3.1.6 CRL Deltas

Source: User Trust Management
Destination: Core2Core
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a list of changes to the CRL since the last CRL Delta or Complete CRL was sent to the Core that is receiving this message.
Response Message Expected: None

4.5.2.3.1.7 Detailed Service Status

Source: Service Monitor
Destination: Core2Core
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes detailed information describing the performance of all Core functions and interfaces.
Response Message Expected: None

4.5.2.3.1.8 Misbehavior Report

Source: Misbehavior Management
Destination: Core2Core
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a misbehavior report from the Misbehavior Reports Log. It sent with the expectation that it will be forwarded to other Core Systems that might use the report.
Response Message Expected: None

4.5.2.3.2 Objects Received by Data Distribution

4.5.2.3.2.1 Geo-cast Info Changes

Source: Misbehavior Management
Destination: Data Distribution
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message describes changes to the geo-cast info of geo-cast supporting devices; this includes performance and operating characteristics, and the permissions System Users must have to access these devices.
Response Message Expected: None

4.5.2.3.3 Objects Received by Misbehavior Management

4.5.2.3.3.1 Authenticity Error Message

Source: User Trust Management
Destination: Misbehavior Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a message and an indication of why it failed authenticity checks. Possible reasons include at minimum format, signature and duplicate message.
Response Message Expected: None

4.5.2.3.3.2 Decryption Error Message

Source: User Trust Management
Destination: Misbehavior Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes an encrypted message and an indication of failed decryption.
Response Message Expected: None

4.5.2.3.3.3 Internal Misbehavior Report

Source: User Permissions
Destination: Misbehavior Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message reports misbehavior. It includes an identification of the reporting function, ID or pseudo-ID of the misbehaving user, and characterization of the type of misbehavior.
Response Message Expected: None

4.5.2.3.3.4 Intrusion Alert

Source: Network Services
Destination: Misbehavior Management
Attributes: Digitally Signed, Secure
Description: This message reports a potentially malicious intrusion detected by Network Services.
Response Message Expected: None

4.5.2.3.3.5 Operator Permissions

Source: User Permissions
Destination: Misbehavior Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes an Operator ID and a description of the Operator's permissions.
Response Message Expected: None

4.5.2.3.3.6 Other Core Misbehavior Report

Source: Core2Core
Destination: Misbehavior Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a misbehavior report received from another Core System.
Response Message Expected: None

4.5.2.3.3.7 Scope

Source: User Trust Management
Destination: Misbehavior Management

Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a description of the operational scope of User Trust Management. At minimum this includes a description of the geographic area, time and types of System Users the Core provides User Trust Management services to.
Response Message Expected: None

4.5.2.3.3.8 SUID, Function Permission

Source: User Permissions
Destination: Misbehavior Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This includes a System User ID, and an indication of whether that System User is permitted to use a specific function. Since the SUID could be PII, this message could impact the privacy of the System User.
Response Message Expected: None

4.5.2.3.3.9 Suspicious Application Request

Source: User Permissions
Destination: Misbehavior Management
Attributes: Digitally Signed and Secure
Description: This message includes data originally provided by an End User Application Deployer that did not pass a permission check. That could be because it was improperly formatted, or it requested permissions that the providing End User Application Deployer is not authorized to request. Since the information contained within this request could include PII, this message could impact the privacy of the End User Application Deployer.
Response Message Expected: None

4.5.2.3.3.10 Suspicious Data

Source: Data Distribution
Destination: Misbehavior Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes data originally provided by a System User that did not pass a permission check. That could be because it was improperly formatted, or it contained data that the providing System User is not authorized to provide to the Core.
Response Message Expected: None

4.5.2.3.3.11 Suspicious Data Subscription Request

Source: Core2Core
Destination: Misbehavior Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a copy of a subscription request that was deemed suspicious and rejected by Data Distribution. It was either improperly formatted or included a request for data that the originating System User is not permitted to receive.
Response Message Expected: None

4.5.2.3.3.12 Suspicious Geo-Cast

Source: Core2Core
Destination: Misbehavior Management
Attributes: Digitally Signed, Secure and Acknowledgement Required

Description: This message includes a Geo-cast Message that was deemed suspicious and thus rejected by Data Distribution. It was either improperly formatted, included a request to publish data that the originating System User is not permitted to publish or included a request to publish data in a geo-cast region (temporal or geographic) that the System User is not permitted.
Response Message Expected: None

4.5.2.3.3.13 Suspicious Permission Request

Source: User Permissions
Destination: Misbehavior Management
Attributes: Digitally Signed and Secure
Description: This message includes data originally provided by a System User that did not pass a permission check. That could be because it was improperly formatted, or it requested permissions that the providing System User is not authorized to request. Since the information contained within this request could include PII, this message could impact the privacy of the System User.
Response Message Expected: None

4.5.2.3.3.14 System User Permissions

Source: User Permissions
Destination: Misbehavior Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a System User ID and a description of the System User's permissions. Since the SUID could be PII, this message could impact the privacy of System Users.
Response Message Expected: None

4.5.2.3.4 Objects Received by Service Monitor

4.5.2.3.4.1 Core Status Registration

Source: Service Monitor
Destination: User Permissions
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes the ID and contact information of another Core, along with its desired update frequency and level of detail for Core performance reporting.
Response Message Expected: None

4.5.2.3.4.2 Service Control Node Performance

Source: Service Monitor
Destination: Network Services
Attributes: Digitally Signed
Description: This message includes a description of performance and loading of SEOs operating on Service Control Nodes.
Response Message Expected: None

4.5.2.3.5 Objects Received by User Permissions

4.5.2.3.5.1 Core ID, function

Source: Core2Core
Destination: User Permissions
Attributes: Digitally Signed, Secure and Acknowledgement Required

Description: This message includes the identity of another Core System and a reference to a function offered by this Core. The message is sent to determine if the other Core is permitted to carry out the function.
Response Message Expected: Permissions

4.5.2.3.5.2 Operator ID, function

Source: All subsystems
Destination: User Permissions
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes the identity of the Operator and a reference to a function offered by the Core. The message is sent to determine if the Operator is permitted to carry out the function.
Response Message Expected: Permissions

4.5.2.3.5.3 Operator ID, Revoke function

Source: Misbehavior Management
Destination: User Permissions
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes the identity of an Operator and a reference to a function offered by the Core System. The message is sent to have the Operator's permissions modified to revoke its use of that function.
Response Message Expected: Operator Permissions

4.5.2.3.5.4 Provider ID, function

Source: Data Distribution
Destination: User Permissions
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes the identity of a System User functioning as a data provider and a reference to a function offered by the Core System. The message is sent to determine if the System User is permitted to carry out the function. Since the Provider ID could be PII, this message could impact the privacy of System Users.
Response Message Expected: Permissions

4.5.2.3.5.5 Subscriber ID, function

Source: Data Distribution
Destination: User Permissions
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes the identity of a System User functioning as a data subscriber and a reference to a function offered by the Core System. The message is sent to determine if the System User is permitted to carry out the function. Since the Subscriber ID could be PII, this message could impact the privacy of System Users.
Response Message Expected: Permissions

4.5.2.3.5.6 System User ID, function

Source: Misbehavior Management, Service Monitor
Destination: User Permissions
Attributes: Digitally Signed, Secure and Acknowledgement Required

Description: This message includes the identity of a System User and a reference to a function offered by the Core System. The message is sent to determine if the System User is permitted to carry out the function. Since the SUID could be PII, this message could impact the privacy of System Users.
Response Message Expected: Permissions

4.5.2.3.5.7 System User ID, Revoke function

Source: Misbehavior Management
Destination: User Permissions
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes the identity of a System User and a reference to a function offered by the Core System. The message is sent to have the System User's permissions modified to revoke its use of that function. Since the SUID could be PII, this message could impact the privacy of System Users.
Response Message Expected: System User Permissions

4.5.2.3.5.8 Cert Owner, App

Source: User Trust Management
Destination: User Permissions
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes the identity of an application. The message is sent to determine the permissions associated with the application.
Response Message Expected: App Permissions

4.5.2.3.5.9 Certificate Owner ID

Source: User Trust Management
Destination: User Permissions
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a certificate ID. The message is sent to determine if the System User associated with this certificate ID is permitted to request a given set of credentials, and if so what permissions should be attached to those credentials. Since the Certificate Owner ID is associated with a System User it could be considered PII, so this message could impact the privacy of System Users.
Response Message Expected: Cert Permission

4.5.2.3.5.10 Permission Change Request

Source: User Trust Management
Destination: User Permissions
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a System User ID and describes changes to the permissions for that user. The message is sent to change the permissions associated with the System User.
Response Message Expected: None

4.5.2.3.5.11 User ID, Apps

Source: User Trust Management
Destination: User Permissions
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes the User ID of a user or class of users. The message is sent to determine the special permissions the user is entitled to.
Response Message Expected: Special App Permissions

4.5.2.3.6 Objects Received by User Trust Management

4.5.2.3.6.1 App Permissions

Source: User Permissions
Destination: User Trust Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message is a response to the Cert Owner, App message. It includes the application permissions for the application specified in the Cert Owner, App.
Response Message Expected: None

4.5.2.3.6.2 Cert Permission

Source: User Permissions
Destination: User Trust Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message is a response to the Certificate Owner ID message. It includes the System User ID from that message and an indication of whether the System User is allowed to request a given set of credentials and if so the types of permissions that must be attached to the digital certificate.
Response Message Expected: None

4.5.2.3.6.3 Complete CRL

Source: Core2Core
Destination: User Trust Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a list of IDs of all certificates that another Core has revoked.
Response Message Expected: None

4.5.2.3.6.4 CRL Deltas

Source: Core2Core
Destination: User Trust Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a list of changes to the CRL since the last CRL Delta or Complete CRL was sent by the Core that sent this message.
Response Message Expected: None

4.5.2.3.6.5 Misbehaving User ID

Source: Misbehavior Management
Destination: User Trust Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes the ID and type of misbehavior the System User has committed. Depending on the method of identification, the contents could include a user ID, a certificate ID or a pseudo-ID.
Response Message Expected: None

4.5.2.3.6.6 Remotely Encrypted Message

Source: All Subsystems (originated from an external entity)
Destination: User Trust Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes a message encrypted using the Core's public-key, originally received from an external source (ESS, System User, other Core).

Response Message Expected: Locally Encrypted Message

4.5.2.3.6.7 Special App Permissions

Source: User Permissions
Destination: User Trust Management
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message is a response to the User ID, Apps message. It includes the System User ID from that message and the application permissions the user is entitled to.
Response Message Expected: None

4.5.2.3.7 Objects Received by All Subsystems

4.5.2.3.7.1 Restore Data

Source: Core2Core
Destination: All Subsystems
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message includes data from another Core to be used to restore a Core data store.
Response Message Expected: None

4.5.2.3.7.2 Time Local Form

Source: Time Synchronization
Destination: All Subsystems
Attributes: Digitally Signed
Description: This message includes time synchronized with the external source in a format usable by Core functions.
Response Message Expected: None

4.5.2.3.7.3 Permission

Source: User Permissions
Destination: All Subsystems
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message is a response message to a Check Core/System User/Operator Permissions message. It includes the ID of the Core/System User/Operator, the function they are trying to access and an indication of whether it is permitted the action.
Response Message Expected: None

4.5.2.3.7.4 Locally Encrypted Message

Source: User Trust Management
Destination: All Subsystems
Attributes: Digitally Signed, Secure and Acknowledgement Required
Description: This message is a response message to a Remotely Encrypted Message. It includes a locally encrypted form of the original message, using an encryption algorithm shared between subsystems.
Response Message Expected: None

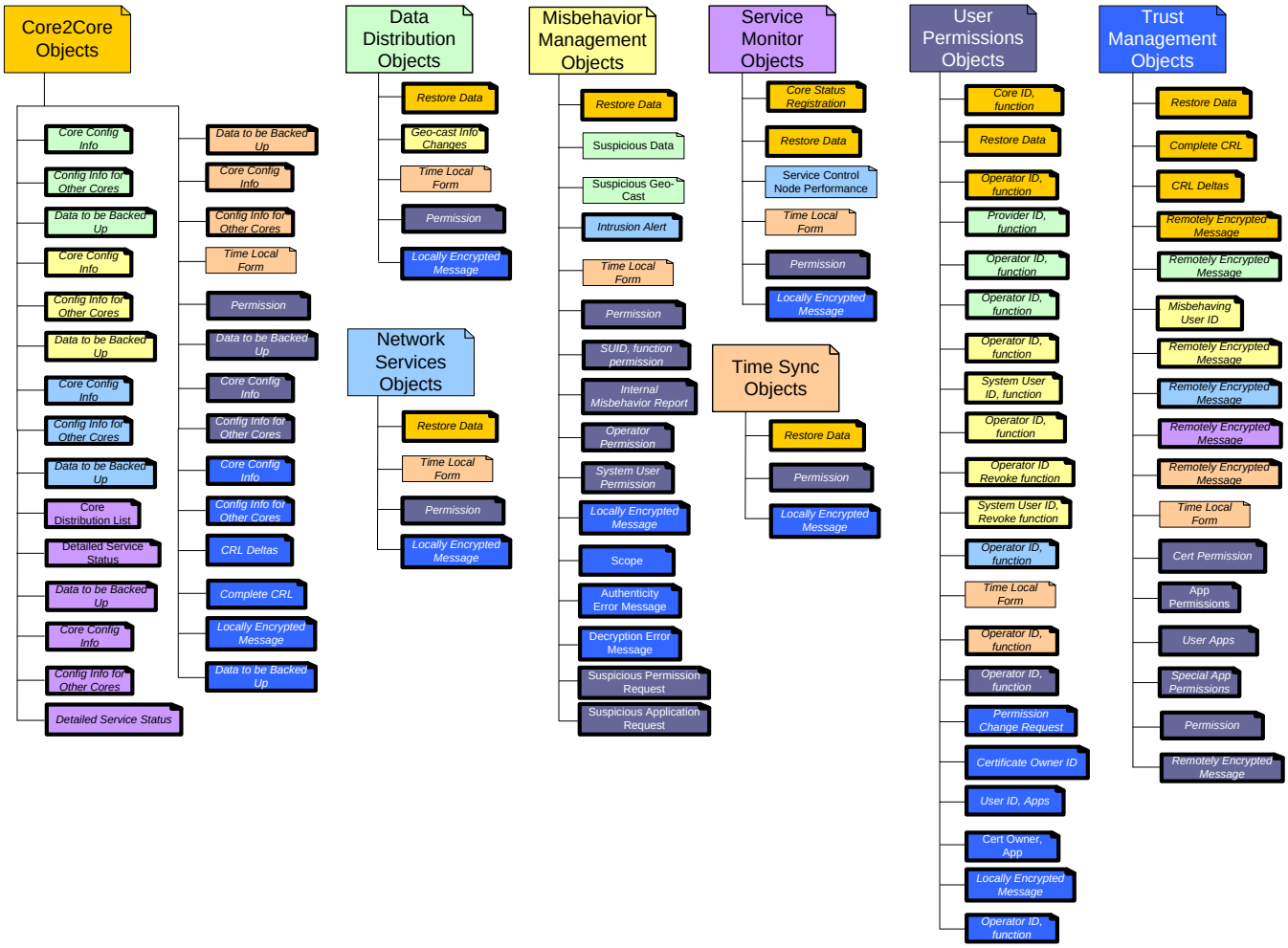


Figure 4-32: Information View – Subsystem-to-Subsystem Objects

4.5.2.4 Configuration Information

The following views must be considered when making changes to this view:

Functional Viewpoint: Functional View – Top Level
Functional Viewpoint: Functional View – Data Distribution
Functional Viewpoint: Functional View – System Configuration
Functional Viewpoint: Functional View – User Configuration
Functional Viewpoint: Functional View – System Monitor and Control
Functional Viewpoint: Functional View – Credentials Distribution
Functional Viewpoint: Functional View – Misbehavior Management
Functional Viewpoint: Functional View – Core Decryption
Functional Viewpoint: Functional View – Networking
Functional Viewpoint: Functional View – Core Backup
Information Viewpoint: Information View – Top Level External Objects
Information Viewpoint: Information View – Top Level Internal Objects

5.0 CONSISTENCY AMONG ARCHITECTURAL VIEWS

One key consideration in an architecting process concerns view consistency. IEEE 1471 requires that each view be consistent. But IEEE 1471 allows a conforming architectural description to have two views, each of which is complete, to be inconsistent with each other. This would allow the existence of some software components in a Functional View that are not allocated to a hardware component in a Connectivity View. Clearly cross-view consistency is a goal, but this is often too difficult to achieve in practice. Instead, the requirement is to record known inconsistencies and provide an analysis of the inconsistencies.

