Continuous Asset Evaluation, Situational Awareness and Risk Scoring Framework Extension (CAESARS-FE)

An Enterprise Continuous Monitoring Technical Reference Model Overview

Peter Sell, National Security Agency

Acknowledgments

- Jointly developed by:
  - The U.S. National Security Agency (NSA)
  - The U.S. Department of Homeland Security (DHS)
  - The National Institute of Standards and Technology (NIST)
  - MITRE
  - Booz Allen Hamilton

Agenda

- Intent of Continuous Monitoring (ConMon)
- Goals of CAESARS FE Technical Reference Model
  - NIST Interagency Report (IR) 7756
- Recent Updates
- Limitations of the CAESARS Reference Architecture
- ConMon Technical Reference Model
- Subsystems
- Technical Challenges to be Addressed by a ConMon Technical Reference Model
- Solutions
- Summary

Intent of Continuous Monitoring (ConMon)

Overall Goals

- Provide a cross data domain view of information
- Provide situational awareness by presenting compliance and risk information
- Enable efficiencies in measurement using automation and standardized data feeds
- Support decision making at all levels of the enterprise

ConMon Data Domains

Source: NIST SP 800-137

How can I use this guidance???

- Agencies implementing ConMon *now*
  - Obtain high level design, workflows, and functional requirements that can guide custom implementation efforts
  - Utilize low level communication specifications together to design and develop standardized ConMon capabilities
- Agencies implementing ConMon *in the future*
  - Leverage the model to plan future ConMon design and procurements to enable federated, interoperable solutions (e.g., a government-wide capability)
- Vendors
  - Utilize specifications that enable the rapid and cost effective ConMon deployments
    - *gaining international consideration – IETF
- Commercial Sector
  - Adopt a standardized approach to data normalization and tool integration

Goals of the CAESARS-FE Technical Reference Model (NISTIR 7756)

- To provide a blueprint to guide ConMon procurement and implementation to a standards based solution
- To functionally decompose the technical aspects of ConMon into modular components
- Promote interoperability through the detailed definition of machine interfaces
  - e.g., Data formats, Communications flows, Transport/wire protocols
- Provide orchestration capabilities enabling coordination of ConMon activities across vendor and product lines
- Provide a standards-based foundation promoting future innovation – even internationally

