Technical Foundations for Continuous Security Monitoring

9/29/2010

Peter Mell, David Waltermire, Harold Booth
NIST Senior Computer Scientists

Disclaimer and Caveats

- This presentation explores emerging and notional ideas for continuous monitoring technical foundations
- Application to existing laws, policy, and guidance is intentionally avoided (e.g., FISMA)
- Their exists NO implied policy or even NIST guidance in this presentation

Continuous Monitoring (CM) Presentation Contents

- Section 1: Conceptual Design Level
  - Definition, Essential Characteristics, Maturity Model, and Enterprise Architecture
- Section 2: Technical Design Level
  - Subcomponent Model, Technical Architecture
- Section 3: Implementation Design Level
  - Interfaces
  - Communication models
  - Derived test requirements (DTRs)

Providing a Layered Understanding

Driving from definitions to product testing requirements

- Definition
    - Essential Characteristics
        - Maturity Model
            - Enterprise Architecture
                - Subsystem Model
                    - Technical Architecture
                        - Interface Specifications
                            - Communication Specifications
                                - Testing Requirements

Section 1: Conceptual Design Level

- CM Definition
- Essential Characteristics
- Maturity Model
- Enterprise Architecture

Notional Definition of Continuous Monitoring (CM) for use with Technical Reference Architectures

Continuous Monitoring is a risk management approach to cybersecurity that maintains an accurate picture of an organization’s security risk posture, provides visibility into assets, and leverages use of automated data feeds to quantify risk, ensure effectiveness of security controls, and implement prioritized remedies.

The purpose of providing this definition is to enable us to determine the technical requirements for a CM reference architecture

Derived CM Characteristics:

- Maintains an accurate picture of an organization’s security risk posture
- Provides visibility into assets
- Leverages automated data feeds
- Quantifies risk
- Ensures continued effectiveness of security controls
- Informs automated or human-assisted implementation of remediation
- Enables prioritization of remedies

Possible domains that CM could support

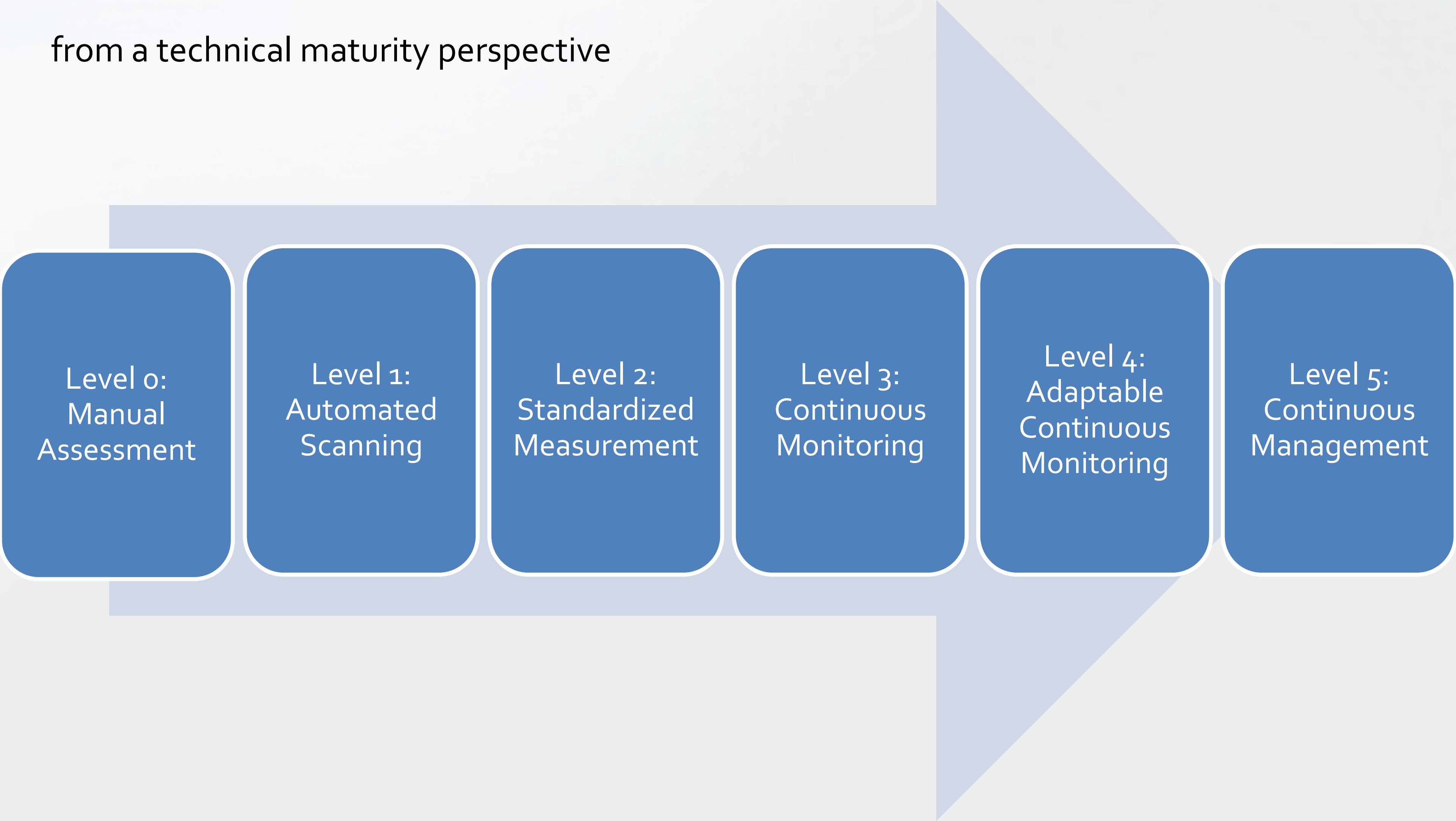
- Asset Management
- Configuration Management
- Event Management
- Incident Management
- Information Management
- License Management
- Malware Detection and Remedy
- Network Management
- Patch Management
- Software Assurance??
- Vulnerability Management

Ways to Achieve CM in Your Organization

- Create ad-hoc system
  - Integrating vendor solutions to create a CM capability
  - Duplicating the work and repeating the mistakes of others
- Procure entire CM solutions from a single vendor
  - Locking into a solution that will be strong in some areas and weak in others
- Leverage a **CM technical reference architecture** and **related security standards** (e.g., SCAP)
  - Use your existing security products
  - Reduce integration costs
  - Combine best of breed solutions

Notional Maturity Model for Continuous Monitoring

from a technical maturity perspective



Level 0: Manual Assessment

Level 1: Automated Scanning

Level 2: Standardized Measurement

Level 3: Continuous Monitoring

Level 4: Adaptable Continuous Monitoring

Level 5: Continuous Management

CM Maturity Levels 0-3

- Level 0: Manual Assessment
  - Security assessments lack automated solutions
- Level 1: Automated Scanning
  - Decentralized use of automated scanning tools
    - Either provided centrally or acquired per system
  - Reports generated independently for each system
- Level 2: Standardized Measurement
  - Reports generated independently for each system
  - Enable use of standardized content (e.g., USGCB/FDCC, CVE, CCE)
- Level 3: Continuous Monitoring
  - Reports generated independently for each system
  - Federated control of automated scanning tools
  - Diverse security measurements aggregated into risk scores
    - Requires standard measurement system, metrics, and enumerations
  - Comparative risk scoring is provided to enterprise (e.g., through dashboards)
  - Remediation is motivated and tracked by distribution of risk scores