The architecture documented in this SAD has no known internal inconsistencies.

6.0 ARCHITECTURAL RATIONALE AND DISCARDED ALTERNATIVES

This section describes trade-offs considered, alternatives not chosen, and other analyses that led to choosing the architecture described in this System Architecture Document. Objects are not defined in this chapter, except for those objects unique to the illustrated view. For the definition of objects, see the relevant selected view in chapter 4.

Some of the colors used to differentiate objects in this section are different because presentation formats changed during the course of SAD development:

- Domains in Enterprise Views are represented by yellow ovals.
- User Trust Management objects are a different shade of blue with black text.

6.1 Enterprise View – Security Credentials Distribution

Several alternative approaches were considered for the distribution of IEEE 1609.2 certificates used for DSRC communications by Mobile and Field Users. Ultimately, it was decided that the distribution of DSRC security credentials be done by an External Support System acting as the certificate authority. This section describes alternatives that could be pursued if it was decided that the Core System should be the CA for DSRC credentials.

6.1.1 Core System as CA

In this view the Core System would serve as Certificate Authority for IEEE 1609.2 DSRC certificates, including both anonymous and identity certificates. One Core would serve as root CA. Since only one Core can serve as root CA, other Cores would then need to all agree to participate in the CA hierarchy. The identity of the root CA Core would be TBD.

The Core System would not serve as registration authority, instead relying on an External Support System, the DSRC Device Registrar, to provide that functionality. This would allows the Core to deal with certificate distribution without having direct knowledge of the identity of the requester.

The DSRC Device Registrar would enter into an agreement with the owner of the device to provide and install certificates. The DSRC Device Registrar would then request certificates from the Core CA. If the Core CA agreed that the request is valid and the DSRC Device Registrar is authorized to request certificates, the CA distribute certificates. (Depending on the functional architecture selected, the CA may instead distribute a cryptographic key which the DSRC Device could use to unlock pre-loaded certificates.)

Other entities involved would provide the basis for exchanging data. Policies and best practice procedures would come from the trade organizations and bodies (often but not always governmental) responsible for the geographic area over which the Core System operates. Standards bodies would provide the standards that define the interfaces between the CA and Core System Manager, Core System Manager and Device, and also the format and attributes of certificates.

ESS DSRC Device Registrar: The ESS DSRC Device Registrar would take the role of RA in the public-key infrastructure architecture to support DSRC certificate distribution.

Relationships:

- The DSRC Device Registrar receives the DSRC Device Registrar Certification Plan, certification results and periodic re-certification results from the Core Certifying Body.

- The DSRC Device Registrar maintains a certificate distribution agreement with the Core System Manager. It sends certificate requests on behalf of the DSRC Device, including requests for special permissions related to specific applications to the Core System Manager, and receives device certificates with the associated permissions for qualified DSRC Devices from the Core System Manager. For requests of special application permissions, the DSRC Device Registrar may have to provide identity credentials for the DSRC Device Owner depending on local policies. Other certificate requests may not require identity credentials to be provided to the CA.
- The DSRC Device Registrar enters into an agreement with the DSRC Device Owner so the DSRC Device Registrar can obtain and configure certificates for the DSRC Device.
- The DSRC Device Registrar receives certificate requests from the DSRC Device. For requests of special application permissions, the DSRC Device Registrar may request that the DSRC Device provide identity credentials for the DSRC Device Owner depending on local policies.

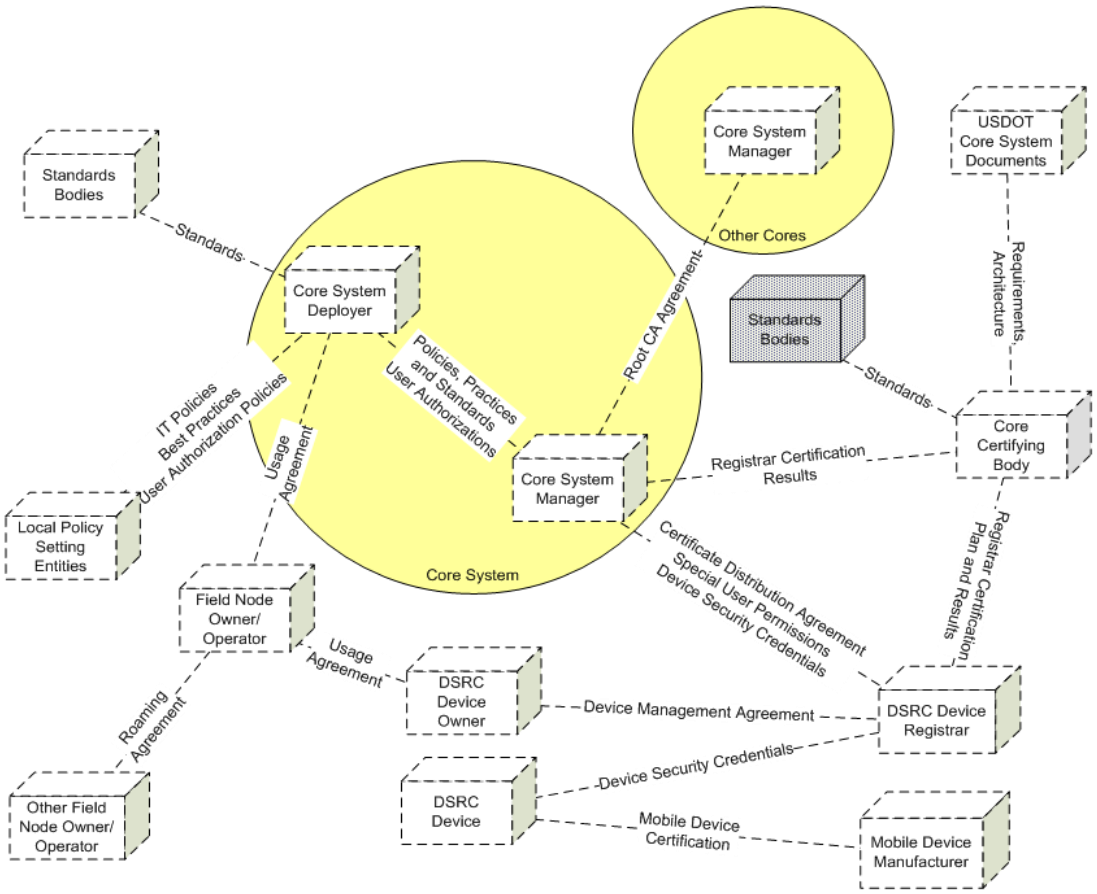


Figure 6-1: Unselected Enterprise View – Security Credentials Distribution, Core CA

6.1.2 Core System as RA

Another alternative ultimately not selected would be to allow some Core Systems to serve as CA as depicted in Figure 6-1 and allow other Cores to serve as RA as shown in Figure 6-2. This approach essentially treats an RA-Core as a DSRC Device Registrar. If this is done, then both Figure 6-1 and Figure

6-2 are valid Views. This approach was originally recommended but not adopted because it was considered too complex.

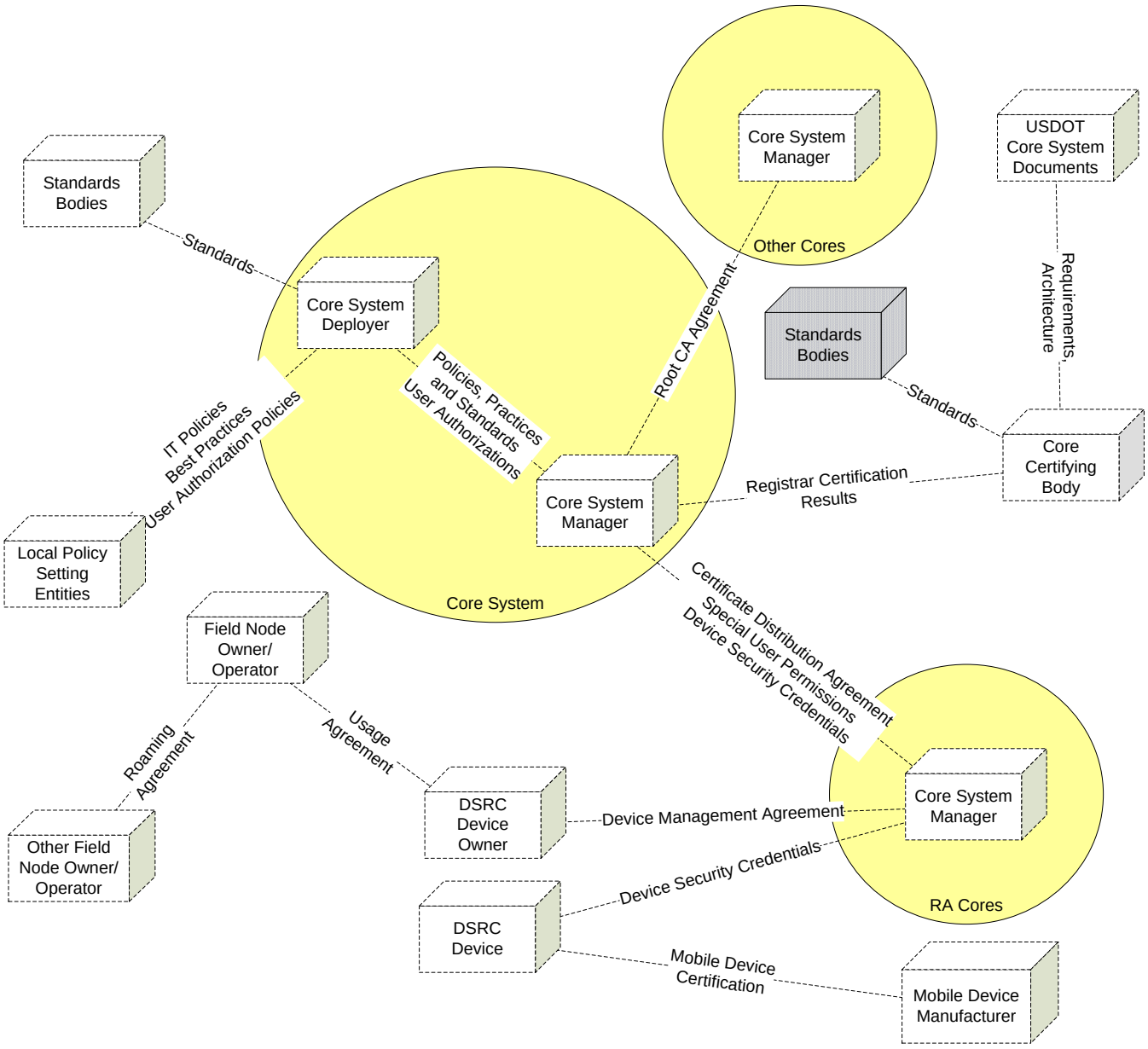


Figure 6-2: Unselected Enterprise View – Security Credentials Configuration, Core RA

6.1.3 Core System as CA and RA

The final alternative for this view would leave the CA functionality in the Core, but removes the DSRC Device Registrar and moves RA functionality into the Core. This does not simplify much, as shown in Figure 6-1. This approach does remove the need to establish monitoring agreements with the DSRC Device Registrar. Unfortunately with both the RA and the CA inside the Core, the identity of the DSRC Device Owner would be known by the same enterprise that issues his digital certificates. This would make identifying DSRC Device Operators simpler, which could possibly lead to compromise of previously established privacy principles. Consequently, this approach was not recommended.

