

CAESARS Framework Extension: An Enterprise Continuous Monitoring Technical Reference Model

Documented in NIST IR 7756, jointly developed by
DHS and NIST with NSA participation

3/21/2011

Presenter: Peter Mell
Senior Computer Scientist
National Institute of Standards and Technology
<http://twitter.com/petermmell>

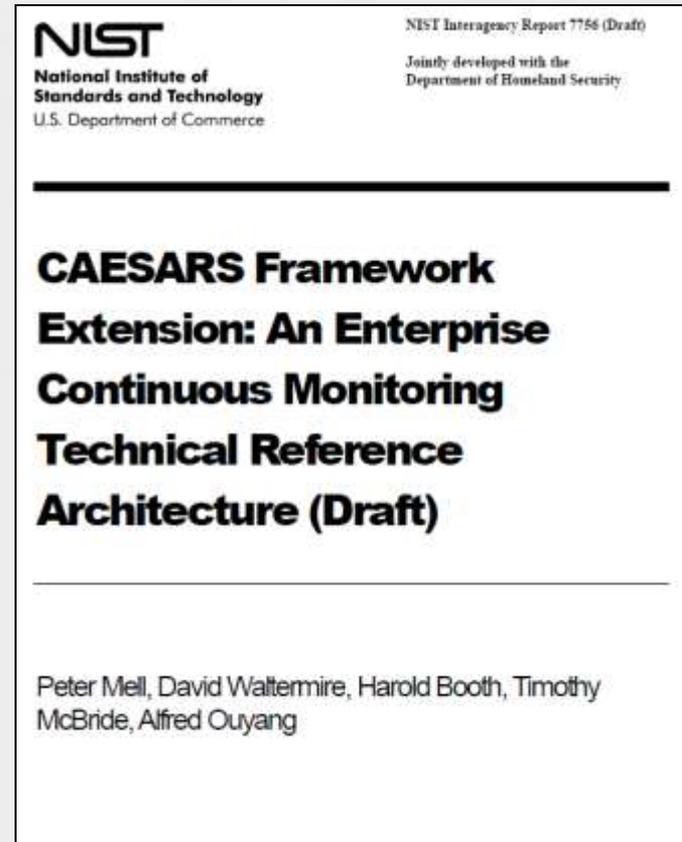
Continuous Monitoring (CM) Architecture Timeline

- 4/2010: Office of Management and Budget CM memo to DHS
 - evaluate CM best practices
- 9/2010: DHS published CAESARS reference architecture
 - based on Department of State, Justice, and Treasury implementations
- 9/2010: ISIMC CM initiated DHS/NSA/NIST research initiative to create the CAESARS Framework Extension (FE)
 - make applicable to entire government, adapt for large enterprises, and further leverage standards
- 2/2011: NIST and DHS published CAESARS FE
 - Draft NIST IR 7756
- 3/2011: CM architecture workshop at NIST March 21
 - <http://scap.nist.gov/events/index.html#cm2011>

CAESARS Framework Extension (FE)

NIST Interagency Report 7756

- U.S. government continuous security monitoring technical reference model
- Jointly created by DHS, NSA, and NIST
- Based on CAESARS: the DHS Continuous Asset Evaluation, Situational Awareness, and Risk Scoring (CAESARS) Reference Architecture
 - <http://www.dhs.gov/xlibrary/assets/fns-caesars.pdf>
- CAESARS FE expands on CAESARS to apply it to large enterprises and to provide enhanced capabilities



http://csrc.nist.gov/publications/drafts/nistir-7756/Draft-nistir-7756_feb2011.pdf

Our CM Architecture Design Team

Peter Sell, David Minge
National Security Agency

Timothy McBride
Department of Homeland Security

Peter Mell, David Waltermire, Harold Booth
National Institute of Standards and Technology

Valery Feldman, Amit Mannan, Zach Ragland, Joe Debra
Booz Allen Hamilton

Alfred Ouyang, Mark Crouter
MITRE

Continuous Monitoring (CM) Architecture Presentation Contents



- Section 1: Architecture Development Plan
- Section 2: Conceptual Design Level
 - Definition, Essential Characteristics, and Enterprise Architecture
- Section 3: Technical Architecture Design Level
 - Subsystems and Component Model
- Section 4: Communication Pattern Level
 - Interfaces
 - Workflow and Subsystem communication
- Section 5: Functional Specification Level
 - Data domain agnostic
 - Data domain specific
- Section 6: CM Maturity Models

Section 1: CAESARS FE Development Plan



- Theoretical Approach
- Goals and Proposed Solutions
- Model Design
- Concrete Specifications