CM Maturity Levels 4-5

- Maturity level 4: Adaptable Continuous Monitoring
  - Enable plug-and-play CM components (e.g., using standard interfaces)
  - Result formats are standardized
  - Centrally initiated ad-hoc automated querying throughout enterprise on diverse devices (e.g., for the latest US-CERT alert)
- Maturity level 5: Continuous Management
  - Risk remedy capabilities added (both mitigation and remediation)
  - Centrally initiated ad-hoc automated remediation throughout enterprise on diverse devices (with review and approval of individual operating units)
    - Requires adoption of standards based remediation languages, policy devices, and validated tools

Maturity Model Level Characteristics

	Level 0	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Level 5**
Interfaces	Undefined	Unused	Unused	Proprietary	Standardized	Standardized
Security Check Content Format	Prose	Proprietary	Some Standardization	Some Standardization	Fully Standardized	Fully Standardized
Reporting	Ad hoc	Proprietary and not Integrated	Proprietary and not Integrated	Coarse integration / some standardization	Standardized integration	Standardized integration
Remedies	Manual	Manual or Proprietary	Manual or Proprietary	Manual or Proprietary	Manual or Proprietary	Standardized Automation

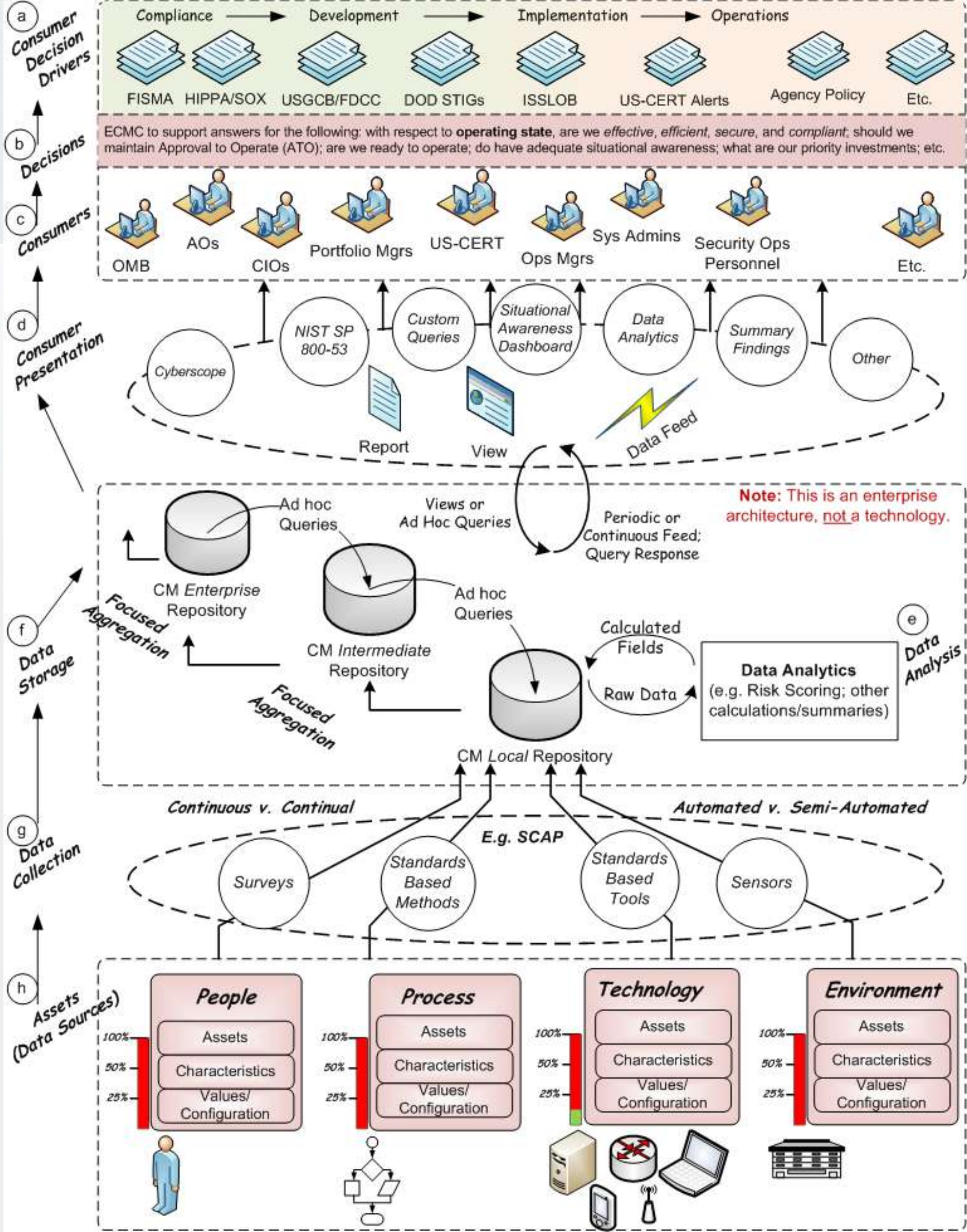
Important CM solution goals:

- Component based approach
  - Based on a standardized reference architecture
  - Solutions from multiple vendors can be combined together to create a CM solution
- Standard-based for interoperability and scoring consistency
  - Languages
    - Using the same machine-readable expressions for checking and remediating machine state (e.g., FDCC policy)
  - Metrics
    - Using the same equations for risk calculations
  - Nomenclatures
    - Using the same names for vulnerabilities, assets, configuration issues, and remediation options.
- Mathematically rigorous scoring approach
  - Motivational scoring is important
  - True risk calculations are also needed

Notional CM Enterprise Architecture

- **This shows an enterprise architecture view, not a technology focus view**

Diagram derived from other government work



a Consumer Decision Drivers

Compliance → Development → Implementation → Operations

FISMA HIPPA/SOX USGCB/FDCC DOD STIGs ISSLOB US-CERT Alerts Agency Policy Etc.

b Decisions

ECMC to support answers for the following: with respect to **operating state**, are we *effective*, *efficient*, *secure*, and *compliant*; should we maintain Approval to Operate (ATO); are we ready to operate; do have adequate situational awareness; what are our priority investments; etc.

c Consumers

OMB AOs CIOs Portfolio Mgrs US-CERT Ops Mgrs Sys Admins Security Ops Personnel Etc.

d Consumer Presentation

Cyberscope NIST SP 800-53 Custom Queries Situational Awareness Dashboard Data Analytics Summary Findings Other

Report View Data Feed

Note: This is an enterprise architecture, not a technology.