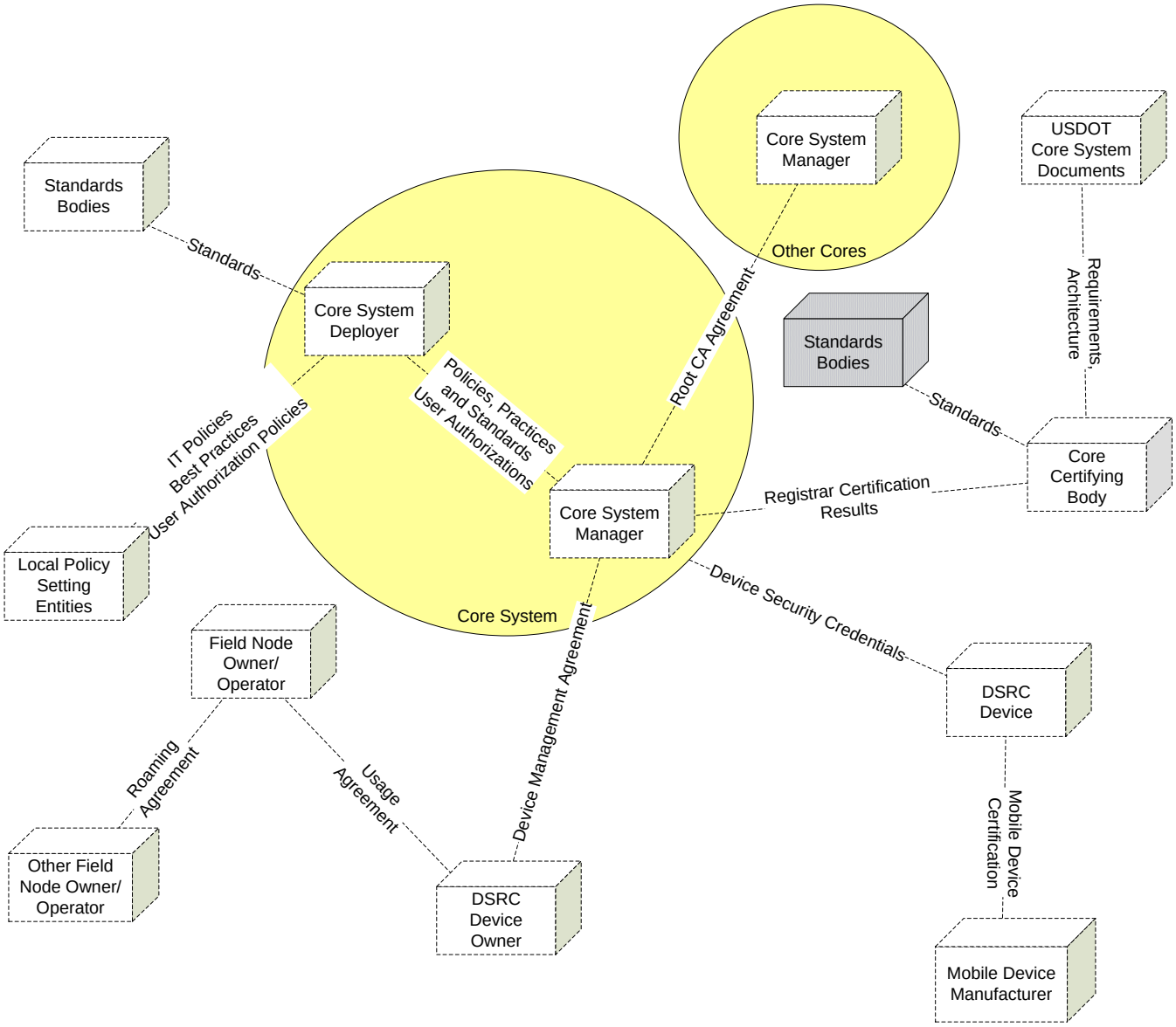


Figure 6-3: Unselected Enterprise View – Security Credentials Distribution, CA and Internal RA

6.2 Enterprise View – Governance

While not selected, an alternative to the Core Certification Authority acting as gatekeeper to entry of Core Systems into operations would be to simply allow the market to sort it all out. Cores would get their root certificates from 3rd party providers (which would still need to be approved just to make sure they complied with the requirements in USDOT Core System Documents, though this would not be strictly necessary since there are today several reputable X.509 CAs in operation).

This alternative would rely on System Users and other Core Systems to differentiate between high performing Cores and lesser performing Cores. Cores that don't perform well and don't meet System Users' needs would not receive as much interaction with System Users, and thus be pressured to improve their operations. Similarly, those that are not good partners with other Cores would not have sharing or backup relationships.

This approach would largely remove the need for the Core Certification Authority. It could even be completely eliminated, saving the expense of its management. It may be easier for an entity that wants to operate a Core to deploy it, since it won't have to interact with the Core Certification Authority.

On the other hand, without the Core Certification Authority controlling entrance to the Core System market, there would be no guarantee that Cores would operate the same way; they may not even be compliant with this architecture. This could jeopardize interoperability between Cores and the availability of digital certificates for Mobile Users. Mobile Users that could not get obtain digital certificates might not receive the same level of safety improvements in the *connected vehicle* environment as those that do have certificates.

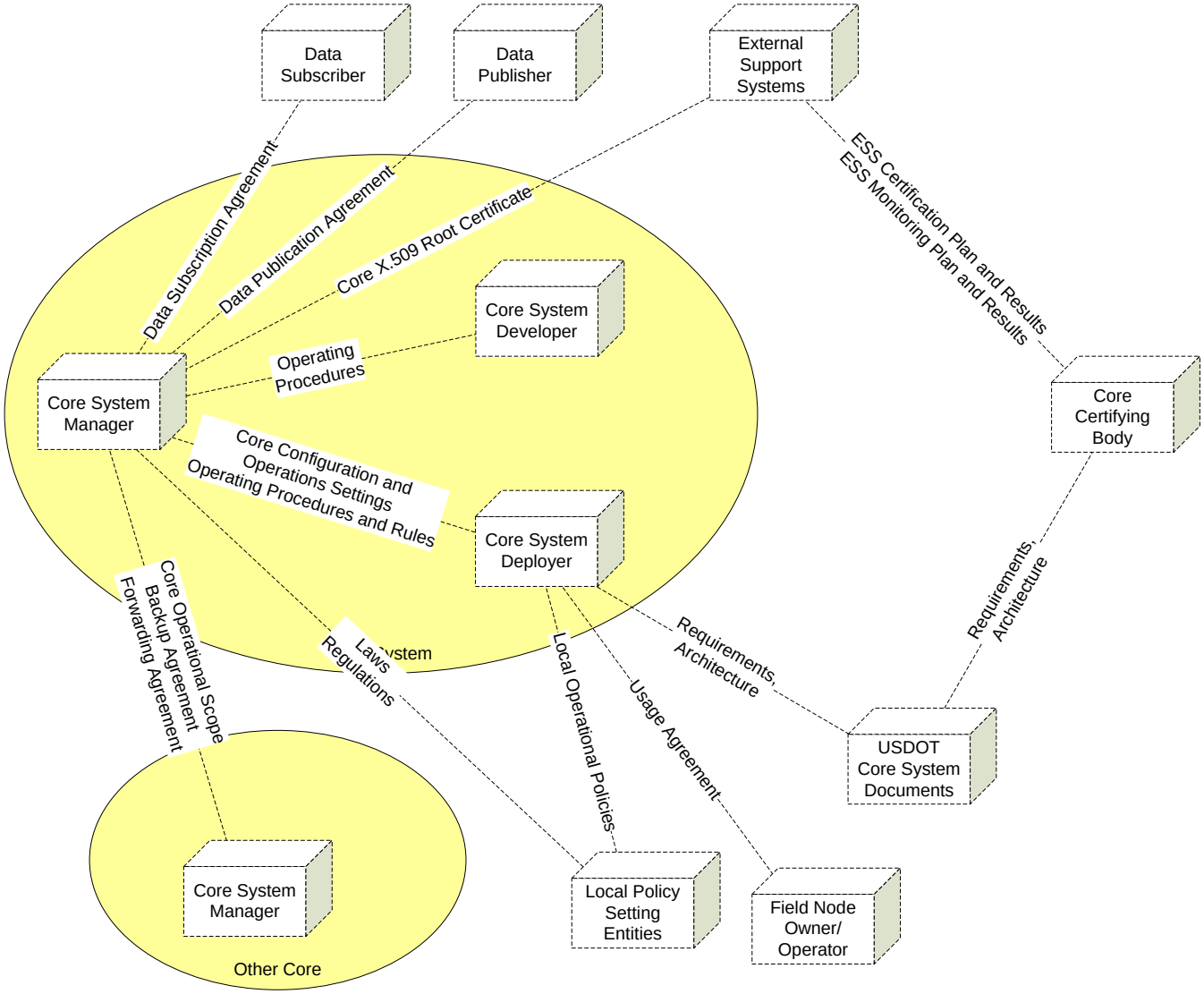


Figure 6-4: Unselected Enterprise View – Governance Market Alternative

6.3 Functional View: Credentials Distribution

Several alternative approaches were considered for the distribution of IEEE 1609.2 certificates used for DSRC communications by Mobile and Field Users. This section contains the alternative Functional Views that were not selected.

6.3.1 Core System as CA

This Functional View corresponds to the Enterprise View where the Core System would serve as CA. It combines Credentials Distribution and Misbehavior Management in one view.

This view includes the functions necessary for the Core to act as a DSRC Certificate Authority (i.e., distribution and management of DSRC certificates) and to detect and mitigate misbehavior. Within those broad high level functions there would be the following activities:

- Distribution of DSRC certificates includes the distribution of both anonymous and identity certificates (including the generation of the public and private keys necessary to encrypt and decrypt messages) and the maintenance and distribution of CRLs. It also includes the exchange of certificate distribution coordination information between Cores and the ability to restrict certificate distribution when the Core changes operational state.
- Detection of misbehavior includes the detection of misbehavior by Field, Center, and Mobile Users and Operators by using data reported by non-anonymous System Users, and detection mechanisms inside the Core. Misbehavior detection can be done independently of certificate distribution. However, acting on detection of misbehavior includes modification of the CRL which can only effectively be done by the entity responsible for certificate distribution. Therefore functions included in misbehavior detection are restricted in the alternative considering CA functionality outside the Core.
- Misbehaving operators will not have DSRC certificates and in fact may be associated with X.509 certificates. However detection of misbehaving operators is shown on this view because most of the other misbehavior-related functions are shown here.

Both certificate distribution and misbehavior management would require coordination between Cores to ensure that CRLs were consistent, that no two Cores issued the same certificates, and to improve misbehavior detection by sharing misbehavior reports between Cores. If more than one Core served as DSRC Certificate Authority, only one Core could serve as Root. Other Cores would serve as sub-CAs.

Several additional objects are defined:

6.3.1.1 Exchange CRLs with other Cores

This object allows the Core to exchange CRLs with other Cores. This allows each Core to maintain a complete CRL, so that System Users need receive a CRL from only one source.

Associated Information Objects:

- **Complete CRL** contains the IDs of all certificates the sending Core knows to be invalid. This is sent by the Core or received from another Core.
- **CRL Deltas** describes the changes to the CRL since the last CRL Delta or Complete CRL was sent to the Core that is receiving this message.

6.3.1.2 Maintain DSRC Anonymous Certificates

This object maintains the inventory of DSRC anonymous certificates distributed by the Core including the key pairs associated with these certificates. It also maintains a list of the identifiers of identity certificates issued by other Cores. It provides these certificates upon valid request.

Associated Information Objects:

- **Anonymous Cert Request** is a request for a new groups of anonymous certificates sent from the Provide DSRC Anonymous Certificates function on behalf of a System User.
- **DSRC Anonymous Certs** contains one or more new anonymous DSRC certificates for the System User.
- **Activated DSRC Anon Cert IDs** contain the IDs of the DSRC anonymous certificates issued to a System User by other Cores; provided by the Coordinate Certificate Distribution with Other Cores function.

6.3.1.3 Maintain DSRC Identity Certificates

This object maintains the inventory of DSRC identity certificates distributed by the Core including the key pairs associated with these certificates. It also maintains a list of the identifiers of identity certificates issued by other Cores. It provides these certificates upon valid request.

Associated Information Objects:

- **Activated DSRC Identity Cert IDs** contain the IDs of the DSRC identity certificates issued to a System User by other Cores; provided by the Coordinate Certificate Distribution with Other Cores function.
- **App Permissions** contains the application permissions for the application queried in Cert Owner, App, received from the Application Permissions Registry.
- **Cert Owner, App** is a query sent to the Application Permissions Registry function asking what the application permissions are for a given application.
- **DSRC Identity Cert Request** is a request for a new identity certificate sent from the Provide DSRC Identity Certificates function on behalf of a System User.
- **DSRC Identity Certs** contains one or more new identity certificates for the System User.

6.3.1.4 Manage DSRC CRL

This object maintains the Certificate Revocation List for all DSRC certificates, including both anonymous and identity certificates. It adds entries to the CRL based on inputs from Manually Confirm/Identify Misbehaving Users. It responds to queries about associations between System User IDs and pseudo-IDs with entries in the CRL.

Associated Information Objects:

- **Anon Cert Pseudo-ID** is received from Provide DSRC Anonymous Certificates; it includes a pseudo-ID associated with an anonymous certificate.
- **Anon CRL Associations** is a response to the Provide DSRC Anonymous Certificates function that indicates the certificate IDs that are on the CRL and are associated with the pseudo-ID previously provided.
- **CRL** is the complete certificate revocation list.
- **CRL Deltas** contains changes to the CRL since the last time it was published.
- **Misbehaving User ID** contains the ID and type of misbehavior the System User has committed and is received from the Manually Confirm/Identify Misbehaving Users function.

- **System User ID** is received from Provide DSRC Identity Certificates and includes the identity of a System User.
- **System User CRL Associations** is a response to the Provide DSRC Identity Certificates function that indicates the certificate IDs that are on the CRL and are associated with the System User ID previously provided.

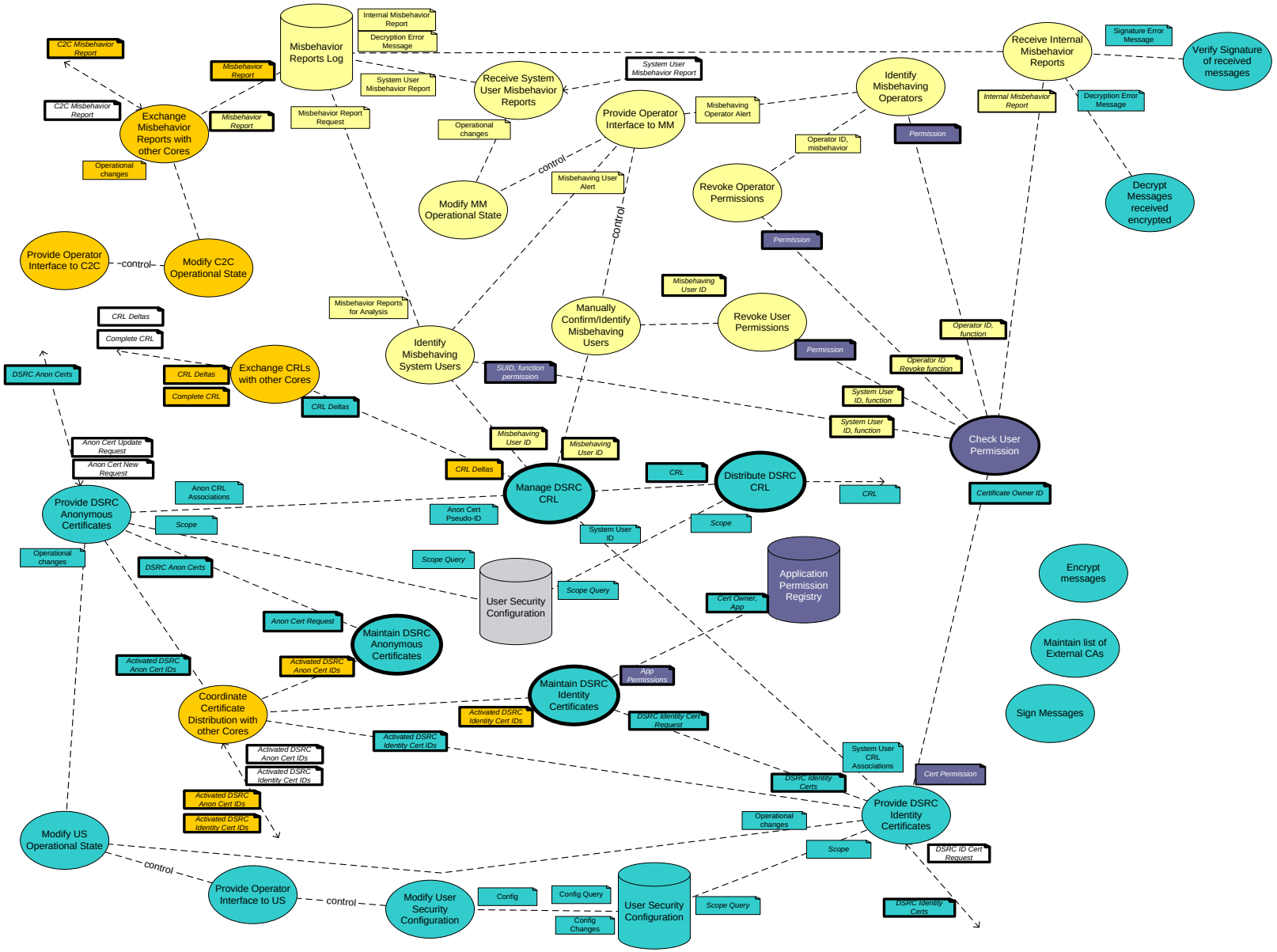


Figure 6-5: Unselected Functional View – IEEE 1609.2 Certificate Distribution and Misbehavior Management

6.3.2 Core System as CA, Pre-loaded Certificates

This alternative was developed by considering that IEEE 1609.2 certificates could be pre-assigned to DSRC users, and stored encrypted in an inactive state. Certificates would then activated upon request and response from the Core. In this case, certificates would not need to be transmitted, but activation codes would. The information objects DSRC Anon Certs and DSRC Identity Certs would become activation codes instead of the complete certificate. There are no other changes to the Figure 6-5. This approach would be compatible with the approach documented above; i.e. some System Users could have pre-loaded certificates while others would need the Core to provide them the whole certificate.