Recent Updates

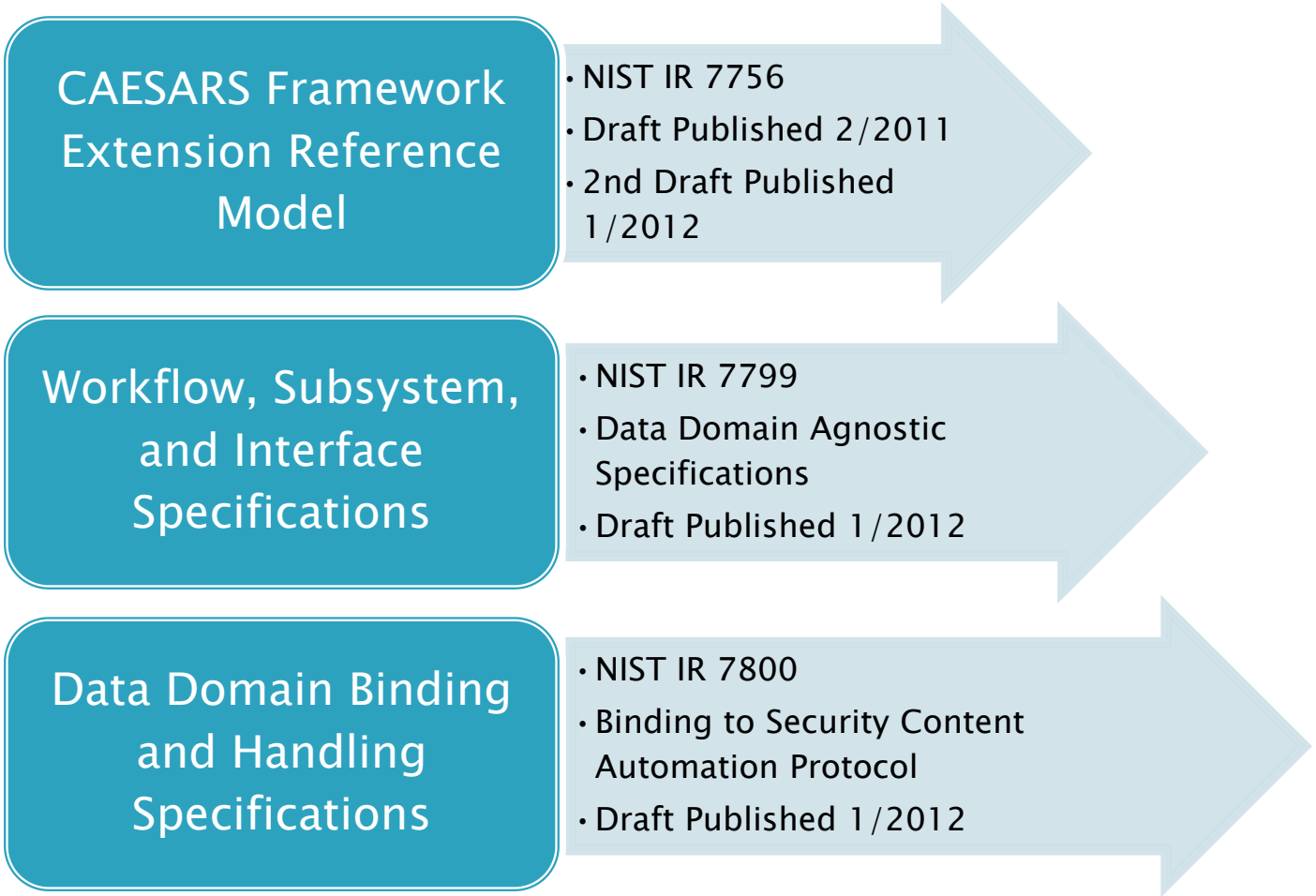
- Public Comment on the 2nd Draft of CAESARS FE led to some notable changes:
  - Additional limitation of CAESARS: Lack of Enforcement Capability
    - Added Enforcement Subsystem (notional)
    - Added Enforcement Controller Component (notional) as part of the Task Manager (TM) Subsystem
  - Within the Presentation/Reporting Subsystem
    - Renamed Dashboard Engine Component to Dashboard Component
    - Added Reporting Engine Component to the Presentation /Reporting Subsystem
  - Added additional connections:
    - TM and Content Subsystem
    - TM and Enforcement Subsystem

Limitations of the CAESARS model

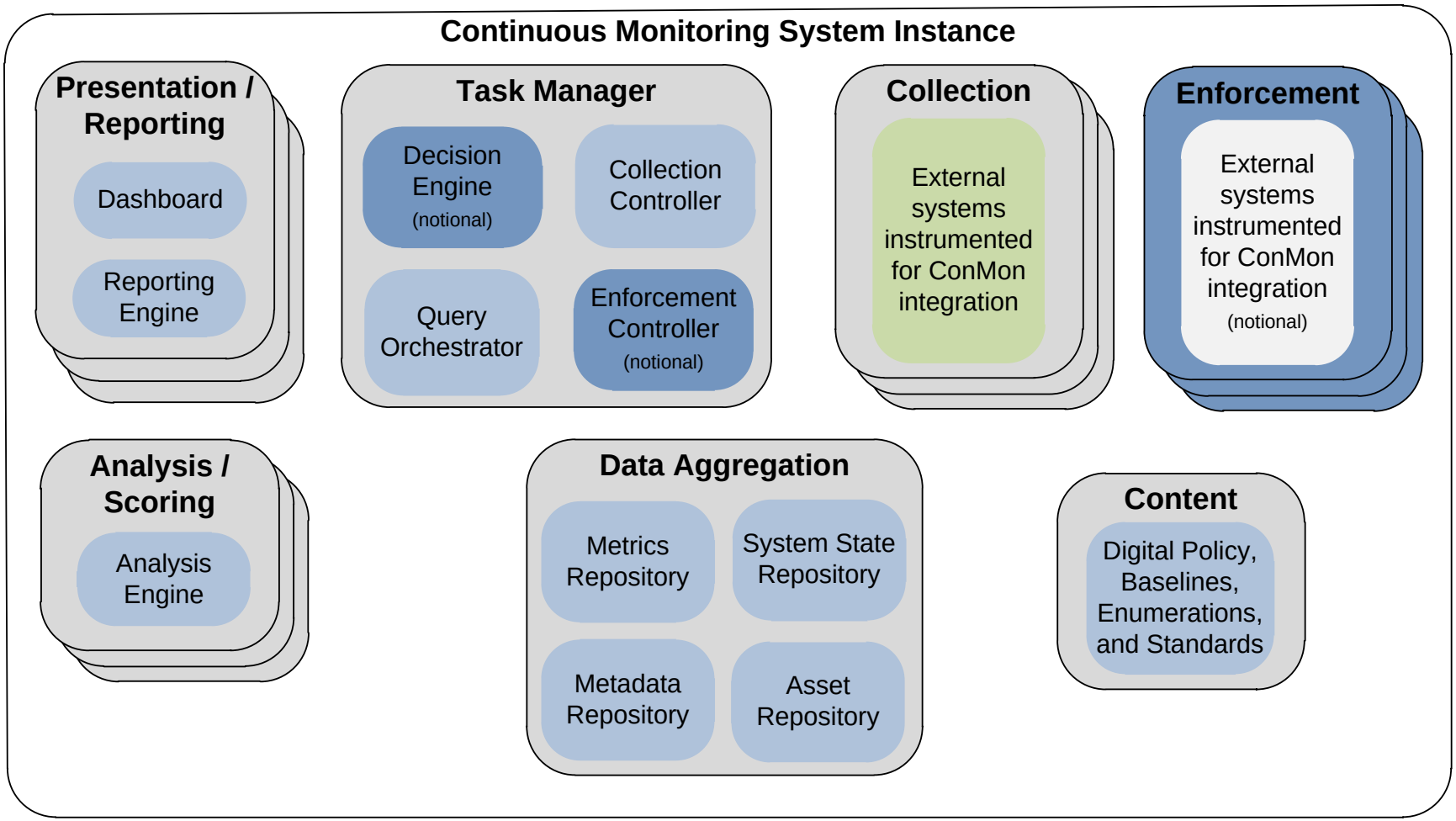
1. Lack of interface specifications
2. Reliance on an enterprise service bus
3. Incomplete communication payload specifications
4. Lack of specifications describing subsystem capabilities
5. Lack of a multi-ConMon instance capability
6. Lack of multi-subsystem instance capability
7. ConMon database integration with security baseline content
8. Lack of detail on the required asset inventory
9. Stringent requirements for risk measurement
10. Lack of Enforcement Capability