CAESARS FE: Providing a Layered Understanding

Driving from definitions to specifications

- Definition
 - Essential Characteristics
 - Maturity Model
 - Enterprise Architecture
 - Subsystem Model
 - Technical Model
 - Use Cases and Workflow
 - Functional Specifications
 - Interface and Payload Specifications

CAESARS FE Model Derivation

CAESAR FE is a reference model that enables derivation of specific architectures



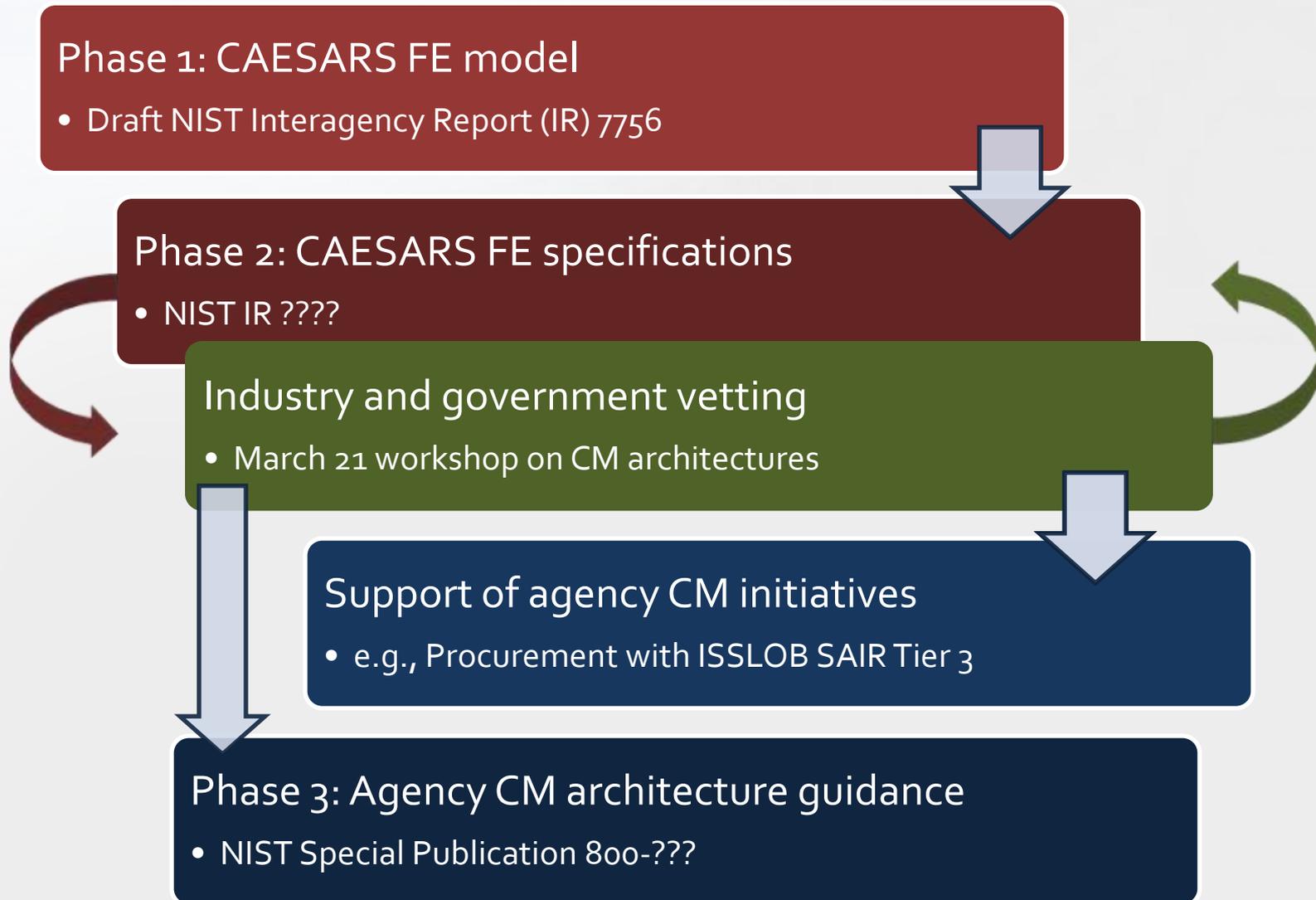
CAESARS FE Reference Model



Derived Architectures

Continuous monitoring domains chosen
Specific systems and software are leveraged
Number of instances determined

CAESARS FE Development Plans



CAESARS FE Reference Model – NIST IR 7756

Phase 1: Draft completed

Leveraged Design Sources:

NSA and NIST SP 800-137

NSA architecture model

DHS CAESARS

CAESARS FE Research

(DHS/NSA/NIST)