6.3.3 Different RA and CA Core Systems

In another variation, it was considered that registration could be performed outside the Core that would perform certification. In other words, one Core would act as a CA, and another Core would function as RA. For the CA Core the functions Provide DSRC Identity Certificates and Provide DSRC Anonymous Certificates would be modified by removing the verification checking from those functions. The link between Provide DSRC Identity Certificates and Check User Permissions would disappear and be replaced by a similar link from Maintain DSRC Identity Certificates, used to determine the permissions to attach to identity certificates. There are no other changes to the CA-Core's diagram.

This approach would have the advantage of splitting knowledge of a System User's true identity between two entities (CA and RA), helping to ensure that the System User's privacy would be difficult to compromise. This approach would be compatible with the approach documented in Figure 6-5 as long as at least one Core served as CA.

Figure 6-6 illustrates the Core that would perform RA functions only. Several functions from the basic architecture are deleted. Three additional objects are defined for that alternative:

6.3.3.1 Provide Misbehaving System User Info to Other Cores

This object provides the results of Misbehavior Management to other Cores. Misbehavior detection must still be handled by the Core because the RA is familiar with the user's identity and roles, not the CA. This function provides the CA-core with the ID of misbehaving System Users and provides information necessary to revoke or modify their certificate.

Associated Information Objects:

- **Misbehaving User ID** contains the ID and type of misbehavior the System User has committed. Depending on the method of identification, the contents could include a user ID, a certificate ID or a pseudo-ID. This message is provided to other Cores that perform CA functions for further action.

6.3.3.2 Request Certificate

This object accepts a certificate request with associated permissions from the Verify DSRC User Identity function and passes that request to a Core with CA functionality. Certificates received from that Core are provided to System User.

Associated Information Objects:

- **DSRC ID Cert Request** is a request for a new identity certificate from a System User.

- **Certificate Request, Permissions** is sent to another Core with CA functionality requesting certificates with the attached permissions for the System User. This can be a request for an anonymous certificate or an identity certificate.
- **DSRC Anon Certs** contains one or more new anonymous certificates for the System User. This message originates from a Core with CA functionality, is sent to the Request Certificate function, then to the Verify DSRC User Identity function, and is then provided to the System User.
- **DSRC Identity Certs** contains new identity certificates for the System User. This message originates from a Core with CA functionality, is sent to the Request Certificate function, then to the Verify DSRC User Identity function, and is then provided to the System User.

6.3.3.2.1 Verify DSRC User Identity

This object accepts requests for DSRC certificates from System Users, verifies the certificate request by examining the provided certificate and verifying its format and content, and then querying Check User Permissions to verify that the owner of this certificate is allowed to request new certificates. It then passes a message to the Request Certificate function, requesting the certificate for the System User in question.

Associated Information Objects:

- **DSRC ID Cert Request** is a request for a new identity certificate from a System User.
- **Cert Permission** is the response from Check User Permissions indicating whether the System User is allowed to request a DSRC identity certificate, and also the types of permissions that must be attached to the identity certificates.
- **Certificate Request, Permissions** is sent to the Request Certificate function, containing a certificate request and the permissions the System User is entitled to.
- **Certificate Owner ID** is sent to Check User Permissions as a query to determine whether the System User is allowed to request a DSRC identity certificate.

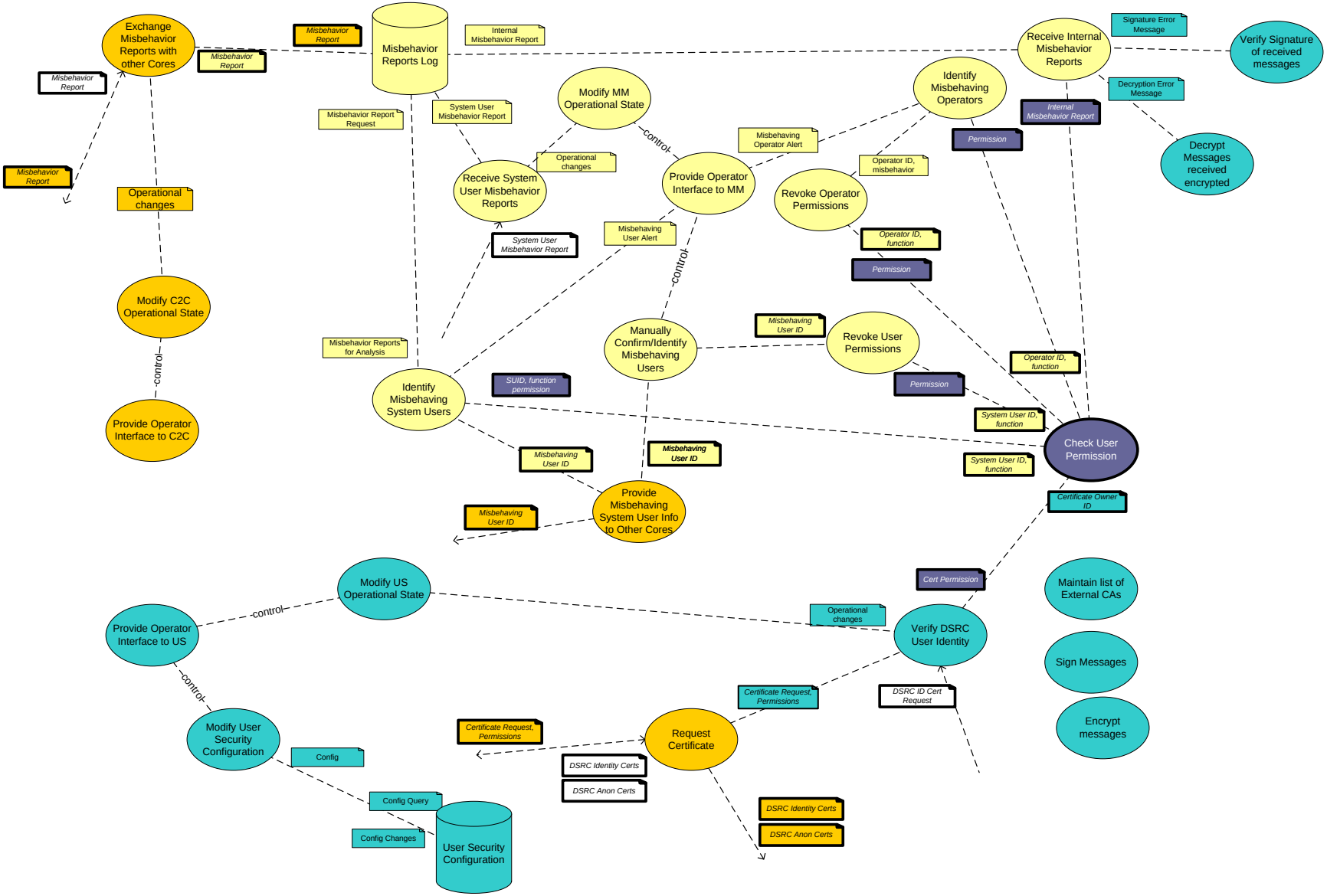


Figure 6-6: Unselected Functional View – IEEE 1609.2 Certificate Distribution, RA Only

6.3.4 External CA for Anonymous DSRC Certificates Only

In this variation IEEE 1609.2 DSRC anonymous certificate distribution could be moved outside the Core but identity certificate distribution would stay inside. This would eliminate the difficulties associated with managing local permissions and enable a common CA for all anonymous System Users. No diagram is provided for this case, as it represents a removal of the basic cases' functionality but no additions. Again, this alternative is not selected since all DSRC certificate distribution has been moved outside the Core.

6.3.5 Multiple Root CAs

In this variation, each Core providing certificates could be configured as a root CA. This makes the job of coordinating certificates more difficult, and changes the security challenges. Since each Core would be a root, if one Core was compromised then effectively all Cores would be compromised until the vulnerability was discovered. (If only one Core served as root, then the compromise of a sub-CA Core would endanger System Users using that Core, but not necessarily other Cores or System Users using other Cores).

6.4 Functional View: Decryption, Connectivity View: High Level and Core Functional Allocation

Alternatives in the area of system security were explored. One alternative had to do with the storing of the Core's private encryption key. Section 4.2.8 described how the Core needs an encryption key in order to decrypt messages sent to it. The selected alternative limited the physical nodes where the key would reside. Another alternative in one in which the Core System's private encryption key would be stored at each node that required the ability to read encrypted messages directed to the Core. This alternative alters the Decryption Functional View because there would be no need for local encryption. It alters the High Level and Core Functional Allocation Connectivity Views because the Decryption Node is removed.

Storing encryption keys at every node increases the exposure of those keys, making it more difficult to secure those keys. For deployments where nodes are not co-located this could be costly. System-wide vulnerability could be reduced by using different keys on different nodes, but this increases the complexity of managing those keys and adds some complexity to the software developed for System Users who wish to communicate with the Core using encrypted messages.

On the positive side, the Core could more easily scale to support more encrypted communications, as the Core Decryptor would not be a single point through which all encrypted communications must pass.

Because of the increased risks to security, this alternative was rejected.

6.5 Connectivity Core Functional Allocation: Common LANs

Section 4.3.2 described the functional allocation and showed the Core Decryptor function operating on a separate LAN from other components. An alternative considered but rejected was one in which the Core Decryptor would operate on the same LAN as the Service Router and SCNs. This approach removed one Core Switch and simplified network configuration, but put encrypted communications traffic on the same network as unencrypted traffic. It was rejected for that reason.

7.0 APPENDICES

This section is a contains the appendix which has a table summarizing the contents of the architecture viewpoints as well as supporting analysis used during the development of the Views in section 4.

7.1 View Point Summary

Table 7-1: Viewpoint Summary

Specifications	Enterprise	Functional	Connectivity	Communication	Information
Overview	Relationship between organizations	Logical interactions between functional objects	Connections between hardware, interfaces, software	Layered communications protocols between nodes	Data object structure, relationship meta data constraints
Defined Terms	• Enterprise Objects • Facilities • Domain • Federation	• Functional Object • Realized Information Objects	• Engineering Object • Node • Link • Port • Application	• Protocol Entity	• Information Object • Metadata • Information Package
Stakeholders	• Users (Mobile, Field , Center) • Operator • Acquirer • Maintainer • Developer • Manager • Policy-Setter • Application Developer • Device Developer • Service Provider	• Users (Mobile, Field , Center) • Operator • Acquirer • Maintainer • Developer • Manager • Tester • Application Developer • Device Developer • Service Provider	• Users (Mobile, Field , Center) • Acquirer • Maintainer • Developer • Tester • Application Developer • Device Developer • Service Provider	• Users (Mobile, Field , Center) • Operator • Acquirer • Maintainer • Developer • Tester • Policy-Setter • Application Developer • Device Developer • Service Provider	• Users (Mobile, Field , Center) • Operator • Acquirer • Maintainer • Developer • Tester • Application Developer • Device Developer
Concerns	• Security • Organization/Resources • Risks • Evolvability • Deployability • Maintainability	• Interfaces • Functionality • Security • Appropriateness • Evolvability	• Performance • Interfaces • Security • Feasibility • Risks • Evolvability • Deployability • Maintainability	• Performance • Interfaces • Functionality • Security • Organization/Resources • Appropriateness • Feasibility • Risks • Evolvability • Deployability	• Security • Interfaces • Applicability
Method(s) to Model Conforming Views	Table 3 2: Enterprise View Graphical Object Definitions	Table 3-3: Functional View Graphical Object Definitions	Table 3-4: Connectivity View Graphical Object Definitions	Table 3-5: Communications View Graphical Object Definitions	Table 3-6: Information View diagrams Graphical Objects Definition
Viewpoint Source	CCSDS 311.0-M-1	CCSDS 311.0-M-1	CCSDS 311.0-M-1	CCSDS 311.0-M-1	CCSDS 311.0-M-1
Security Issues	• Organizational Roles • Policies • Trust Relationships • Domain Boundaries • Cross-support Security Agreements	Access control interfaces on functions and specific functional elements	Physical elements that are used to implement security policies and barriers	Implementation of security protocols between: • Mobile Users and the Core System, • Mobile Users and other System Users	Protection from unauthorized access

7.2 Analysis Views

The following views are illustrated using the conventions of the Connectivity Viewpoint, but used as a basis for analysis of the communications loads that Core2Core, Data Distribution and User Trust Management subsystems experience. Core2Core and User Trust Management are heavily influenced by the assumption that the Core would be distributing IEEE 1609.2 certificates. Thus, these two views are not relevant to the current architecture, but to the rejected alternatives documented in chapter 6.

These views consider the Core Access Node to be part of the Service Router.

7.2.1 Connectivity View – Core2Core Analysis

7.2.1.1 Introduction

This View illustrates Core2Core internal and external interfaces. This View assumes that the Core System is serving as CA but not RA for IEEE 1609.2 certificates and providing Data Distribution services. This View also assumes that each subsystem is placed on a separate SCN. It acknowledges the Core Decryptor as a separate node from User Trust Management.