CAESARS is a good foundation. We needed to expand upon its framework to address its limitations and add additional capabilities.

ConMon Documentation Structure



ConMon Technical Reference Model
(Organizations may have multiple ConMon instances)



ConMon Reference Model Subsystems

- CAESARS FE contains seven distinct subsystems that together compose the ConMon Reference Model:
  1. **Presentation/Reporting**: takes user input, creates data queries, and renders available data as reports and visualizations.
  2. **Content**: stores digital policy and supporting data (e.g., for checking system states)
  3. **Collection**: detects system state information in accordance with organizational policy
  4. **Data Aggregation**: stores system state information, related calculated results, and associated metadata
  5. **Analysis/Scoring**: analyzes system state information and other data, generates measures and scores
  6. **Task Manager**: orchestrates the activities of the other subsystems and communicates with other ConMon instances in enabling fulfillment of user data queries
  7. **Enforcement (notional)**: enforces policy by affecting changes to the operational state of systems and by directing organization behavior (e.g., trouble ticketing) based on human decisions

Technical Challenges to be Addressed by a ConMon Technical Reference Model

These are areas that need to be addressed to achieve a usable the enterprise architecture, but for which commercial tools are often insufficient:

- Current ConMon implementations lack modularity
- No capability to orchestrate activity between ConMon instances and different tool sets
- Queries generated for ConMon systems are static, often using proprietary code
- Lack of coordination among multiple solutions across the Enterprise
- Lack of enforcement capabilities
- No standardized normalization of collected data
  - Specifically re: asset management
- Many ConMon solutions only collect results, not raw data
- No streamlined manner to customize analysis and scoring

Challenge #1: Lack of Modularity

Problem:

- Current ConMon implementations lack modularity
- Often require monolithic solutions

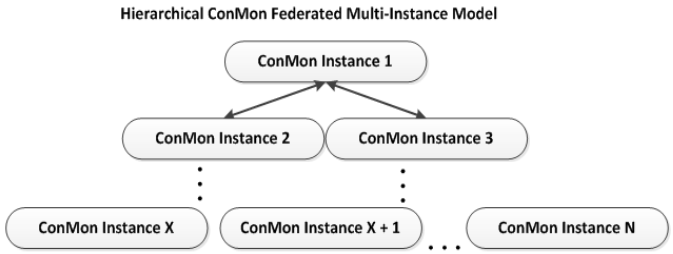
Our Solution:

- Use a component-based approach based on a functional decomposition of ConMon

Benefits:

- Facilitates Data Agnostic design (e.g., Content Repository, Task Manager)
- Enables organizations to select best of breed technologies for a specific function
- Multiple instances of presentation, collection, analysis, and enforcement, provided by various vendors, support different user roles

Challenge #2: Cross-Product/Instance Orchestration



Problem:

- Lack of capability to orchestrate activity between multiple ConMon instances and different tool sets

Our Solution:

- Define standardized interfaces enabling cross-product and inter-instance orchestration

Benefits:

- Greater interoperability and easier integration
- Provides a foundation that enables innovation

Challenge #3: Static Datasets

Problem:

- Datasets supported by ConMon systems are static, often supported using proprietary code
- Datasets, and thus queries, may not cross information domains

Our Solution:

- Define standardized methods to execute “named” static queries
- Provide a framework that supports dynamic queries that can cross information domains
- Provide hooks enabling queries to be reviewed and approved as needed

Benefits:

- Queries can be propagated to other ConMon instances, supporting automated data collection
- The model supports the operational need to query data that is outside of a predefined view, perhaps based on human interaction
- With proper moderation, dynamic queries do not result in system degradation

Challenge #4: Lack of a Tie to Enforcement Mechanisms

Problem:

- Monitoring supports human decision cycles; automating based on human decisions requires access to the same information

Our Solution:

- Reuse of monitoring interfaces to orchestrate and provide data to enforcement capabilities
- Enforcement subsystem supports different mechanisms to affect change based on organizational policies:
  - Remediation tools
  - Network Policy Enforcement (e.g. TNC, NAC)
  - Tie-in with trouble ticketing solutions and other human-oriented methods

Benefits:

- Describes a more comprehensive end-state that supports more than "read-only" data collection
- Monitoring and operations teams can utilize a common toolset

Challenge #5: Monitoring Data is not Normalized

Problem:

- Monitoring data is not expressed using standardized formats

Our Solution:

- Use of standardized asset data exchange models enables use of asset information from a variety of sources
- Collected data is represented using standardized data exchange specifications

Benefits:

- Greater interoperability
- Reuse of existing data sources

Challenge #6: Many Tools Only Collect Findings

Problem:

- Many ConMon solutions only collect compliance results, not raw data
- New data needs to be collected if the compliance rules change

Our Solution:

- When possible, collect raw data, not just results.
- Store raw data as close to the source as possible
  - Take advantage of distributed ConMon instances
  - Minimize network bandwidth
- Use "Big Data" analytical approaches for large data volumes

Benefits:

- Enables reuse of raw data and intermediate computations; "scan once, report many"
- Differentiates:
  - Raw Data – Actual system state, low-level data points
  - Findings – Boolean values, compliance results derived from raw data
  - Scoring – High-level measures and scores derived from findings

Challenge #7: Standardized Analytics

Problem:

- Little to no standardization for the orchestration and parameterization of analysis and scoring

Our Solution:

- Develop and reference standards for orchestration of analysis tasks
- Provide a framework for parameterizing analysis

Benefits:

- Enables customization of analysis and scoring based on current threats, weaknesses, and organization's requirements
- “Collect Once, Reuse Many”: The same collected data can be used by multiple analysis and scoring algorithms
- Reports can be tailored as per the audience
  - Executives
  - System administrators
  - Security analysts

The ConMon Reference Model Provides Tangible Guidance

- **Applicable to large enterprises**
  - Leverage the ConMon reference model to create multiple ConMon instances
  - Organize ConMon instances in a tiered, federated architecture.
- **Enable end-user organizations to implement ConMon more rapidly**
  - Leverage ConMon reference model compliant tools to compose enterprise ConMon capabilities without lengthy and costly custom integration efforts.
- **Provide standards to allow integration of subsystems – vendor solutions**
  - Leverage ConMon reference model interfaces, data normalization, and reports to integrate Federal- and agency-level ConMon data.
- **Leverage Federal buying power to reduce the cost of implementing ConMon**
  - ConMon reference model serves as a foundation for product procurement and testing. Without this, procurements may be non-interoperable and risk measurement results may be non-comparable.

Summary

Goals of the reference model is to enable organizations to:

- Collect and aggregate data from across a diverse set of security and systems management tools
- Analyze that data
- Perform scoring
- Facilitate user queries
- Provide overall situational awareness in support of risk-based decision making
- Provide a foundation to enable future automation in response to human decision making
  - Human directed
  - Automated digital policy

- QUESTIONS?