Ad hoc Queries

Views or Ad Hoc Queries

Periodic or Continuous Feed; Query Response

CM *Enterprise* Repository

CM *Intermediate* Repository

Ad hoc Queries

Focused Aggregation

Focused Aggregation

Calculated Fields

Raw Data

Data Analytics (e.g. Risk Scoring; other calculations/summaries)

CM *Local* Repository

e Data Analysis

f Data Storage

g Data Collection

Continuous v. Continual

Automated v. Semi-Automated

E.g. SCAP

Surveys Standards Based Methods Standards Based Tools Sensors

h Assets (Data Sources)

People: Assets, Characteristics, Values/ Configuration

Process: Assets, Characteristics, Values/ Configuration

Technology: Assets, Characteristics, Values/ Configuration

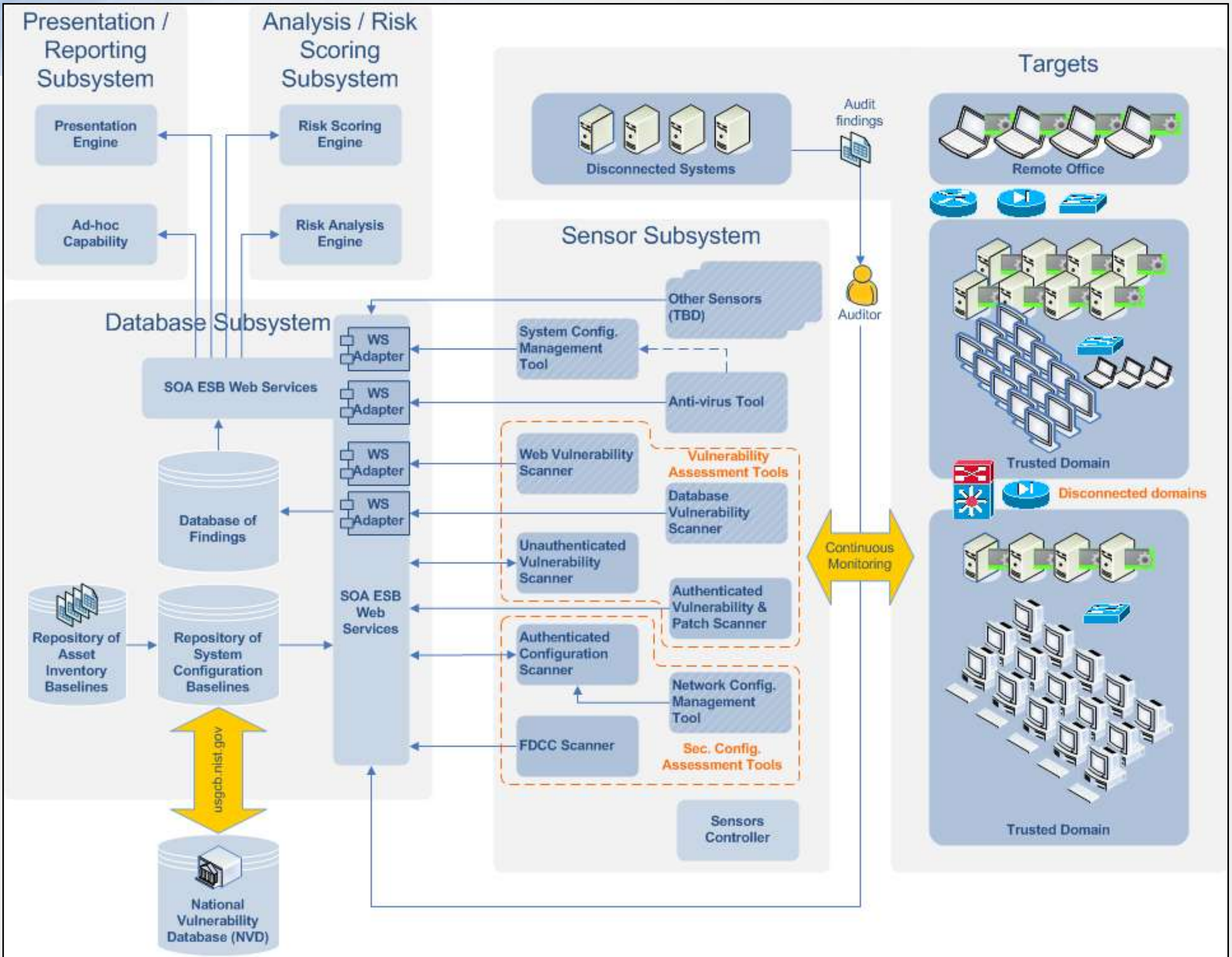
Environment: Assets, Characteristics, Values/ Configuration

100% 50% 25%

Section 2: Technical Architecture Design Level

- Technical Models
- Subcomponent Design
- Interface Identification

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

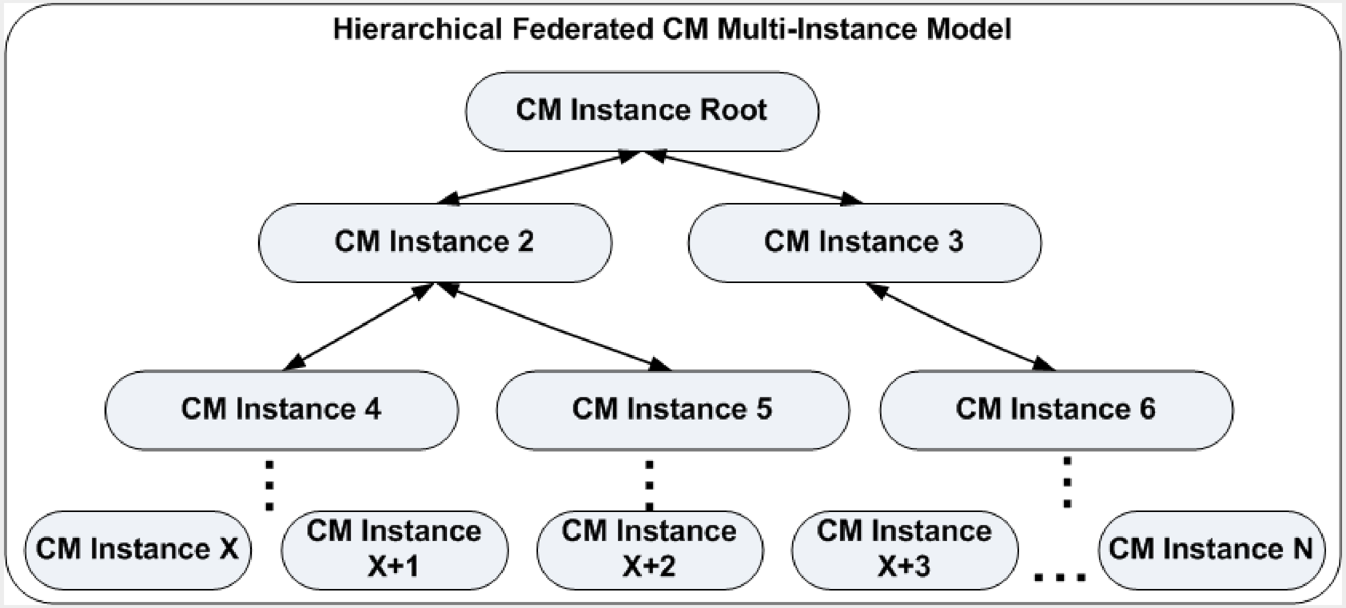

Presentation / Reporting Subsystem
Presentation Engine
Ad-hoc Capability
Analysis / Risk Scoring Subsystem
Risk Scoring Engine
Risk Analysis Engine
Database Subsystem
SOA ESB Web Services
WS Adapter
WS Adapter
WS Adapter
WS Adapter
SOA ESB Web Services
Database of Findings
Repository of Asset Inventory Baselines
Repository of System Configuration Baselines
usgcb.nist.gov
National Vulnerability Database (NVD)
Disconnected Systems
Audit findings
Auditor
Sensor Subsystem
Other Sensors (TBD)
System Config. Management Tool
Anti-virus Tool
Web Vulnerability Scanner
Vulnerability Assessment Tools
Database Vulnerability Scanner
Unauthenticated Vulnerability Scanner
Authenticated Vulnerability & Patch Scanner
Authenticated Configuration Scanner
Network Config. Management Tool
FDCC Scanner
Sec. Config. Assessment Tools
Sensors Controller
Continuous Monitoring
Targets
Remote Office
Trusted Domain
Disconnected domains
Trusted Domain