Table 7-2: Core2Core Interface Analysis Summary

Information Object	Assigned Interface	Bandwidth Minimum (Mbits/sec)
SUMMARY I-C2CUTM	I-C2CUS	14.81557
SUMMARY I-C2CCD	I-C2CCD	0.00160
SUMMARY I-C2CMM	I-C2CMM	0.00381
SUMMARY I-C2CSM	I-C2CSM	0.00034
SUMMARY I-C2CTS	I-C2CTS	0.00000
SUMMARY I-C2CNS	I-C2CNS	0.00089
SUMMARY I-C2CDD	I-C2CDD	0.44444
SUMMARY I-C2CUP	I-C2CUP	0.44444
SUMMARY EXTERNAL	E-C2C	15.32211
SUMMARY I-C2CNS+EXT	I-C2CSR	15.32300

Assumptions: 5 Cores, 100 MB backup data, 100M Mobile serviced with linked pre-distributed certificates, daily contact to activate certificates

Information Object	Assigned Interface	Bandwidth Minimum (Mbits/sec)
SUMMARY I-C2CUTM	I-C2CUS	18.35240
SUMMARY I-C2CCD	I-C2CCD	0.03960
SUMMARY I-C2CMM	I-C2CMM	0.09431
SUMMARY I-C2CSM	I-C2CSM	0.00840
SUMMARY I-C2CTS	I-C2CTS	0.00002
SUMMARY I-C2CNS	I-C2CNS	0.02200
SUMMARY I-C2CDD	I-C2CDD	11.00000
SUMMARY I-C2CUP	I-C2CUP	11.00000
SUMMARY EXTERNAL	E-C2C	40.57792
SUMMARY I-C2CNS+EXT	I-C2CSR	40.59992

Assumptions: 100 Cores, 100 MB backup data, 100M Mobile serviced with linked pre-distributed certificates, daily contact to activate certificates

7.2.1.2 Object Definitions and Roles

Nodes

Core Decryptor: This node operates the Core's decryption and local encryption SEOs.
Core2Core Node: This node operates Core2Core SEOs that implement all Core2Core functions. This could include multiple sub-nodes each running a portion of Core2Core software. It has links to Core Decryptor, Service Router, Misbehavior Management, Service Monitor, Time Sync, User Permissions and User Trust Management nodes. It also operates the local decryption component of User Trust Management.
Data Distribution Node: This node runs Data Distribution SEOs. This could include multiple sub-nodes each running a portion of Data Distribution software.
Misbehavior Management Node: This node operates Misbehavior Management SEOs. This could include multiple sub-nodes each running a portion of Misbehavior Management software.
Service Monitor Node: This node operates

Service Monitor SEOs. This could include multiple sub-nodes each running a portion of Service Monitor software.
Service Router Node: This node operates Network Services Engineering Objects. It provides the interface between the C2C Node and System Users.
Time Sync Node: This node operates Time Synchronization SEOs.
User Permissions: This node operates User Permissions SEOs.
User Trust Management Node: This node operates User Trust Management SEOs. This could include multiple sub-nodes each running a portion of User Trust Management software.

Links

I-C2CCD: The internal interface between Core2Core and Core Decryptor Nodes.
I-C2CMM: The internal interface between Core2Core and Misbehavior Management Nodes.
I-C2CSM: The internal interface between Core2Core and Service Monitor Nodes.
I-C2CTS: The internal interface between Core2Core and Time Synchronization Nodes.
I-C2CNS: The internal interface between Core2Core and Service Router (Network Service) Nodes. For the purposes of the network loading analysis this includes only the data exchanged between the engineering objects that make up the Network Services and Core2Core subsystems. Network design must consider that the loading on this physical interface is equal to the sum of the loads on this interface and the E-C2C interface. This is noted in Table 7-2 as I-C2CSR
I-C2CDD: The internal interface between Core2Core and Data Distribution Nodes.
I-C2CUP: The internal interface between Core2Core and User Permission Nodes.
I-C2CUTM: The internal interface between Core2Core and User Trust Management Nodes.
E-C2C: The external interface Core2Core and the Internet.

7.2.1.3 View Description

Table 7-2 summarizes two samples of the communications analysis for the Core2Core subsystem. For planning purposes this analysis assumes that 100 MB of backup data would be shared with each Core that was participating in backup exchanges, but that no Core would participate with more than 9 other Cores. It also assumes that each Core would share in the distribution of activation codes for 1609.2 certificates to 100 million DSRC-equipped Mobile devices. A detailed breakdown of the message passing that supports this analysis tables can be found in Table 7-3.

Core backup and DSRC certificate activation are the primary drivers of connectivity requirements for the Core2Core subsystem. While the number of backups required can be easily limited by operational policies and agreements, the number of vehicles that must be supported is less predictable. While Cores do not pass along probe data between Cores, they do maintain a list of what data each Core accepts, and a duplicate of the subscription database for those Cores that have service takeover agreements for Data Distribution. Still, this database is not a significant driver of Core2Core communications.

Figure 7-1 shows the relationship between the number of Mobile devices that require DSRC certificate activations and the Core2Core subsystem's external interface bandwidth requirements. This graph assumes 100 million Mobile DSRC devices requiring IEEE 1609.2 certificates, activated daily using a scheme similar to that recommended by the *CAMP Security Final Report* recommendations. As the number of Core Systems increases so does the certificate-related load on the C2C interface, as each Core sends certificate activation notifications to each other Core. However, since each Core added decreases

the number of devices that a given Core must support, the load asymptotically approaches a value of 18.5 Mbps.

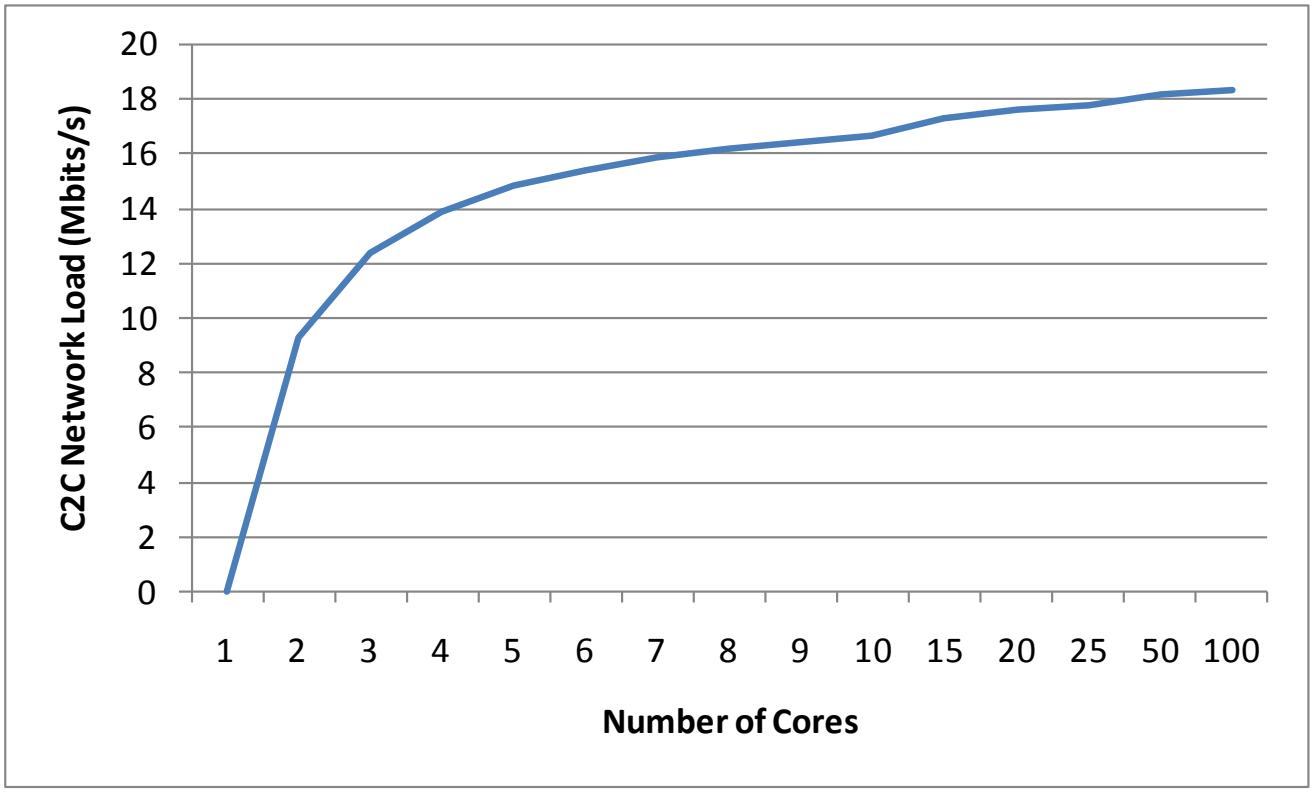


Figure 7-1: External Interface Bandwidth Requirements Supporting 1609 Certificate Activation

Figure 7-2 illustrates the Connectivity View from the perspective of the Core2Core subsystem.
As noted in the discussion above, only the external interface shows the potential for difficulty in meeting its requirements. There is not much that can be done about this architecturally; the external interface simply must move this data. This does however bring attention to the notion of federated Cores, and the roles that each Core plays. Clearly the Core2Core interface requirements will require fewer network resources if Cores do not need to share information about which DSRC certificates have been activated.