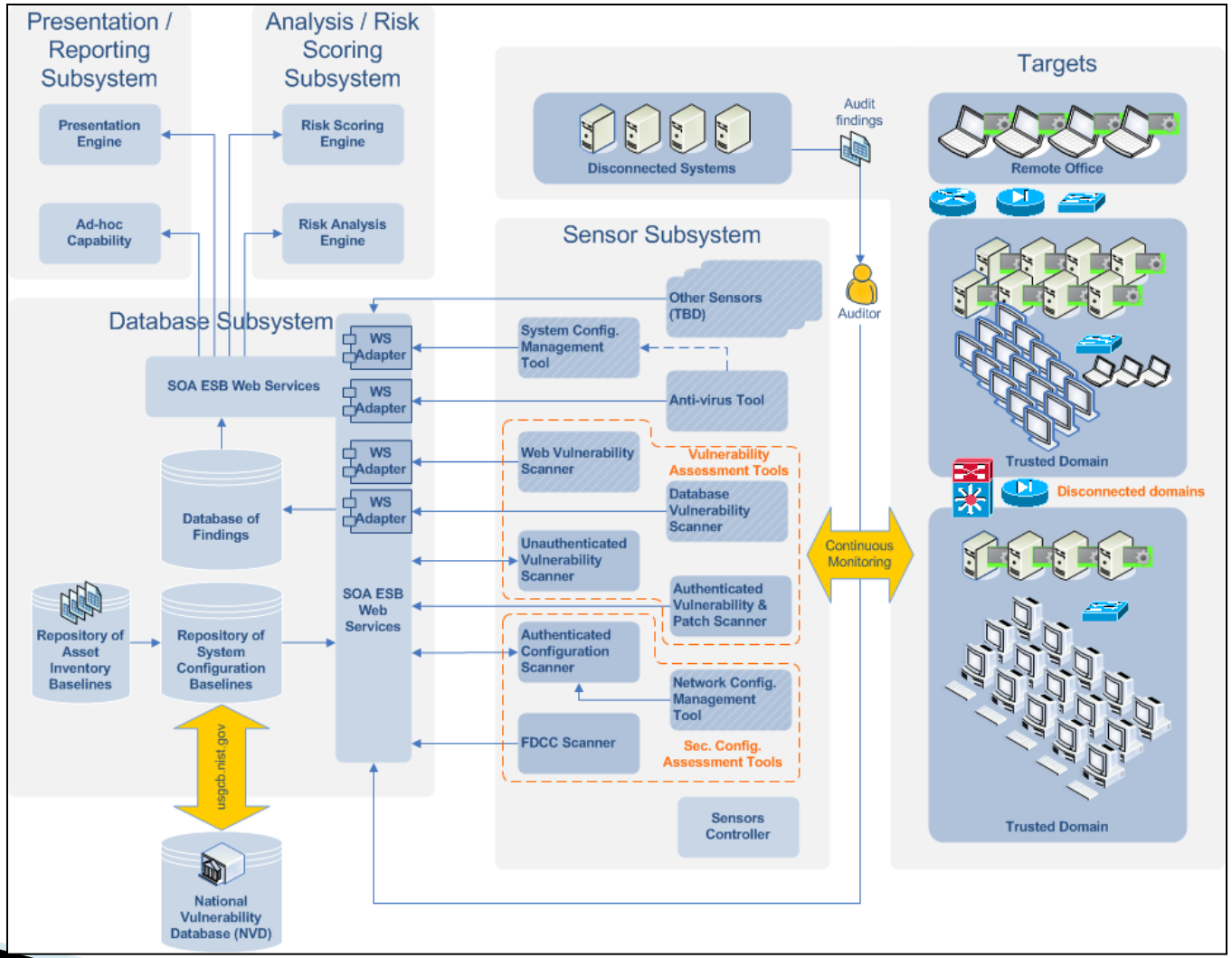
Contact Information:

Peter J. Sell
pjsell@missi.ncsc.mil

Dave Waltermire - NIST
david.waltermire@nist.gov

BACKUP

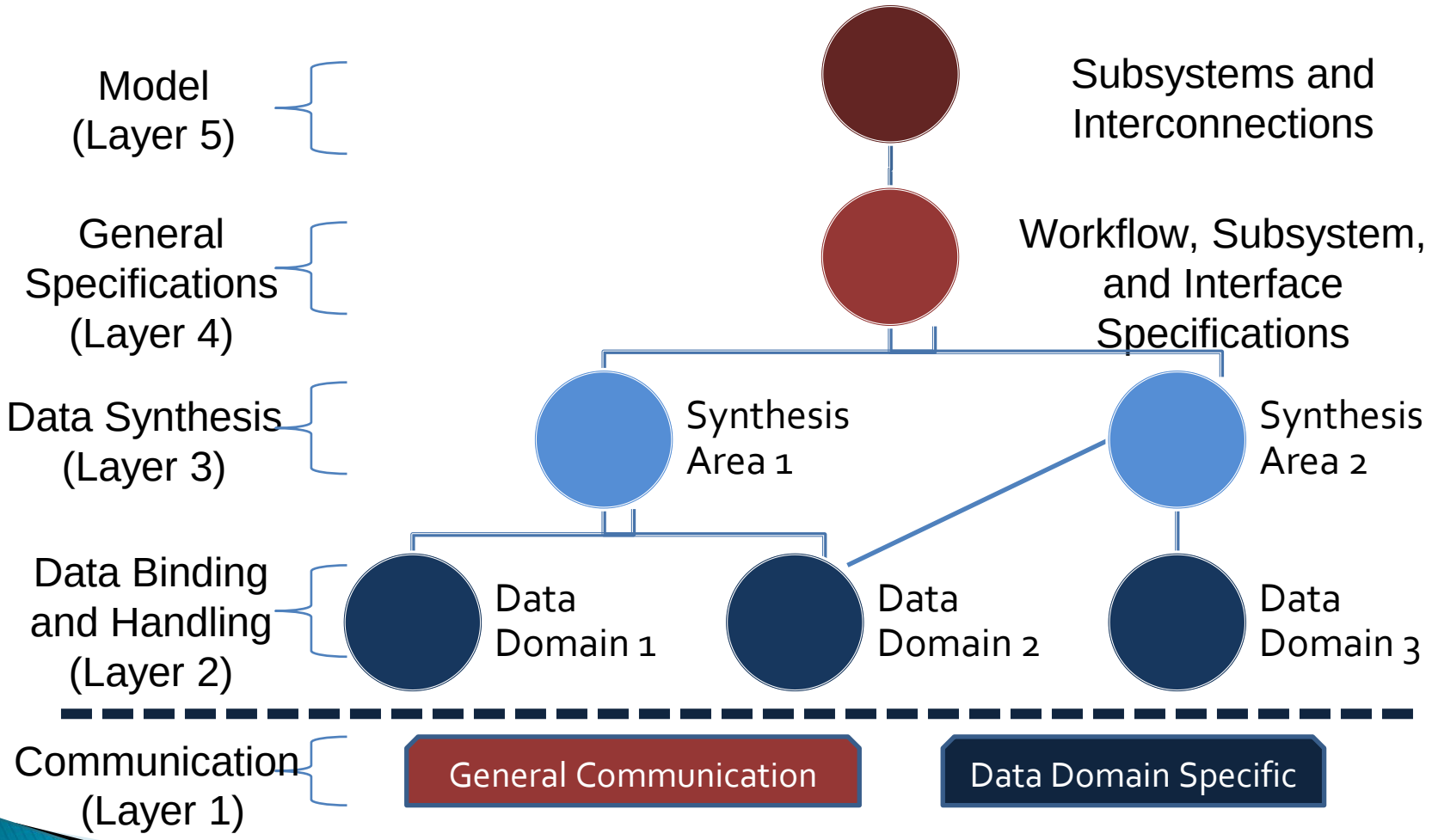
DHS Continuous Asset Evaluation, Situational Awareness, and Risk Scoring (CAESARS) Reference Architecture