Notional Ideas for Enhanced Models

- Multiple subsystems instances (already in CAESARS)
- CM multi-instance capability (e.g., hierarchical tiers)
- Interface definitions
- Enhanced communication payload definitions
- Specifications describing subcomponent functionality
  - Could lead to a product validation program or agency procurement (e.g., DHS ISSLOB)

Hierarchical Federated Architecture

- Large organizations will have more than one CM instance
- CM instances are usually arranged in a logical hierarchy
  - Aggregated reports travel up the tree
  - Data calls and configuration requirements travel down the tree
- Often CM instances have a degree of autonomy resulting in a federated style of communication
  - Each instance may have approval authority on directives from higher levels
- Lateral communication in the tree is also possible



Notional CM Instance Subcomponents

- Organizations may have multiple CM instances
- CM System Instance Subsystems
  - 1+ Presentation / Reporting Subsystem
  - 1+ Analysis / Risk Scoring Subsystem
  - 1 Data Aggregation Subsystem
  - 1+ Sensor Subsystem
  - 0-1 Content Subsystem (need 1 somewhere in enterprise)
  - 0-1 Task Manager Subsystem (optional but valuable)
- Outside entities
  - National Vulnerability database (NVD)
  - U.S. Government Configuration Baseline (USGCB)

Why Have a Task Manager Subsystem?

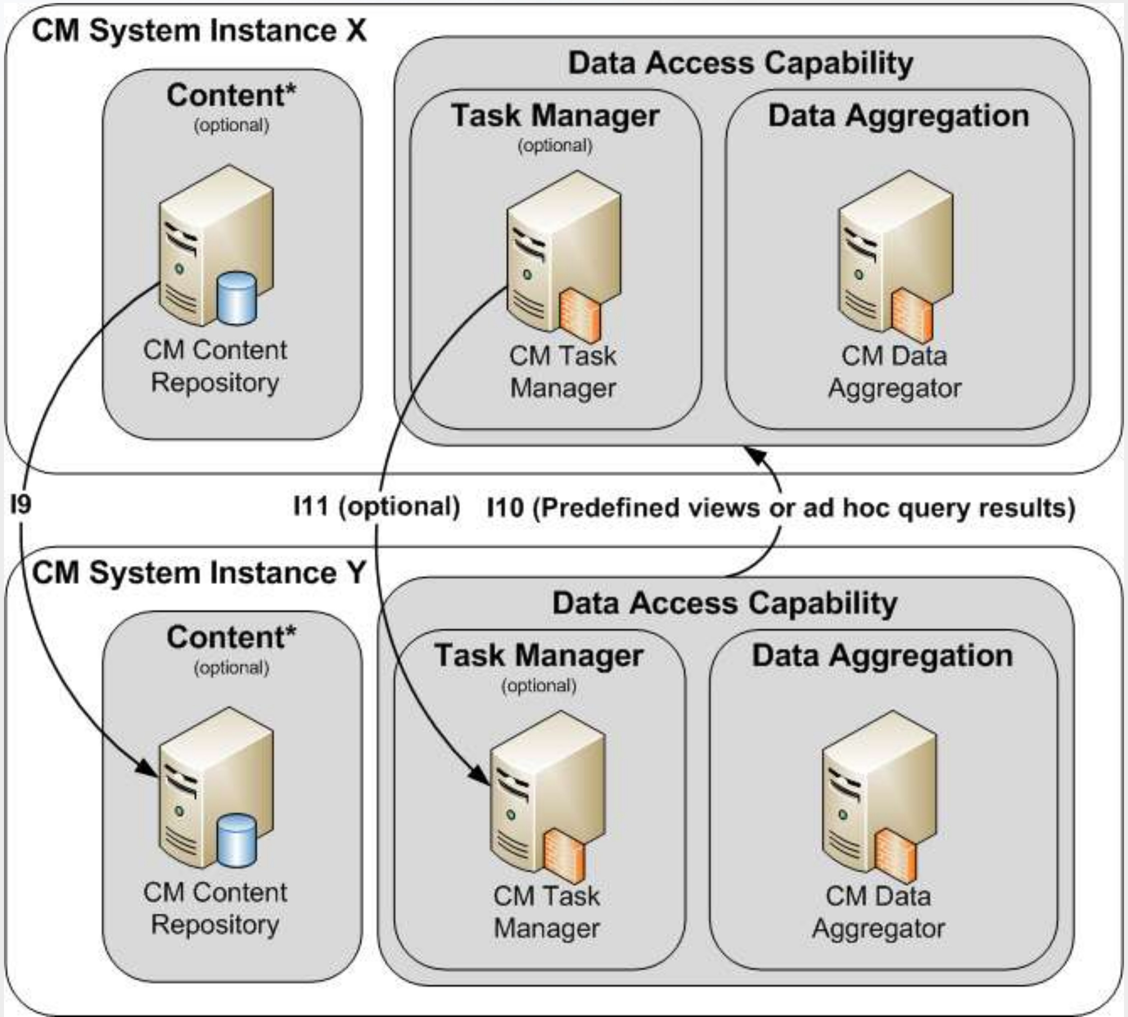
- Single CM Instance
  - Orchestrates scanning, aggregation and reporting activities within the system
    - Harness input from diverse security devices
  - Enable ad hoc queries from dashboard
    - Automatically retrieve data not already in data aggregation subsystem
- Multi-instance Federated Hierarchical Architecture
  - Avoid tendency (and possibly need) to aggregate all data up through all tiers
  - Enable higher tiers to request specific data from lower tiers
  - Provide policy management of requests entering a tier
  - Could enable a “big easy” button with safety controls and tiered human review and approval

Why Have a Content Subsystem?