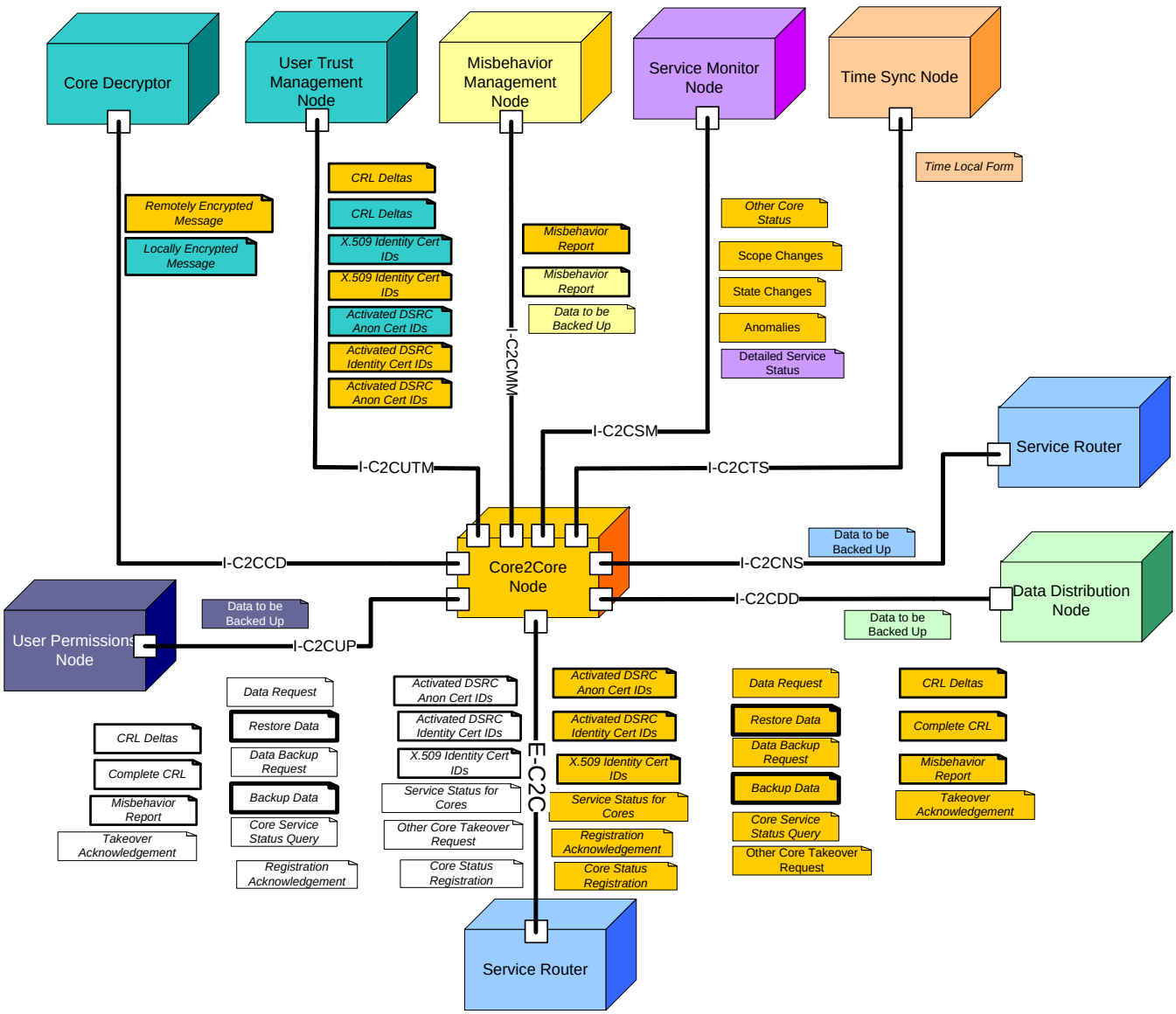


Figure 7-2: Analysis Connectivity View – Core2Core

Table 7-3: Core2Core Interface Analysis, 100 Cores

Information Object	Approximate Message Size (bytes)	Messages/ hour	# of Cores	Approximate Frequency (msg/sec)	Origin	Destination	Assigned Interface	Bandwidth Minimum (Mbits/sec)
CRL Deltas	10000	4	100	0	C2C	US	I-C2CUS	0.00880
CRL Deltas	10000	4	100	0	US	C2C	I-C2CUS	0.00880
X.509 Identity Cert IDs	1000	1	100	0	C2C	US	I-C2CUS	0.00022
X.509 Identity Cert IDs	1000	1	100	0	US	C2C	I-C2CUS	0.00022
Activated DSRC Anon Cert IDs	1000	41,667	100	0	C2C	US	I-C2CUS	9.16674
Activated DSRC Anon Cert IDs	1000	41,667	100	0	US	C2C	I-C2CUS	9.16674
Activated DSRC Identity Cert IDs	1000	4	100	0	C2C	US	I-C2CUS	0.00088
SUMMARY I-C2CUTM							I-C2CUS	18.35240
Remotely Encrypted Message	1500	60	100	0	C2C	CD	I-C2CCD	0.01980
Locally Encrypted Message	1500	60	100	0	CD	C2C	I-C2CCD	0.01980
SUMMARY I-C2CCD							I-C2CCD	0.03960
Misbehavior Report	100	60	100	0	C2C	MM	I-C2CMM	0.00132
Misbehavior Report	100	60	100	0	MM	C2C	I-C2CMM	0.00132
Data to be Backed Up	10000000	0.04	100	0	MM	C2C	I-C2CMM	0.09167
SUMMARY I-C2CMM							I-C2CMM	0.09431
Other Core Status	1000	4	100	0	C2C	SM	I-C2CSM	0.00088
Scope Changes	10000	1	100	0	C2C	SM	I-C2CSM	0.00220
State Changes	200	1	100	0	C2C	SM	I-C2CSM	0.00004
Anomalies	200	60	100	0	C2C	SM	I-C2CSM	0.00264
Detailed Service Status	3000	4	100	0	C2C	SM	I-C2CSM	0.00264
SUMMARY I-C2CSM							I-C2CSM	0.00840
Time Local Form	100	1	100	0	T	C2C	I-C2CTS	0.00002
SUMMARY I-C2CTS							I-C2CTS	0.00002
Data to be Backed Up	100000	1	100	0	T	C2C	I-C2CNS	0.02200
SUMMARY I-C2CNS							I-C2CNS	0.02200
Data to be Backed Up	50000000	1	100	0	T	C2C	I-C2CDD	11.00000
SUMMARY I-C2CDD							I-C2CDD	11.00000
Data to be Backed Up	50000000	1	100	0	T	C2C	I-C2CUP	11.00000
SUMMARY I-C2CUP							I-C2CUP	11.00000
CRL Deltas	10000	4	100	0	C2C	External	E-C2C	0.00880
Complete CRL	5000000	1	100	0	C2C	External	E-C2C	1.10000
Misbehavior Report	100	60	100	0	C2C	External	E-C2C	0.00132
Takeover Acknowledgement	1000	1	100	0	C2C	External	E-C2C	0.00022
Data Request	100	0.01	100	0	C2C	External	E-C2C	0.00000
Restore Data	110100000	0.01	100	0	C2C	External	E-C2C	0.24222
Data Backup Request	100	0.4	100	0	C2C	External	E-C2C	0.00001
Backup Data	110100000	0.4	100	0	C2C	External	E-C2C	9.68880
Core Service Status Query	1000	60	100	0	C2C	External	E-C2C	0.01320
Other Core Takeover Request	1000	1	100	0	C2C	External	E-C2C	0.00022
Activated DSRC Anon Cert IDs	1000	41,667	100	0	C2C	External	E-C2C	9.16674
Activated DSRC Identity Cert IDs	1000	4	100	0	C2C	External	E-C2C	0.00088
Core Status Registration	1000	1	100	0	C2C	External	E-C2C	0.00022
Core Status Registration Acknow	1000	1	100	0	C2C	External	E-C2C	0.00022
X.509 Identity Cert IDs	1000	1	100	0	C2C	External	E-C2C	0.00022
Service Status for Cores	5000	60	100	0	C2C	External	E-C2C	0.06600
CRL Deltas	10000	4	100	0	External	C2C	E-C2C	0.00880
Complete CRL	5000000	1	100	0	External	C2C	E-C2C	1.10000
Misbehavior Report	100	60	100	0	External	C2C	E-C2C	0.00132
Takeover Acknowledgement	1000	1	100	0	External	C2C	E-C2C	0.00022
Data Request	100	0.01	100	0	External	C2C	E-C2C	0.00000
Restore Data	110100000	0.01	100	0	External	C2C	E-C2C	0.24222
Data Backup Request	100	0.4	100	0	External	C2C	E-C2C	0.00001
Backup Data	110100000	0.4	100	0	External	C2C	E-C2C	9.68880
Core Service Status Query	1000	60	100	0	External	C2C	E-C2C	0.01320
Other Core Takeover Request	1000	1	100	0	External	C2C	E-C2C	0.00022
Activated DSRC Anon Cert IDs	1000	41667	100	0	External	C2C	E-C2C	9.16674
Activated DSRC Identity Cert IDs	1000	4	100	0	External	C2C	E-C2C	0.00088
X.509 Identity Cert IDs	1000	1	100	0	External	C2C	E-C2C	0.00022
Other Core Takeover Request	1000	1	100	0	External	C2C	E-C2C	0.00022
Service Status for Cores	5000	60	100	0	External	C2C	E-C2C	0.06600
SUMMARY EXTERNAL							E-C2C	40.57792
SUMMARY I-C2CNS+EXT							I-C2CSR	40.59992

Assumptions: 100 Cores, 100 MB backup data, 100M vehicles serviced with linked pre-distributed certificates, daily contact to activate certificates

7.2.2 Connectivity View – Data Distribution Analysis

7.2.2.1 Introduction

This View illustrates Data Distribution internal and external interfaces. This View assumes that each subsystem is placed on a separate Node. It acknowledges the Core Decryptor as a separate node from User Trust Management. It assumes there is no Network Services Node, but there is a separate Service Router Node that implements the Network Services Engineering Objects.

7.2.2.2 Object Definitions and Roles

Nodes

Core Decryptor: This node operates the Core's decryption and local encryption functions.
Core2Core Node: This node operates Core2Core SEOs that implement all Core2Core functions. This could include multiple sub-nodes each running a portion of Core2Core software.
Data Distribution Node: This node runs Data Distribution SEOs. This could include multiple sub-nodes each running a portion of Data Distribution software. It has links to Core Decryptor, Service Router, Misbehavior Management, Service Monitor, Time Sync, User Permissions and Core2Core nodes. It also operates the local decryption component of User Trust Management.
Misbehavior Management Node: This node operates Misbehavior Management SEOs. This could include multiple sub-nodes each running a portion of Misbehavior Management software.
Service Monitor Node: This node operates Service Monitor SEOs. This could include multiple sub-nodes each running a portion of Service Monitor software.
Time Sync Node: This node operates Time Synchronization SEOs.
User Trust Management Node: This node operates User Trust Management SEOs. This could include multiple sub-nodes each running a portion of User Trust Management software.

Links

I-DDCD: The internal interface between Data Distribution and Core Decryptor Nodes.
I-DDMM: The internal interface between Data Distribution and Misbehavior Management Nodes.
I-DDSM: The internal interface between Data Distribution and Service Monitor Nodes.
I-DDTS: The internal interface between Data Distribution and Time Synchronization Nodes.
I-DDSR: The internal interface between Data Distribution and Service Router (Network Service) Nodes.
I-DDC2C: The internal interface between Data Distribution and Core2Core Nodes.
I-DDUP: The internal interface between Data Distribution and User Permission Nodes.
E-DD: The external interface Data Distribution and other the Internet. Physically equivalent to I-DDSR.

7.2.2.3 View Description

Table 7-4 summarizes two samples of the communications analysis for the Data Distribution subsystem. For planning purposes this analysis considers only Mobile User data providers. It assumes that one million Mobile Users each provide 100 messages per hour, with 15 System User subscribers to the entire data feed. 100 messages is a reasonable number for Mobile Users given a probe generation scheme similar to the one outlined in SAE J2735 and used in the VII Proof of Concept. Fifteen was chosen as a realistic number of transportation-related Centers operating in the Core System's service area that would want to subscribe to the data, based on regional ITS architectures for moderately sized metropolitan areas. The first table assumes that aggregation or sampling is used to collapse message contents so that for every 100 messages provided by System Users, one is sent to each subscriber with the aggregated or sampled content. The second table assumes that no aggregation or sampling is used. A detailed breakdown of the message passing that supports this analysis can be found in the Appendices, Table 7-5 and Table 7-6.

Table 7-4: Data Distribution Interface Analysis Summary

Information Object	Assigned Interface	Bandwidth Minimum (Mbits/sec)
SUMMARY I-DDCD	I-DDCD	0.00040
SUMMARY I-DDMM	I-DDMM	111.11222
SUMMARY I-DDSM	I-DDSM	0.00008
SUMMARY I-DDTS	I-DDTS	0.00000
SUMMARY I-DDC2C	I-DDC2C	0.11119
SUMMARY I-DDUP	I-DDUP	0.44711
SUMMARY EXTERNAL	E-DD	144.56966
SUMMARY I-C2SR	I-DDSR	144.56966
Assumptions:		
System Users providing data:	1,000,000	
Message/hour/System User:	100	
Subscribers:	15	
Aggregation Factor:	100	

Information Object	Assigned Interface	Bandwidth Minimum (Mbits/sec)
SUMMARY I-DDCD	I-DDCD	0.00040
SUMMARY I-DDMM	I-DDMM	111.11222
SUMMARY I-DDSM	I-DDSM	0.00008
SUMMARY I-DDTS	I-DDTS	0.00000
SUMMARY I-DDC2C	I-DDC2C	0.11119
SUMMARY I-DDUP	I-DDUP	0.44711
SUMMARY EXTERNAL	E-DD	3444.56966
SUMMARY I-C2SR	I-DDSR	3444.56966
Assumptions:		
System Users providing data:	1,000,000	
Message/hour/System User:	100	
Subscribers:	15	
Aggregation Factor:	1	

For the first case, where aggregation is used, communications loads are significant on the external interface and the interface to the Misbehavior Management node.

The loading on the Misbehavior Management interface is strictly dependent on the amount of incoming data, all of which must be scanned for misbehavior. This is a significant but not insurmountable amount of data to move internally.

The loading on the external interface is primarily a function of three factors: the amount of incoming data, the number of subscribers, and what level of aggregation or sampling is used to collapse the data stream. Clearly if no sampling or aggregation is used, then a Core of the size analyzed here would have communications needs that make implementation challenging, effectively limiting the size of a Core by its Data Distribution communications needs.

Aggregation however, is compute-intensive. Assuming a compute-load of 10,000 operations/incoming message for the above scenario, then the compute-load on the Data Distribution node would be:

100,000,000 users * 100 messages/user-hour * 1 hour/3600 seconds * 10,000 operations = 278 million operations/second.

This is a significant load, but given existing computing technology, not unrealistic. Sampling is where one out of every *n* data samples is used, where *n* is the inverse of the sampling rate. Sampling will be less compute-intensive than aggregation. Of course nothing prevents both approaches from being used.

Figure 7-3 illustrates the Connectivity View from the perspective of the Data Distribution subsystem that was used in this analysis.

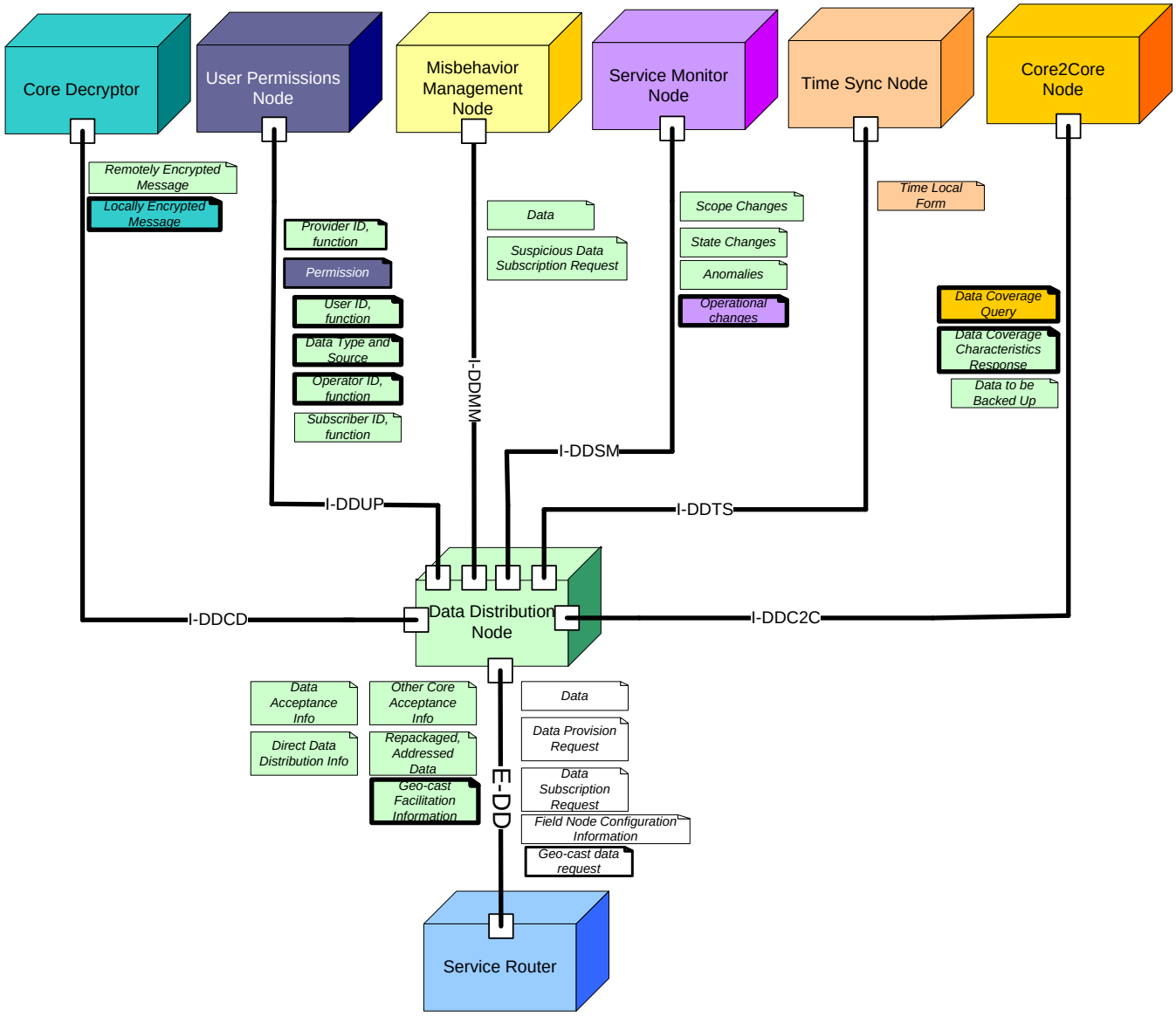


Figure 7-3: Analysis Connectivity View – Data Distribution

Table 7-5: Data Distribution Interface Analysis, 100M Mobile Users, 100x Aggregation