Project Timeline

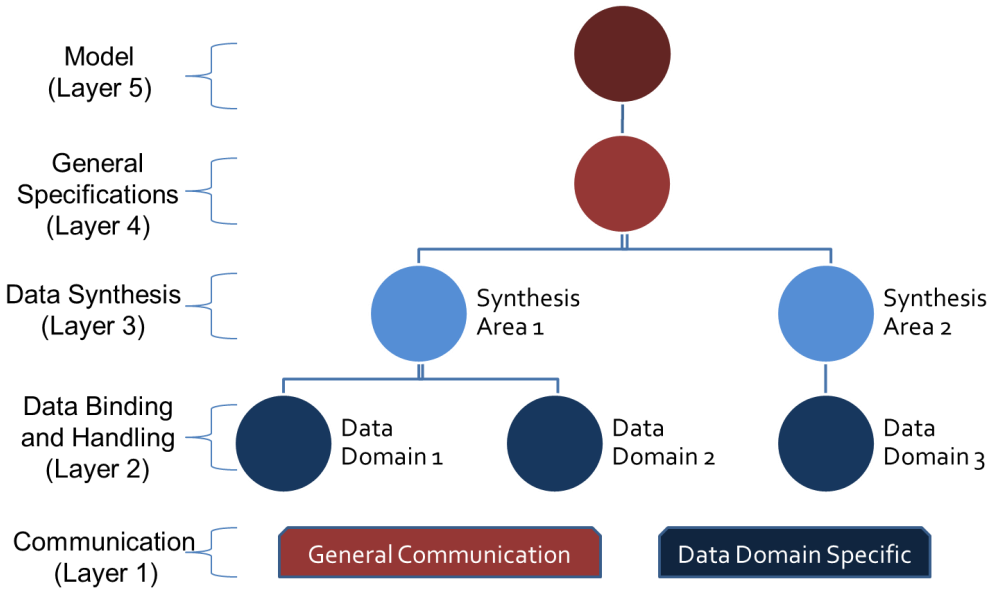
- 9/2010: DHS published CAESARS reference architecture
- **9/2010: ISIMC CMWG initiated DHS/NSA/NIST research initiative to create the CAESARS Framework Extension (FE)**
- **2/2011: NIST and DHS published CAESARS FE (draft NIST IR 7756)**
- 3/2011: ConMon modeling workshop at NIST March 21
- 11/2011: Presentation of model at the 7th Annual IT Security Automation Conference
- **1/2012: Public drafts of ConMon specifications**
- 7/2012: Security Automation Developer Days

ConMon Reference Model Specification Layers



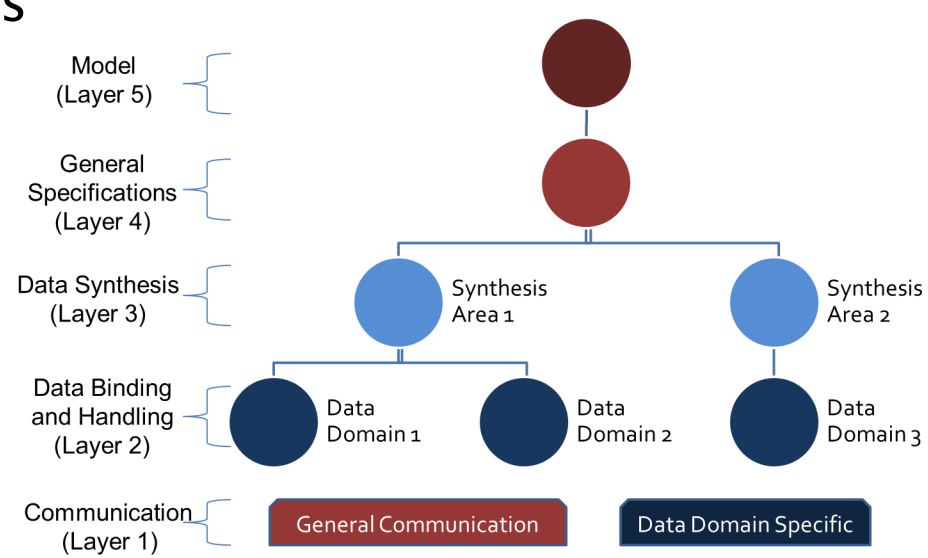
Layer 5: The Model

- Defined in NISTIR 7756
- Subsystems
  - Presentation/Reporting
  - Analysis/Scoring
  - Data Aggregation
  - Collection
  - Content
  - Task Management
  - Enforcement (notional)
- Subsystem Components
- Subsystem Interconnections
  - Describes required communication pathways