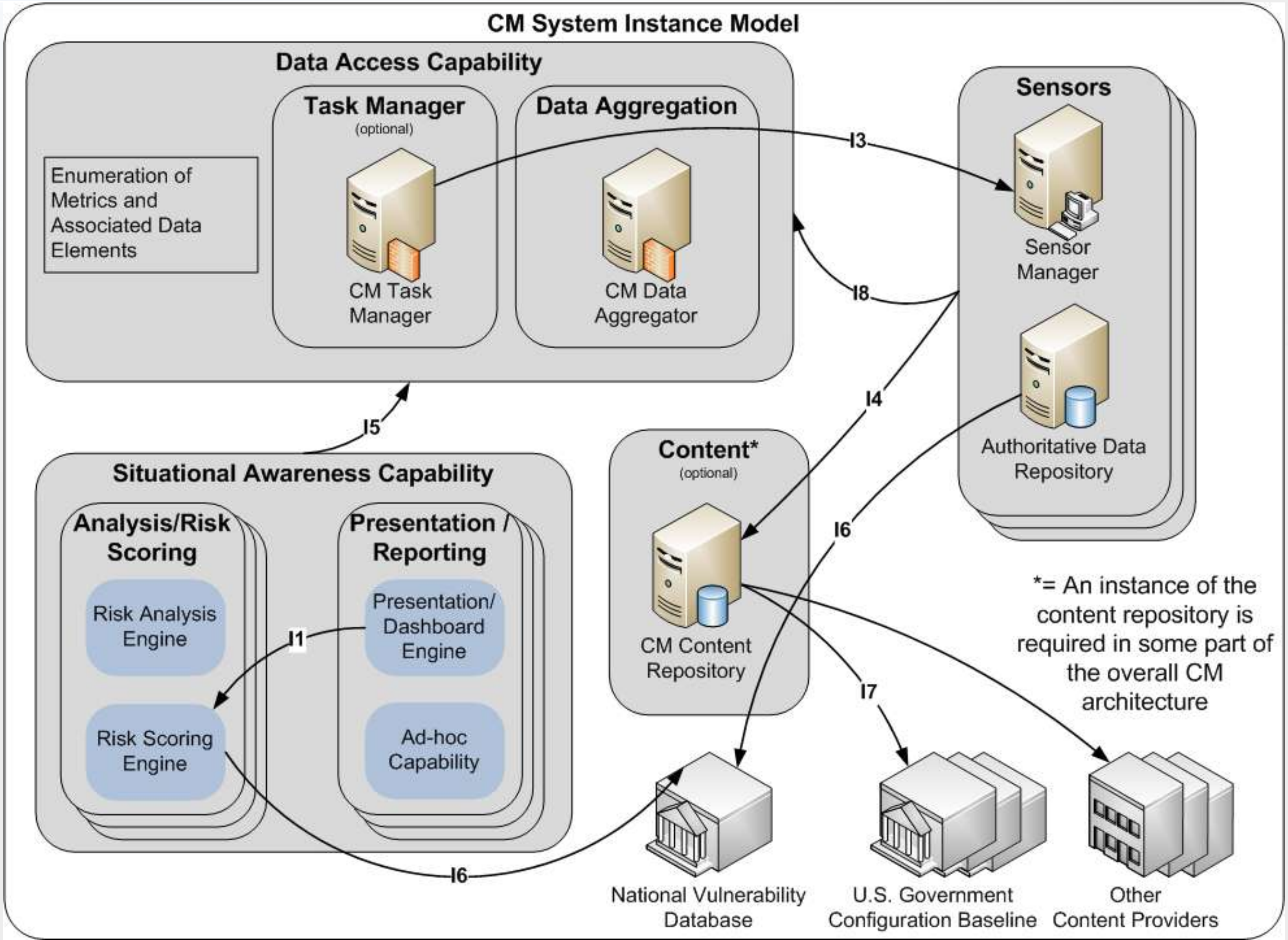
- Enables both organization-wide and locally-scoped content
- Holds machine readable security baselines (e.g., Federal Desktop Core Configuration)
- Allows organizations to tailor or augment baselines for their own needs
- Typical baseline standards include:
  - Extensible Configuration Checklist Description Format (XCCDF)
  - Open Vulnerability and Assessment Language (OVAL)
  - Open Checklist Interactive Language (OCIL)
- Content subsystem implementation approaches
  - 1 content subsystem for entire organization
  - 1 content subsystem per CM instance
    - Adds complexity to score aggregation (apples vs. oranges)
  - Hybrid model (allow only certain tiers to customize)

Notional Multi-instance CM Architecture

- This view shows the relationship between CM instances
- These interfaces enable the hierarchical federated CM architecture



Notional CM Instance Architecture



Section 3: Implementation Design Level

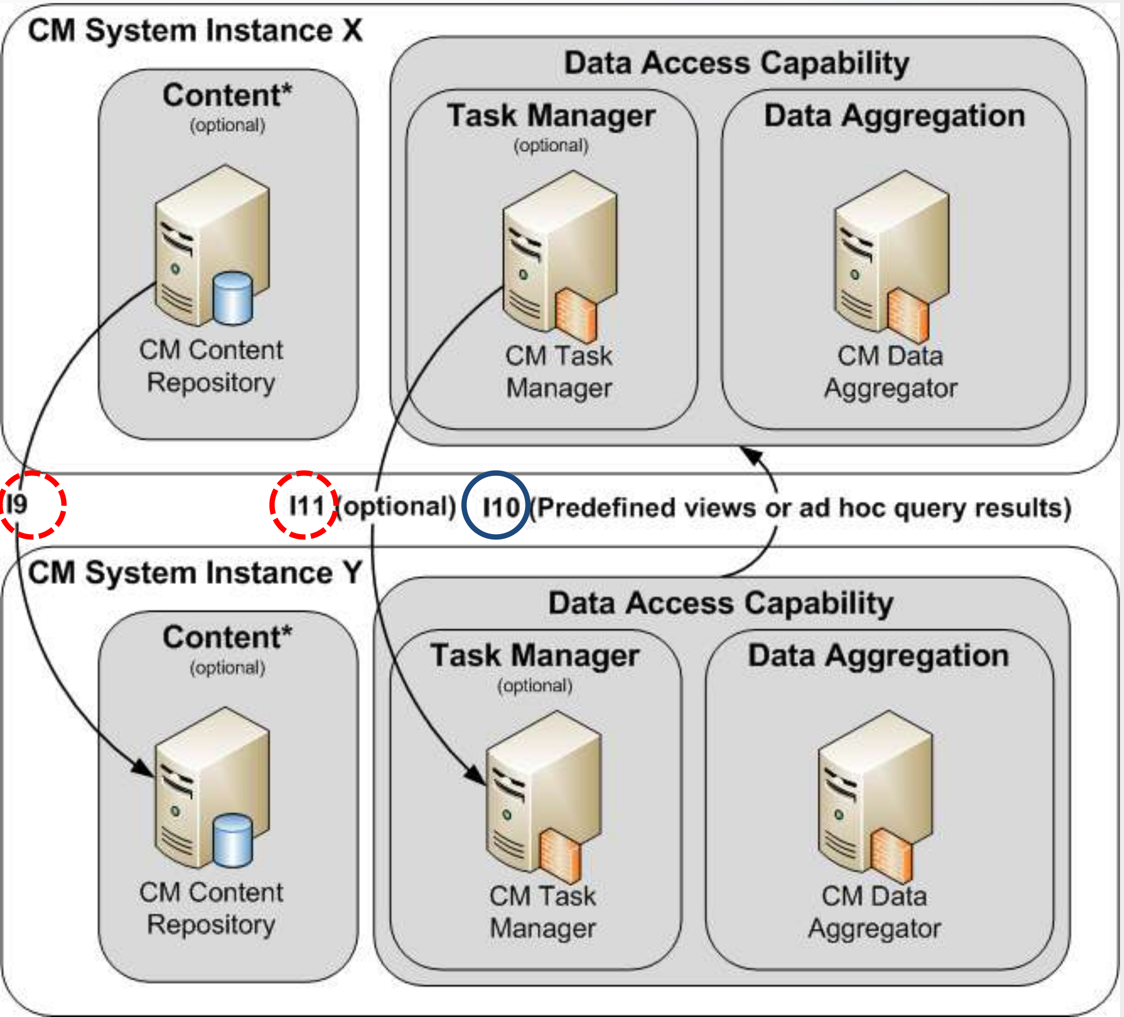
- Interface Specifications
- Communication Models
- Derived Test Requirements

Challenges in Defining Interfaces (payload + communication mechanism)

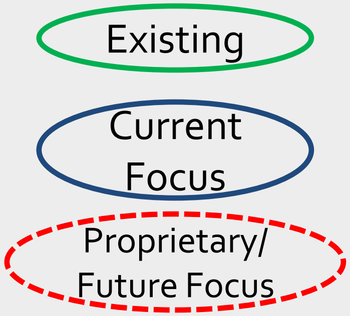
- I4, I8, and I10: Focus of this work (I8 substantially addressed through DHS CAESARS)
- I6: National Vulnerability Database (NVD) XML file and WSDL interfaces are defined
- I1, I3, and I5: No current standards exist for arbitrary data retrieval
  - Use of SQL would require mandating a particular database schema to be implemented within products
  - Interfaces could be left proprietary in the short term and we could watch for best of breed solutions to appear from vendors
  - Refinement of Policy Language for Assessments Results Reporting (PLARR) to address part of the problem
- I7, I9, I11: Future work on multi-tier request and security automation content propagation