Information Object	Approximate Message Size (bytes)	Messages/ hour	# of Cores	Approximate Frequency (msg/sec)	Origin	Destination	Assigned Interface	Bandwidth Minimum (Mbits/sec)
Remotely Encrypted Message	1500	60	2	0	DD	CD	I-DDCD	0.00020
Locally Encrypted Message	1500	60	2	0	CD	DD	I-DDCD	0.00020
SUMMARY I-DDCD							I-DDCD	0.00040
Data	500	100000000	2	0	DD	MM	I-DDMM	111.11111
Suspicious Data Sub Request	5000	100	2	0	DD	MM	I-DDMM	0.00111
SUMMARY I-DDMM							I-DDMM	111.11222
Scope Changes	10000	1	2	0	DD	SM	I-DDSM	0.00002
State Changes	200	1	2	0	DD	SM	I-DDSM	0.00000
Anomalies	200	60	2	0	DD	SM	I-DDSM	0.00003
Operational Changes	3000	4	2	0	SM	DD	I-DDSM	0.00003
SUMMARY I-DDSM							I-DDSM	0.00008
Time Local Form	100	1	2	0	T	DD	I-DDTS	0.00000
SUMMARY I-DDTS							I-DDTS	0.00000
Data Coverage Query	5000	5	2		C2C	DD	I-DDC2C	0.00006
Data Cover Char Response	2000	5	2		DD	C2C	I-DDC2C	0.00002
Data to be Backed Up	50000000	1	2	0	DD	C2C	I-DDC2C	0.11111
SUMMARY I-DDC2C							I-DDC2C	0.11119
Subscriber ID, function	1000	100	2	0	DD	UP	I-DDUP	0.00022
Provider ID, function	1000	50000	2		DD	UP	I-DDUP	0.11111
Permission	1000	100600	2		UP	DD	I-DDUP	0.22356
User ID, function	1000	400	2		DD	UP	I-DDUP	0.00089
Data Type and Source	1000	50000	2		DD	UP	I-DDUP	0.11111
Operater ID, function	1000	100	2		DD	UP	I-DDUP	0.00022
SUMMARY I-DDUP							I-DDUP	0.44711
Data Acceptance Info	10000	4	2	0	DD	External	E-DD	0.00009
Direct Data Distribution Info	5000000	1	2	0	DD	External	E-DD	0.01111
Other Core Acceptance Info	100	60	2	0	DD	External	E-DD	0.00001
Repackaged, Addressed Data	1000	15000000	2	0	DD	External	E-DD	33.33333
Geo-Cast Facilitation Info	100	0.01	2	0	DD	External	E-DD	0.00000
Data	500	100000000	2	0	External	DD	E-DD	111.11111
Data Provision Request	1000	50000	2	0	External	DD	E-DD	0.11111
Data Subscription Request	1000	100	2	0	External	DD	E-DD	0.00022
Field Node Configuration Info	200000	1	2	0	External	DD	E-DD	0.00044
Geo-cast data Request	20000	50	2	0	External	DD	E-DD	0.00222
SUMMARY EXTERNAL							E-DD	144.56966
SUMMARY I-C2SR							I-DDSR	144.56966
Assumptions:								
System Users providing data:							1,000,000	
Message/hour/System User:							100	
Subscribers:							15	
Aggregation Factor:							100	

Table 7-6: Data Distribution Interface Analysis, 100M Mobile Users, No Aggregation

Information Object	Approximate Message Size (bytes)	Messages/ hour	# of Cores	Approximate Frequency (msg/sec)	Origin	Destination	Assigned Interface	Bandwidth Minimum (Mbits/sec)
Remotely Encrypted Message	1500	60	2	0	DD	CD	I-DDCD	0.00020
Locally Encrypted Message	1500	60	2	0	CD	DD	I-DDCD	0.00020
SUMMARY I-DDCD							I-DDCD	0.00040
Data	500	100000000	2	0	DD	MM	I-DDMM	111.11111
Suspicious Data Sub Request	5000	100	2	0	DD	MM	I-DDMM	0.00111
SUMMARY I-DDMM							I-DDMM	111.11222
Scope Changes	10000	1	2	0	DD	SM	I-DDSM	0.00002
State Changes	200	1	2	0	DD	SM	I-DDSM	0.00000
Anomalies	200	60	2	0	DD	SM	I-DDSM	0.00003
Operational Changes	3000	4	2	0	SM	DD	I-DDSM	0.00003
SUMMARY I-DDSM							I-DDSM	0.00008
Time Local Form	100	1	2	0	T	DD	I-DDTS	0.00000
SUMMARY I-DDTS							I-DDTS	0.00000
Data Coverage Query	5000	5	2		C2C	DD	I-DDC2C	0.00006
Data Cover Char Response	2000	5	2		DD	C2C	I-DDC2C	0.00002
Data to be Backed Up	50000000	1	2	0	DD	C2C	I-DDC2C	0.11111
SUMMARY I-DDC2C							I-DDC2C	0.11119
Subscriber ID, function	1000	100	2	0	DD	UP	I-DDUP	0.00022
Provider ID, function	1000	50000	2		DD	UP	I-DDUP	0.11111
Permission	1000	100600	2		UP	DD	I-DDUP	0.22356
User ID, function	1000	400	2		DD	UP	I-DDUP	0.00089
Data Type and Source	1000	50000	2		DD	UP	I-DDUP	0.11111
Operater ID, function	1000	100	2		DD	UP	I-DDUP	0.00022
SUMMARY I-DDUP							I-DDUP	0.44711
Data Acceptance Info	10000	4	2	0	DD	External	E-DD	0.00009
Direct Data Distribution Info	5000000	1	2	0	DD	External	E-DD	0.01111
Other Core Acceptance Info	100	60	2	0	DD	External	E-DD	0.00001
Repackaged, Addressed Data	1000	1.5E+09	2	0	DD	External	E-DD	3333.33333
Geo-Cast Facilitation Info	100	0.01	2	0	DD	External	E-DD	0.00000
Data	500	100000000	2	0	External	DD	E-DD	111.11111
Data Provision Request	1000	50000	2	0	External	DD	E-DD	0.11111
Data Subscription Request	1000	100	2	0	External	DD	E-DD	0.00022
Field Node Configuration Info	200000	1	2	0	External	DD	E-DD	0.00044
Geo-cast data Request	20000	50	2	0	External	DD	E-DD	0.00222
SUMMARY EXTERNAL							E-DD	3444.56966
SUMMARY I-C2SR							I-DDSR	3444.56966
Assumptions:								
System Users providing data:							1,000,000	
Message/hour/System User:							100	
Subscribers:							15	
Aggregation Factor:							1	

7.2.3 Connectivity View – User Trust Management Analysis

7.2.3.1 Introduction

This view illustrates User Trust Management internal and external interfaces. This View assumes that each subsystem is placed on a separate Node. It acknowledges the Core Decryptor as a separate node from User Trust Management. It assumes there is no Network Services Node, but there is a separate Service Router Node that implements the Network Services Engineering Objects.

7.2.3.2 Object Definitions and Roles

Nodes

Core Decryptor: This node operates the Core's decryption and local encryption functions.
Core2Core Node: This node operates Core2Core SEOs that implement all Core2Core functions. This could include multiple sub-nodes each running a portion of Core2Core software.
Misbehavior Management Node: This node operates Misbehavior Management SEOs. This could include multiple sub-nodes each running a portion of Misbehavior Management software.
Service Monitor Node: This node operates Service Monitor SEOs. This could include multiple sub-nodes each running a portion of Service Monitor software.
Service Router Node: This node operates Network Services Engineering Objects. It provides the interface between the User Trust Management Node and System Users.
Time Sync Node: This node operates Time Sync SEOs.
User Permissions Node: This node operates User Permissions SEOs. This could include multiple sub-nodes each running a portion of User Permissions software.
User Trust Management Node: This node operates User Trust Management SEOs. This could include multiple sub-nodes each running a portion of User Trust Management software.

Links

I-UTMCD: The internal interface between User Trust Management and Core Decryptor Nodes.
I-UTMMM: The internal interface between User Trust Management and Misbehavior Management Nodes.
I-UTMSM: The internal interface between User Trust Management and Service Monitor Nodes.
I-UTMTS: The internal interface between User Trust Management and Time Synchronization Nodes.
I-UTMSR: The internal interface between User Trust Management and Service Router (Network Service) Nodes.
I-UTMC2C: The internal interface between User Trust Management and Core2Core Nodes.
I-UTMUP: The internal interface between User Trust Management and User Permission Nodes.
E-UTM: The external interface between User Trust Management and the Internet. This is physically equivalent to I-UTMSR.

7.2.3.3 View Description

Table 7-7 summarizes two samples of the communications analysis for the User Trust Management subsystem. For planning purposes this analysis assumes similar deployment conditions to the Core2Core analysis: 100 million DSRC-equipped Mobile Users, and two scenarios, 5 Core Systems or 100 Core Systems. Network loads are more significant with more Cores, but at less than 20 Mbps still achievable with current technology.

Figure 7-4: Analysis Connectivity View – User Trust Management illustrates the Connectivity View from the perspective of the User Trust Management subsystem that was used in this analysis.

Table 7-7: User Trust Management Interface Analysis Summary

Information Object	Assigned Interface	Bandwidth Minimum (Mbits/sec)
SUMMARY I-UTMCD	I-UTMCD	5.55556
SUMMARY I-UTMMM	I-UTMMM	0.00093
SUMMARY I-UTMSM	I-UTMSM	0.00008
SUMMARY I-UTMTS	I-UTMTS	0.00000
SUMMARY I-UTMC2C	I-UTMC2C	9.64859
SUMMARY I-UTMUP	I-UTMUP	1.29630
SUMMARY EXTERNAL	E-UTM	3.96157
SUMMARY I-C2SR	I-UTMSR	3.96157
Assumptions:		
#DSRC Units		100,000,000
Interactions/day	1.00000	Interactions/day
Identity Cert Rate	1%	Identity Cert Rate
# Cores:	5	
Error Sig Rate	0.10%	Error Sig Rate

Information Object	Assigned Interface	Bandwidth Minimum (Mbits/sec)
SUMMARY I-UTMCD	I-UTMCD	0.27778
SUMMARY I-UTMMM	I-UTMMM	0.00005
SUMMARY I-UTMSM	I-UTMSM	0.00008
SUMMARY I-UTMTS	I-UTMTS	0.00000
SUMMARY I-UTMC2C	I-UTMC2C	18.64685
SUMMARY I-UTMUP	I-UTMUP	0.06481
SUMMARY EXTERNAL	E-UTM	0.45184
SUMMARY I-C2SR	I-UTMSR	0.45184
Assumptions:		
#DSRC Units		100,000,000
Interactions/day	1.00000	Interactions/day
Identity Cert Rate	1%	Identity Cert Rate
# Cores:	100	
Error Sig Rate	0.10%	Error Sig Rate

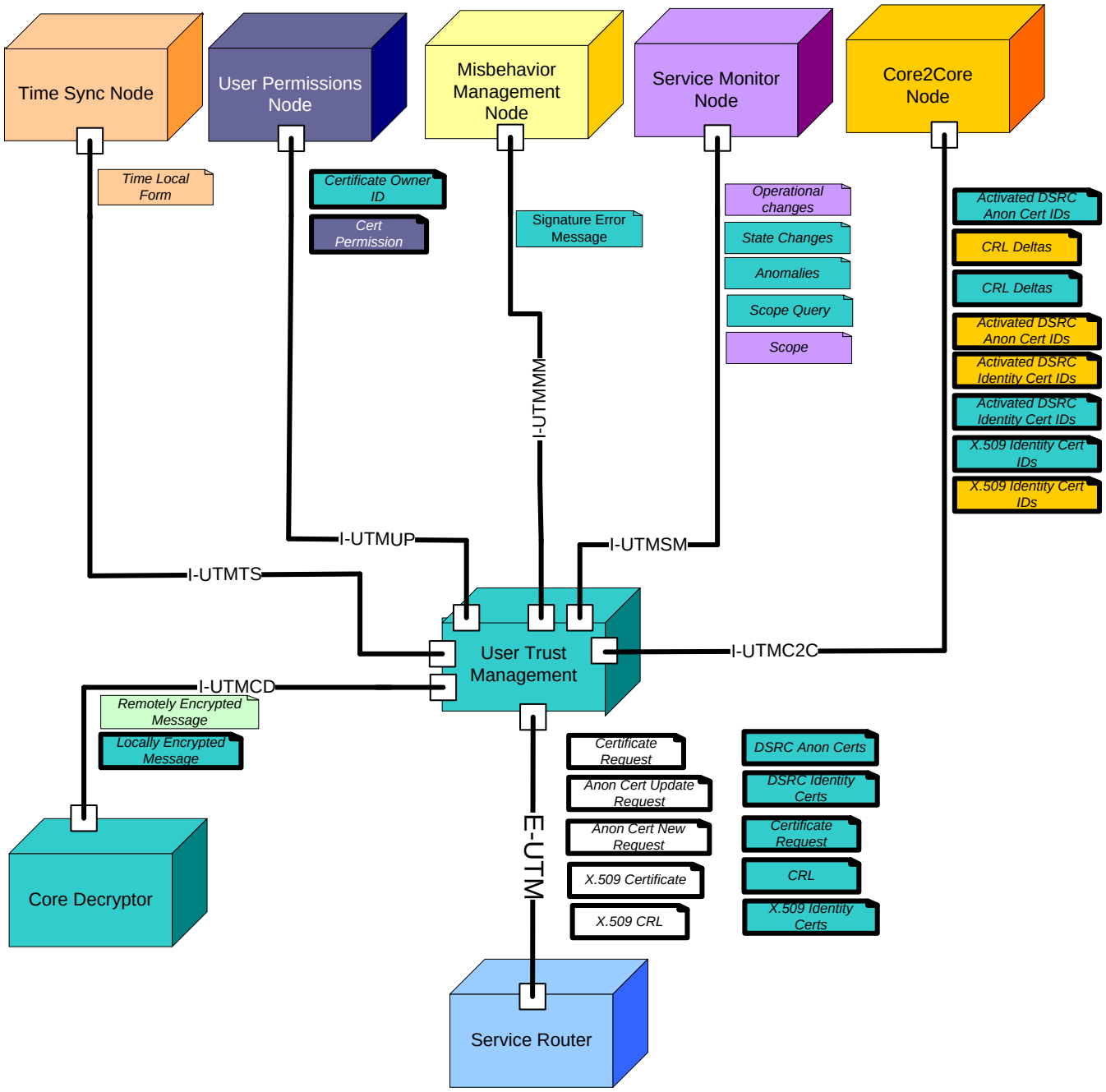


Figure 7-4: Analysis Connectivity View – User Trust Management

Table 7-8: User Trust Management Interface Analysis, 100M Mobile Users, 100 Cores