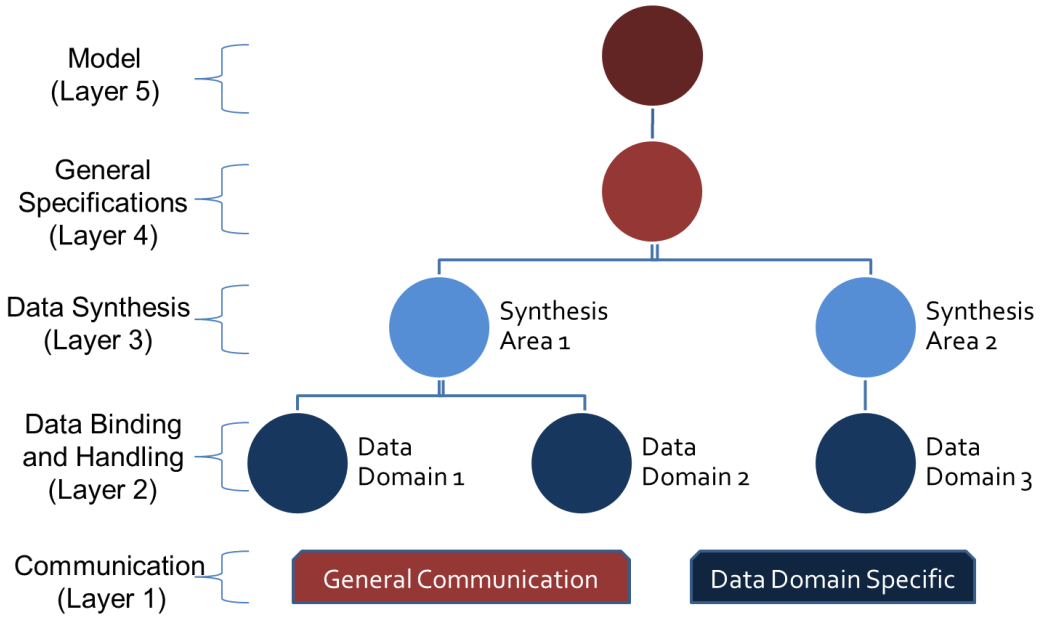
Layer 4: General Specifications

- Defined in NISTIR 7799
- Workflows
  - Data Acquisition and Analysis
  - Query Fulfillment
  - Digital Policy and Content Propagation
- Subsystem Specifications
- Interface Specifications
  - Result Reporting
  - Content Acquisition
  - Query and Tasking
  - Advanced Data Retrieval



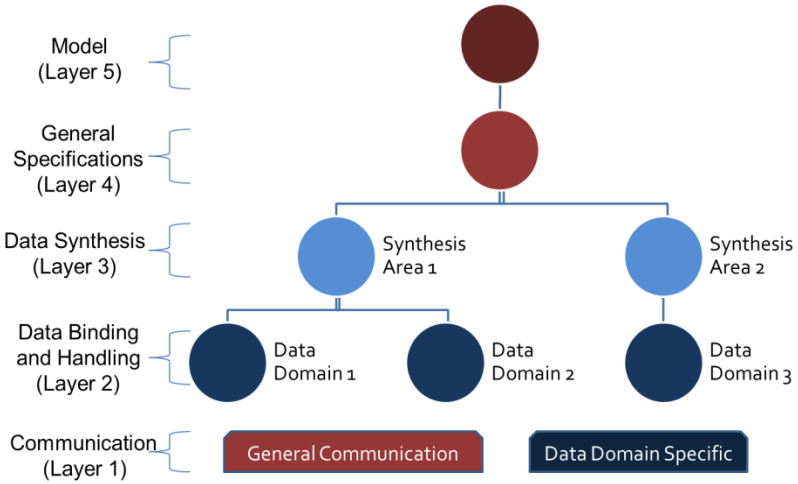
Layer 3: Data Synthesis

- Goal: Extract knowledge from the combination of multiple data domains.
- The other layers "FREE THE DATA" for analysis.
- Due to differing approaches, it is difficult to identify a best practice to document.
- Slated for future work once best practices emerge.



Layer 2: Data Binding and Handling

- Specifications for binding the high level model to handling data from specific data domains (e.g., configuration management)
- Initial layer 2 specification defined in NISTIR 7800:
  - Asset Management
  - Configuration Management
  - Vulnerability Management
- References layer 1 specs



Layer 1: Communications

- Provides foundational data exchange specifications:
  - data domain agnostic specifications used to support layer 4 (e.g., generic reporting wrappers)
  - data domain specific specifications used to support layers 2 and 3 (e.g., vulnerability information)
- The reference model uses specifications from SCAP 1.2 to support Asset, Configuration and Vulnerability Management:
  - Asset Reporting Format (ARF)
  - Common Configuration Enumeration (CCE)
  - Common Vulnerability Enumeration (CVE)
  - Common Platform Enumeration (CPE)
  - eXtensible Checklist Configuration Description Format (XCCDF)
  - Open Vulnerability and Assessment Language (OVAL)