Notional Multi-instance CM Architecture

- This view shows the relationship between CM instances
- These interfaces enable the hierarchical federated CM architecture

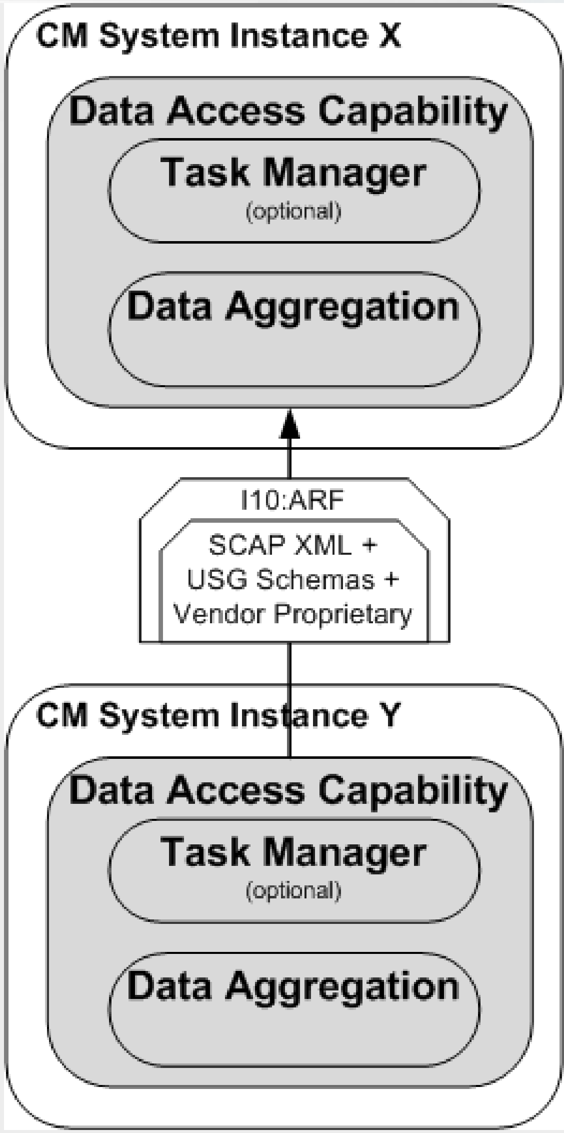


Interface and Payload Specifications:

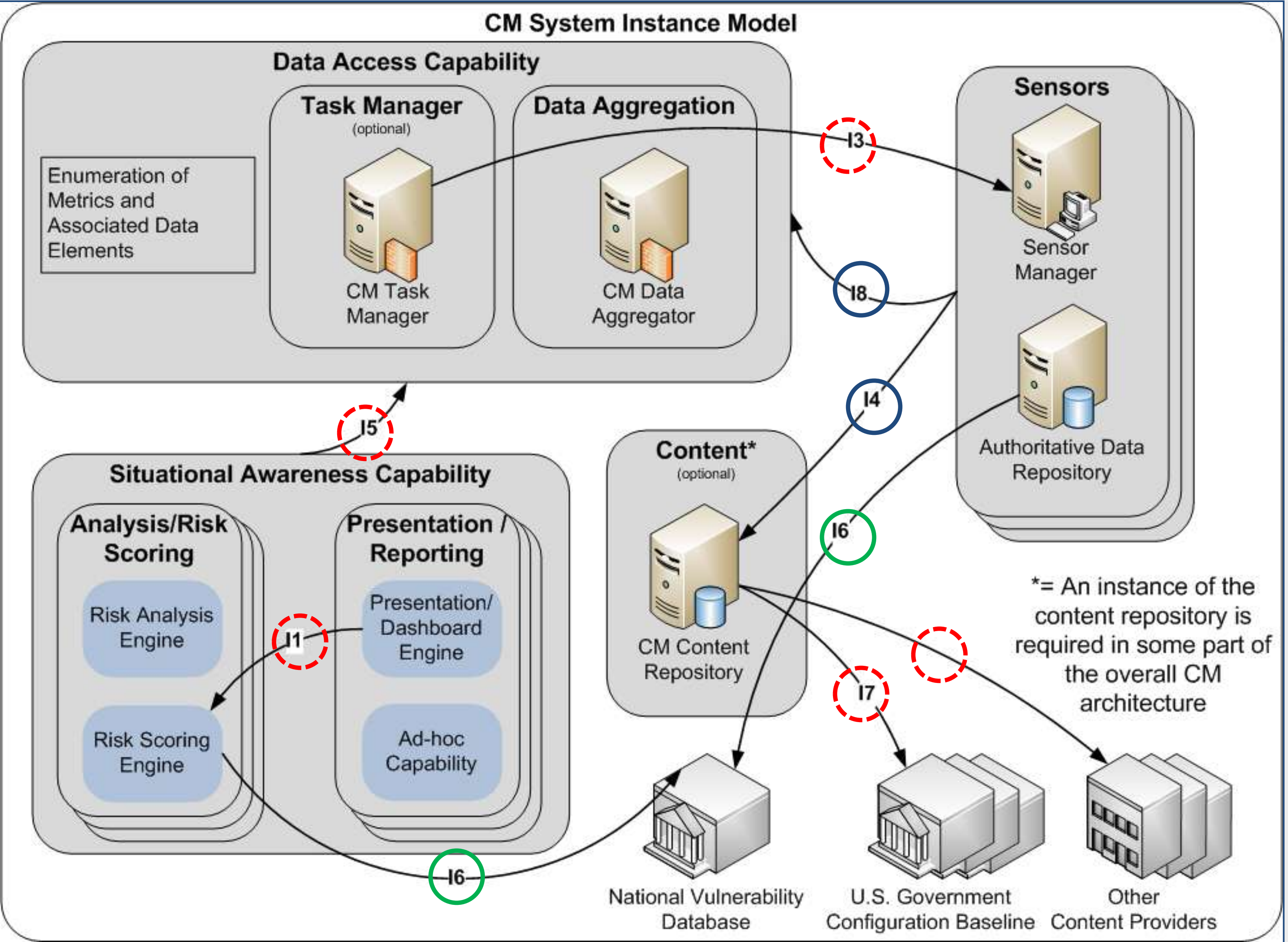


Notional Interface Overview: I10

- Interfaces:
  - Service Oriented Architecture
    - Web Services Description Language (WSDL) direct connection
    - Enterprise Service Bus
  - Other interfaces??
- XML communication envelope: Asset Reporting Format (ARF)
- XML payload options:
  - USG XML schema data (based on USG agreed upon metrics)
  - SCAP XML (e.g., XCCDF results, OVAL results)
  - Vendor proprietary XML
- Use of proprietary payloads may require additional integration and loss of plug and play compatibility



Notional CM Instance Architecture



*= An instance of the content repository is required in some part of the overall CM architecture

Interface and Payload Specifications:

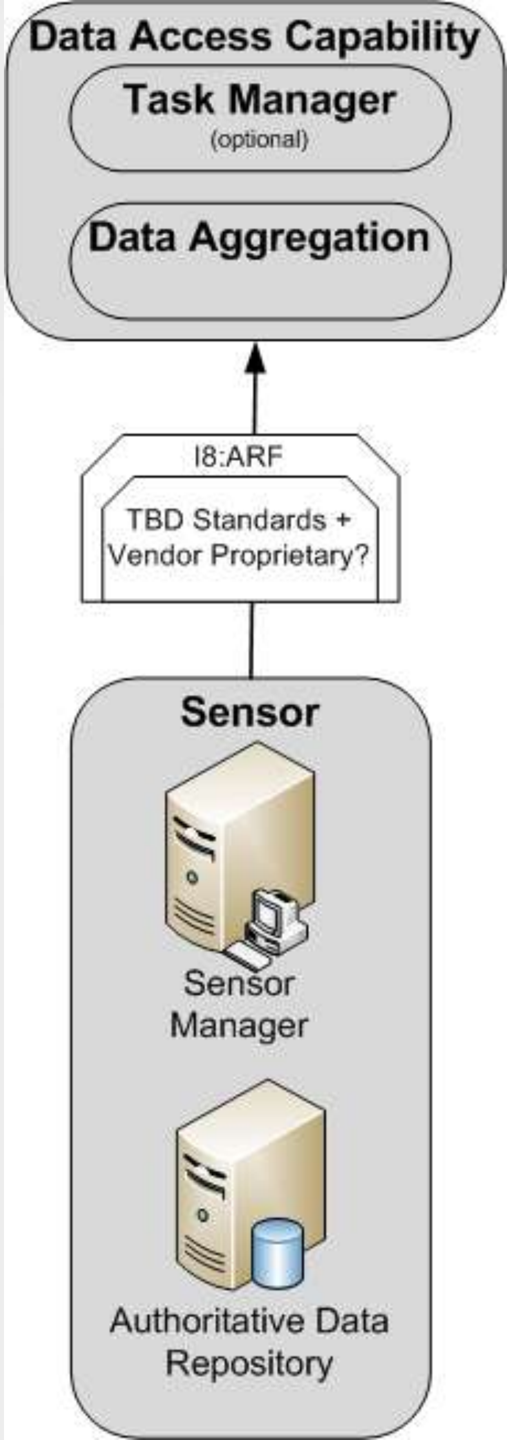
Existing

Current Focus

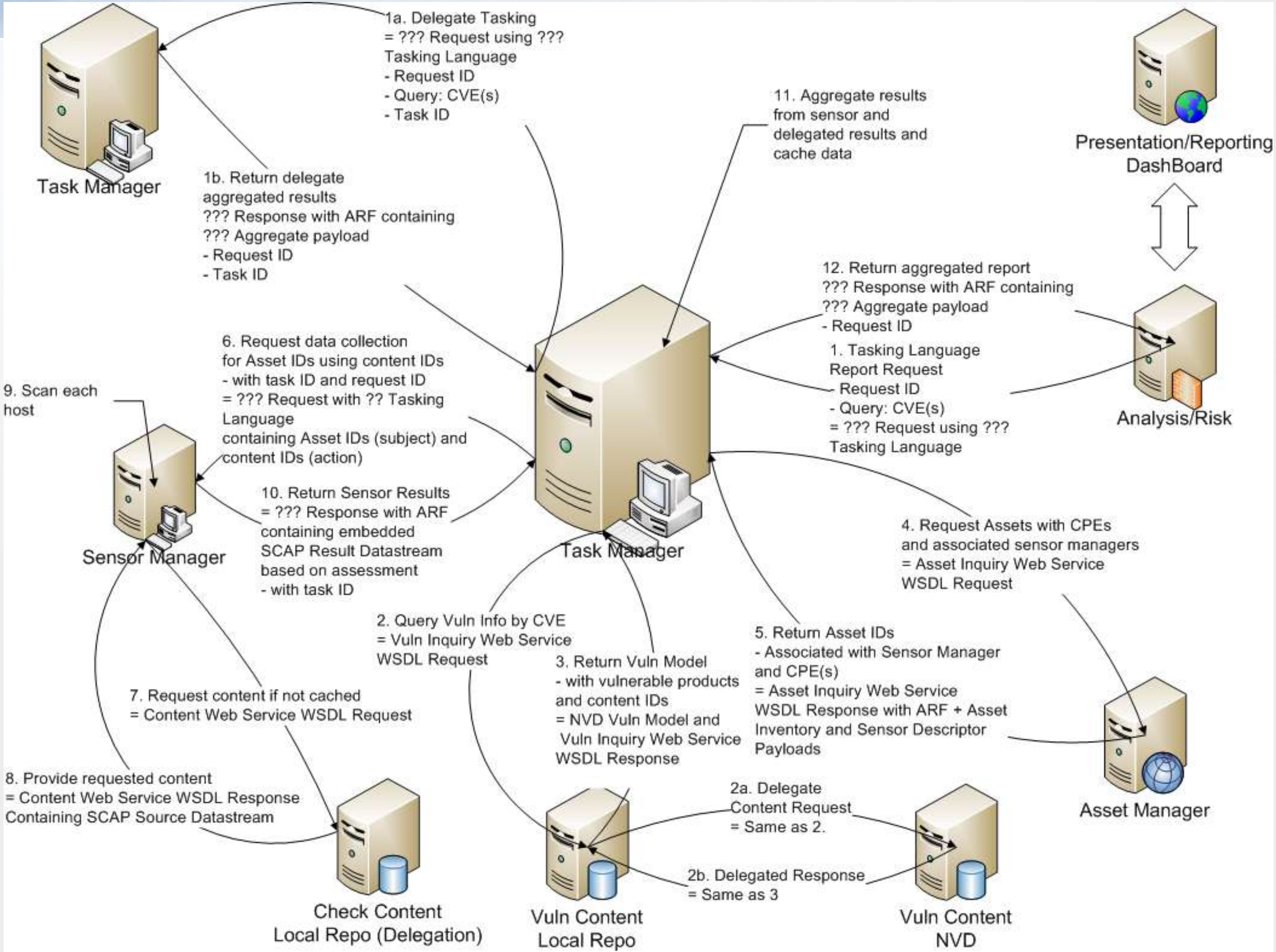
Proprietary/ Future Focus

Notional Interface Overview: I8

- Interfaces:
  - Service Oriented Architecture
    - WSDL direct connection
    - Enterprise Service Bus
  - Other interfaces??
- XML communication envelope: ARF
- XML payload options:
  - Need to define standards-based payload(s) to support all sensor types
    - System configuration management
    - Anti-virus
    - Web vulnerability scanner
    - Database vulnerability scanner
    - Unauthenticated vulnerability scanner
    - Authenticated vulnerability and patch scanner
    - Authenticated configuration scanner
    - Network configuration management tools
    - Federal Desktop Core Configuration scanner
  - Leverage Security Content Automation Protocol XML (e.g., XCCDF results, OVAL results)
  - Allow vendor proprietary XML??



Notional and Under Development Communication Models



1. Tasking Language Report Request
- Request ID
- Query: CVE(s)
= ??? Request using ??? Tasking Language

1a. Delegate Tasking
= ??? Request using ??? Tasking Language
- Request ID
- Query: CVE(s)
- Task ID

1b. Return delegate aggregated results
??? Response with ARF containing ??? Aggregate payload
- Request ID
- Task ID

2. Query Vuln Info by CVE
= Vuln Inquiry Web Service WSDL Request

2a. Delegate Content Request
= Same as 2.