Information Object	Approximate Message Size (bytes)	Messages/ hour	# of Cores	Origin	Destination	Assigned Interface	Bandwidth Minimum (Mbits/sec)
Remotely Encrypted Message	1500	41,667	100	UTM	CD	I-UTMCD	0.13889
Locally Encrypted Message	1500	41,667	100	CD	UTM	I-UTMCD	0.13889
SUMMARY I-UTMCD						I-UTMCD	0.27778
Signature Error Message	500	42	100	UTM	MM	I-UTMMM	0.00005
SUMMARY I-UTMMM						I-UTMMM	0.00005
Scope Changes	10000	1	100	SM	UTM	I-UTMSM	0.00002
State Changes	200	1	100	UTM	SM	I-UTMSM	0.00000
Anomalies	200	60	100	UTM	SM	I-UTMSM	0.00003
Scope Query	200	4	2	UTM	SM	I-UTMSM	0.00000
Operational Changes	3000	4	100	SM	UTM	I-UTMSM	0.00003
SUMMARY I-UTMSM						I-UTMSM	0.00008
Time Local Form	100	1	100	TS	UTM	I-UTMTS	0.00000
SUMMARY I-UTMTS						I-UTMTS	0.00000
Activated DSRC Anon Cert IDs (in)	1000	41,667	100	C2C	UTM	I-UTMC2C	9.16667
CRL Deltas (in)	10000	99	100	C2C	UTM	I-UTMC2C	0.21780
CRL Deltas (out)	10000	1	100	UTM	C2C	I-UTMC2C	0.00220
Activated DSRC Anon Cert IDs (out)	1000	41,250	100	UTM	C2C	I-UTMC2C	0.09167
Activated DSRC Identity Cert IDs (in)	1000	41,250	100	C2C	UTM	I-UTMC2C	9.07500
Activated DSRC Identity Cert IDs (out	1000	417	100	UTM	C2C	I-UTMC2C	0.00093
X.509 Identity Cert IDs (in)	1000	41,250	100	C2C	UTM	I-UTMC2C	0.09167
X.509 Identity Cert IDs (out)	1000	417	100	UTM	C2C	I-UTMC2C	0.00093
SUMMARY I-UTMC2C						I-UTMC2C	18.64685
Cert Owner ID	200	41,667	100	UTM	UP	I-UTMUP	0.01852
Cert Permission	500	41,667	100	UP	UTM	I-UTMUP	0.04630
SUMMARY I-UTMUP						I-UTMUP	0.06481
DSRC Anon Certs	1000	41,250	100	UTM	External	E-UTM	0.09167
DSRC Identity Certs	1000	417	100	UTM	External	E-UTM	0.00093
Certificate Request	5000	1	100	UTM	External	E-UTM	0.00001
CRL	5000000	24	100	UTM	External	E-UTM	0.26667
X.509 Identity Certs	1000	417	100	UTM	External	E-UTM	0.00093
Certificate Request	1000	417	100	External	UTM	E-UTM	0.00093
Anon Cert Update Request	1000	39,583	100	External	UTM	E-UTM	0.08796
Anon Cert New Request	500	2,083	100	External	UTM	E-UTM	0.00231
X.509 Certificates	200	1	100	External	UTM	E-UTM	0.00000
X.509 CRL	50000	4	100	External	UTM	E-UTM	0.00044
SUMMARY EXTERNAL						E-UTM	0.45184
SUMMARY I-C2SR						I-UTMSR	0.45184
Assumptions:							
#DSRC Units							100,000,000
Interactions/day						1.00000	Interactions/day
Identity Cert Rate						1%	Identity Cert Rate
# Cores:						100	
Error Sig Rate						0.10%	Error Sig Rate

8.0 GLOSSARY AND ACRONYMS

Table 8-1: Glossary of Terms

Term	Definition
Access Control	Refers to mechanisms and policies that restrict access to computer resources. An access control list (ACL) for example, specifies what operations different users can perform on specific files and directories.
Aggregate	To combine data elements of similar format into a single data element that is a statistical representation of the original elements.
Analysis	The process of studying a system by partitioning the system into parts (functions, components, or objects) and determining how the parts relate to each other.
Anonymity	Lacking individuality, distinction, and recognizability within message exchanges.
Application	One or more pieces of software designed to perform some specific function; it is a configuration of interacting Engineering Objects. A computer software program with an interface, enabling people to use a computer as a tool to accomplish a specific task.
Authentication	The process of determining the identity of a user that is attempting to access a network.
Authorization	The process of determining what types of activities or access are permitted on a network. Usually used in the context of authentication: once you have authenticated a user, they may be authorized to have access to a specific service.
Assumption	A judgment about unknown factors and the future which is made in analyzing alternative courses of action.
Back Office	See Center
Bad Actor	A role played by a user or another system that provides false or misleading data, operates in such a fashion as to impede other users, operates outside of its authorized scope.
Center	An entity that provides application, management, administrative, and support functions from a fixed location not in proximity to the road network. The terms “back office” and “center” are used interchangeably. Center is a traditionally a transportation-focused term, evoking management centers to support transportation needs, while back office generally refers to commercial applications. From the perspective of the Core System these are considered the same.
Concept of Operations (ConOps)	A user-oriented document that describes a system’s operational characteristics from the end user’s viewpoint.
Constraint	An externally imposed limitation on system requirements, design, implementation or on the process used to develop or modify a system. A constraint is a factor that lies outside – but has a direct impact on – a system design effort. Constraints may relate to laws and regulations or technological, socio-political, financial, or operational factors.

Term	Definition
Contract	In project management, a legally binding document agreed upon by the customer and the hardware or software developer or supplier; includes the technical, organizational, cost, and/or scheduling requirements of a project.
Coordinated Universal Time (UTC)	The primary time standard by which the world regulates clocks and time. UTC serves to accommodate the timekeeping differences that arise between atomic time (which is derived from atomic clocks) and solar time (which is derived from astronomical measurements of the Earth's rotation on its axis relative to the Sun). Since Jan. 1, 1972, UTC has been modified by adding "leap seconds" when necessary.
Cross Support	An agreement between two or more organizations to exploit the technical capability of interoperability for mutual advantage, such as one organization offering support services to another in order to enhance or enable some aspect of a mission.
Data Acceptance Criteria	Criteria describing the data a Core System accepts as part of Data Distribution. Includes data type, source type, location and time criteria.
Data Consumer	A user or system that is receiving or using data from another user or system.
Data Provider	A user or system that is supplying or transmitting data to another user or system.
Deployability	Able to be deployed in existing roadway environments, without requiring replacement of existing systems in order to provide measurable improvements.
Digital Certificates	A digital certificate is an electronic "identification card" that establishes your credentials when doing business or other transactions on the Web. It is issued by a certification authority. It contains your name, a serial number, expiration dates, a copy of the certificate holder's public key (used for encrypting messages and digital signatures), and the digital signature of the certificate-issuing authority so that a recipient can verify that the certificate is real. Note: From the SysAdmin, Audit, Network, Security Institute – www.sans.org website.
Domain	An Enterprise Object that is under single organizational, administrative or technical control
Element	(1) A constituent part of something; (2) any thing that is one of the individual parts of which a composite entity is made up; (3) an identifiable component, process or entity of a system.
Encryption	Scrambling data in such a way that it can only be unscrambled through the application of the correct cryptographic key.
End-User	The ultimate user of a product or service, especially of a computer system, application, or network.
Engineering Object	An implementation or realization of some abstract function. It may be implemented as hardware (Node) or as software (application or software component).
Enterprise Object	An entity that is governed by a single authority that has its own objectives and policies for operating the object. An Enterprise Object may be a component of another larger Enterprise Object

Term	Definition
Entity	A physical or abstract thing of interest.
Environment	The circumstances, objects, and conditions that surround a system to be built; includes technical, political, commercial, cultural, organizational, and physical influences as well as standards and policies that govern what a system must do or how it will do it.
Extensibility	The ability to add or modify functionality or features with little or no design changes.
External Support System	An entity that provides a service the Core needs to deliver. This service is provided by the ESS because it makes more sense to manage, maintain and share the service between multiple Cores due to overriding institutional, performance or functional constraints.
Facility	A physical infrastructure element that supports the use of services and other resources.
Federation	A group of domains that coordinate to share resources while each domain retains its authority over its own resources. Federations are governed by negotiated agreements.
Flexibility	The ability to adjust or adapt to external changes with little or no design changes.
Functional Object	An abstract model of a functional entity that receives requests, performs actions, and generates or processes data.
Functionality	The capabilities of the various computational, user interface, input, output, data management, and other features provided by a product.
Geo-cast	The delivery of a message to a group of network destinations identified by their geographical locations.
Hardware	Hardware refers to the physical parts of a computer and related devices. Internal hardware devices include motherboards, hard drives, and memory. External hardware devices include monitors, keyboards, mice, printers, and scanners.
Information Object	Description of data along with the necessary structure and syntax to allow interpretation and use of these Objects.
Information Package	An Information Object with associated Metadata necessary to use the Information Object.
Jurisdictional Scope	The power, right, or authority to interpret and apply the law within the limits or territory which authority may be exercised.
Link	The locus of relations among Nodes. It provides interconnections between Nodes for communication and coordination. It may be implemented by a wired connection or with some radio frequency (RF) or optical communications media. Links implement the primary function of transporting data. Links connect to Nodes at a Port.
Maintainability	To keep in an existing operational state preserved from failure or decline of services (with minimum repair, efficiency, or validity).

Term	Definition
Metadata	'Data about data', the information that describes content. It is information about the meaning of data, as well as the relationships among Information Objects, rules for their use and transformation, and policies on access.
Node	A physical hardware Engineering Object that is a run-time computational resource and generally has at least memory and processing capability. Run-time software Engineering Objects reside on nodes. A Node has some well-understood, possibly rapidly moving, location. A Node may be composed of two or more (sub) Nodes.
Object	An abstract model of an entity in the real world. It contains information, has behavior and may offer services. An object is characterized by that which makes it distinct from other Objects.
Ownership	Having administrative and fiscal responsibility for the owned element and the right to exclusively control and use it for one's own purposes. It is the state or fact of having exclusive possession or control of some object, facility, intellectual property or some other kind of property
Parse	To extract individual elements from a larger message.
Permission	Authorization granted to do something. To the Core System, permissions are granted to System Users and Operators determining what actions they are allowed to take when interacting with the Core.
Policy	A set of guidelines and constraints on the behaviors and states exhibited by the objects in the Core System.
Port	The physical element of a Node where a Link is connected. Nodes may have one or more Ports.
Priority	A rank order of status, activities, or tasks. Priority is particularly important when resources are limited.
Protocol Entity	An object that performs actions to exchange or transfer data (as distinguished from a Functional Object that generates or processes data). Protocol Entities are used to support interactions between two Engineering Objects or among groups of Engineering Objects that are contained in separate Nodes (see the Connectivity Viewpoint for discussion of Engineering Objects and Nodes).
Privacy	From the VII Privacy Policies Framework: the respect for individual choices about, and control over an individual's personal information.
Problem domain	A set of similar problems that occur in an environment and lend themselves to common solutions.
Pseudo-ID	An identifier used by an individual that is not associated with the individual's identity, and may change to prevent such association.
Reliability	Providing consistent and dependable system output or results.
Resource	Anything available to a system that can support the achievement of objectives; i.e., any physical or virtual element that may be of limited availability within a system.

Term	Definition
Role	The way in which an object participates in a relationship; an object's set of behaviors and actions associated with the relationship of that object with other objects.
Sample	To select a subset of a larger group, usually in a regular pattern (e.g.., 1 of every 10).
Scalability	The capable of being easily grown, expanded or upgraded upon demand without requiring a redesign.
Service	A provision of an interface of an object to support actions of another object.
Software	Software is a general term that describes computer programs. Terms such as software programs, applications, scripts, and instruction sets all fall under the category of computer software.
Special Permissions	Authorization granted to perform actions specific to 3rd party applications, using IEEE 1609.2 certificates as the permission grant mechanism. Also called certificate-managed application permissions.
Standard	A formal specification that defines and governs functions and protocols at interfaces of a data system. It describes in detail the capabilities and establishes the requirements to be met by interfacing systems to achieve compatibility.
States or Modes	A distinct system setting in which the same user input will produce different results than it would in other settings. The Core System as a whole is always in one state. Individual subsystems may be in different modes.
Stratum-2	Time servers that are 2 steps removed from a basic time source (e.g., atomic clock). Stratum-2 time servers request time from Stratum-1 servers. Stratum-1 time servers are directly connected to a basic time source by a hardwire connection (e.g., RS-232).
System	(A) A collection of interacting components organized to accomplish a specified function or set of functions within a specified environment. (B) A group of people, objects, and procedures constituted to achieve defined objectives of some operational role by performing specified functions. A complete system includes all of the associated equipment, facilities, material, computer programs, firmware, technical documentation, services, and personnel required for operations and support to the degree necessary for self-sufficient use in its intended environment.
Traceability	The identification and documentation of derivation paths (upward) and allocation or flow down paths (downward) of work products in the work product hierarchy. Important kinds of traceability include: to or from external sources to or from system requirements; to or from system requirements to or from lowest level requirements; to or from requirements to or from design; to or from design to or from implementation; to or from implementation to test; and to or from requirements to test.
Type-ID	An identifier associated with a class of individuals.

Term	Definition
User	An entity that uses a computer, program, network, and related services of a hardware and/or software system. In the case of the Core System this includes System Users that refers to the combination of Mobile, Field, and Center based devices and applications. The term End User refers to the human user of the System User device. End Users do not interact directly with the Core System, but are referred to as the ultimate beneficiaries or participants in the *connected vehicle* environment.
User-ID	An identifier associated with a specific individual that can be traced to that individual's identity.

Table 8-2: Acronyms and Abbreviations

Abbreviation/ Acronym	Definition
AASHTO	American Association of State Highway and Transportation Officials
ACL	Access Control List
C2C	Core2Core
CA	Certificate Authority
CAMP	Crash Avoidance Metrics Partnership
CCSDS	Consultative Committee for Space Data Systems
ConOps	Concept of Operations
COTS	Commercial off-the-shelf
CRL	Certification Revocation List
DD	Data Distribution
DOT	Department of Transportation
DSRC	Dedicated Short Range Communication
DMS	Dynamic Message Sign
DTLS	Datagram Transport Layer Security
ESS	External Support System
FCC	Federal Communications Commission
FHWA	Federal Highway Administration
FTA	Federal Transit Administration
Gbps	Gigabits per second
HEO	Hardware Engineering Object
ID	Identifier
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMT-A	International Mobile Telecommunications Advanced
IP	Internet Protocol
IPsec	Internet Protocol Security
ISO	International Organization for Standardization
IPS	Intrusion Prevention System
ITS	Intelligent Transportation Systems
JPO	Joint Program Office
Kbps	Kilobits per second
LA	Linkage Authority
LAN	Local Area Network
LLC	Logical Link Control
LTE	Long Term Evolution

Abbreviation/ Acronym	Definition
MAC	Medium Access Control
Mbps	Megabits per second
MM	Misbehavior Management
OSI	Open Systems Interconnect
NHTSA	National Highway Traffic Safety Administration
NS	Network Services
OMG	Object Management Group
PII	Personally Identifiable Information
PKI	Public Key Infrastructure
PSID	Provider Service Identifiers
RA	Registration Authority
RASDS	Reference Architecture for Space Data Systems
RITA	Research and Innovative Technology Administration
RF	Radio frequency
RM-ODP	Reference Model of Open Distributed Processing
RSE	Roadside Equipment
SAD	System Architecture Document
SAE	Society of Automobile Engineers
SCN	Service Component Node
SE	Systems Engineering
SEO	Software Engineering Object
SM	Service Monitor
SUID	System User Identifier
SyRS	System Requirements Specification
SysML	Systems Modeling Language
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TS	Time Synchronization
TIA	Telecommunications Industry Association
UDP	User Datagram Protocol
UP	User Permissions
USDOT	US Department of Transportation
UTC	Coordinated Universal Time
UTM	User Trust Management
VII	Vehicle Infrastructure Integration
VPN	Virtual Private Network
WAN	Wide Area Network

Abbreviation/ Acronym	Definition
WAVE	Wireless Access in Vehicular Environments
WiMAX	Worldwide Interoperability for Microwave Access
WSM	WAVE Short Message
WSMP	WAVE Short Message Protocol