2b. Delegated Response
= Same as 3

3. Return Vuln Model
- with vulnerable products and content IDs
= NVD Vuln Model and Vuln Inquiry Web Service WSDL Response

4. Request Assets with CPEs and associated sensor managers
= Asset Inquiry Web Service WSDL Request

5. Return Asset IDs
- Associated with Sensor Manager and CPE(s)
= Asset Inquiry Web Service WSDL Response with ARF + Asset Inventory and Sensor Descriptor Payloads

6. Request data collection for Asset IDs using content IDs
- with task ID and request ID
= ??? Request with ?? Tasking Language
containing Asset IDs (subject) and content IDs (action)

7. Request content if not cached
= Content Web Service WSDL Request

8. Provide requested content
= Content Web Service WSDL Response Containing SCAP Source Datastream

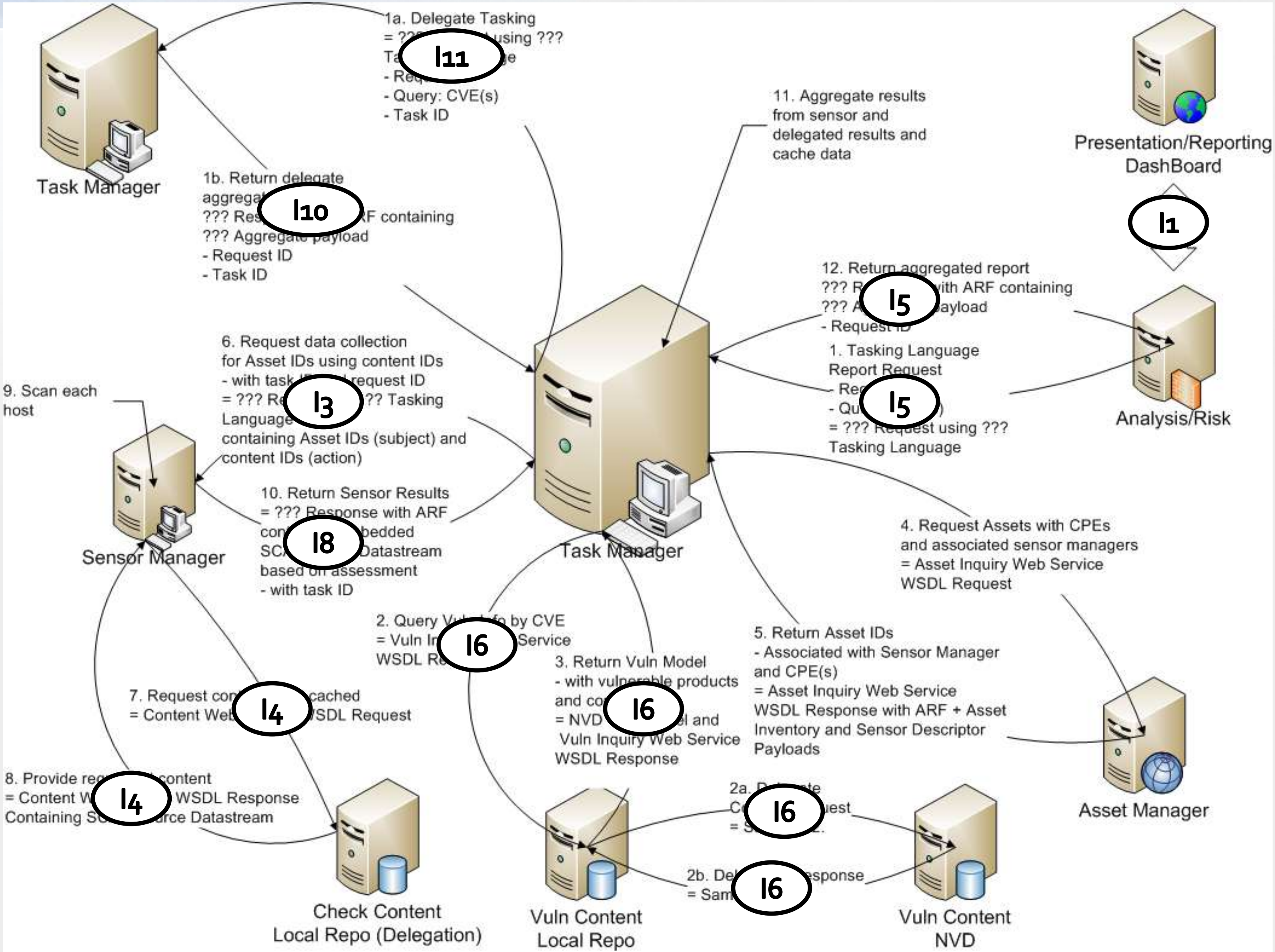
9. Scan each host

10. Return Sensor Results
= ??? Response with ARF containing embedded SCAP Result Datastream based on assessment
- with task ID

11. Aggregate results from sensor and delegated results and cache data

12. Return aggregated report
??? Response with ARF containing ??? Aggregate payload
- Request ID

Communication Models Map to Interfaces



1a. Delegate Tasking
= ??? using ???
Ta ge
- Req
- Query: CVE(s)
- Task ID

1b. Return delegate
aggrega
??? Res F containing
??? Aggregate payload
- Request ID
- Task ID

11. Aggregate results
from sensor and
delegated results and
cache data

12. Return aggregated report
??? R with ARF containing
??? A ayload
- Request ID

1. Tasking Language
Report Request
- Re
- Qu)
= ??? Request using ???
Tasking Language

6. Request data collection
for Asset IDs using content IDs
- with task request ID
= ??? Re ?? Tasking
Language
containing Asset IDs (subject) and
content IDs (action)

9. Scan each
host

10. Return Sensor Results
= ??? Response with ARF
con bedded
SC Datastream
based on assessment
- with task ID

4. Request Assets with CPEs
and associated sensor managers
= Asset Inquiry Web Service
WSDL Request

2. Query Vuln Info by CVE
= Vuln In Service
WSDL Re

3. Return Vuln Model
- with vulnerable products
and co
= NVD el and
Vuln Inquiry Web Service
WSDL Response

5. Return Asset IDs
- Associated with Sensor Manager
and CPE(s)
= Asset Inquiry Web Service
WSDL Response with ARF + Asset
Inventory and Sensor Descriptor
Payloads

7. Request con cached
= Content Web SDL Request

8. Provide re content
= Content W WSDL Response
Containing SC rce Datastream

2a. D te
Co uest
= S .

2b. De esponse
= Sam

Closing Thoughts

- There exists great momentum surrounding CM (both executive level and grass roots)
  - Dashboards, “big easy” buttons, aggregated reporting of technical metrics
- Agencies can leverage their existing security tools to evolve towards an automated CM solution
  - Enhance their own capability and meet upcoming reporting demands
- Reference architectures
  - Can reduce integration efforts
  - Enable CM plug-and-play component capabilities
    - Product validation and procurement programs can assist with tool adoption of necessary technical specifications
  - Focus agencies on evolving toward the full potential of CM
- The long term vision will take time and effort, but significant gains are achievable today.

Acknowledgements and Credit

- Much of this was inspired and encouraged by others
  - Information Security and Identity Management Committee (ISIMC) CM working group
  - DHS Federal Network Security (Cyberscope and CAESARS)
  - NSA Information Assurance Directorate (IAD)
  - NIST Security Content Automation Protocol (SCAP) team
  - MITRE McLean CAESARS team
  - MITRE Bedford SCAP team

Summary and Questions

Presenters:

Peter Mell
NIST Senior Computer Scientist
301-975-5572
peter.mell@nist.gov

David Waltermire
NIST Senior Computer Scientist
301-975-3390
david.waltermire@nist.gov