CAESARS FE Research

(DHS/NSA/NIST)

CAESARS FE Design Levels:

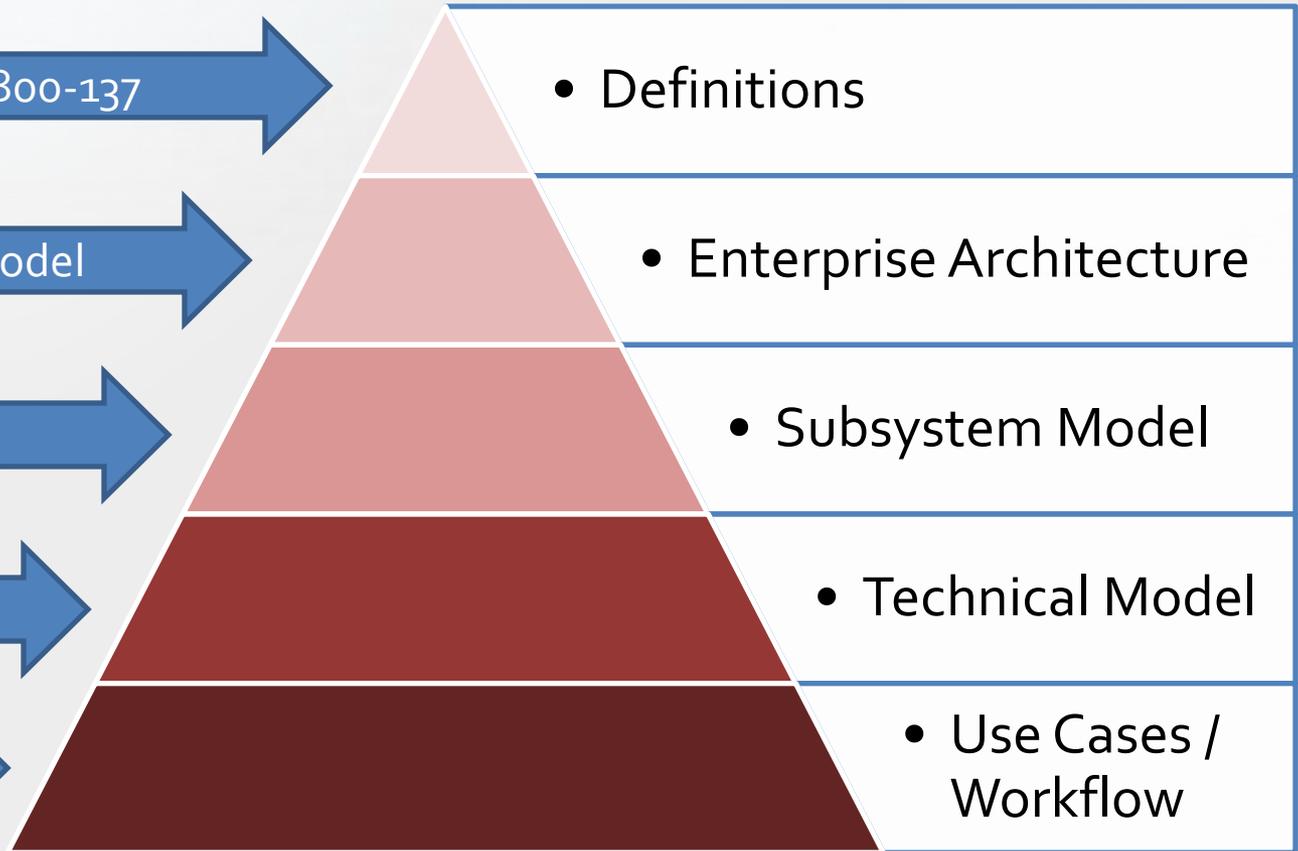
• Definitions

• Enterprise Architecture

• Subsystem Model

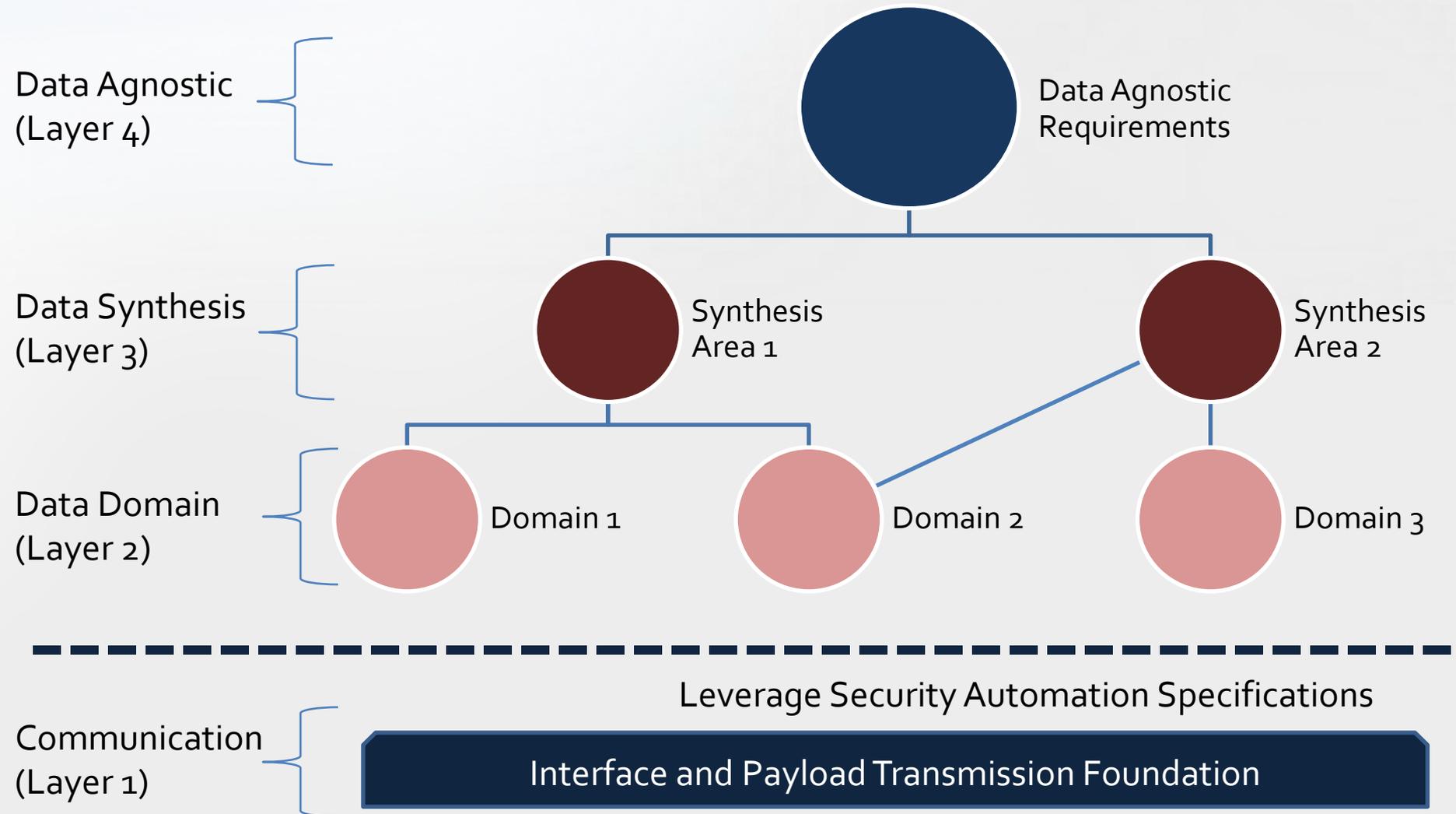
• Technical Model

• Use Cases /
Workflow



CAESARS FE Specification Model

Phase 2: Under development



ISIMC CM Subcommittee Goals 1-3 and Proposed CAESARS FE Solutions



- Goal 1: Enable Federal agencies to implement CM more rapidly.
 - Proposed solution: Leverage CAESARS FE compliant tools to compose enterprise CM capabilities without lengthy and costly custom integration efforts.
- Goal 2: Provide Federal standards to allow integration of information at the Federal Level.
 - Proposed solution: Leverage CAESARS FE interfaces, data normalization, and reports to integrate Federal and agency level CM data.
- Goal 3: Leverage Federal buying power to reduce the cost of implementing CM.
 - Proposed solution: Create the CAESARS FE reference model as a foundation for product procurement (e.g., ISSLOB SAIR Tier 3) and testing. Without this, procurements may be non-interoperable and risk measurement results may be non-comparable.

Section 2: Conceptual Design Level



- CM Definitions
- Essential Characteristics
- Enterprise Architecture

General CM Definition

Continuous monitoring is ongoing observance with intent to provide warning. A continuous monitoring capability is the ongoing observance and analysis of the operational states of systems to provide decision support regarding situational awareness and deviations from expectations.

Thus CM applies to both cybersecurity and information technology domains

Domains that CM can support

- 1) Vulnerability Management
- 2) Patch Management
- 3) Event Management
- 4) Incident Management
- 5) Malware Detection
- 6) Asset Management
- 7) Configuration Management
- 8) Network Management
- 9) License Management
- 10) Information Management
- 11) Software Assurance

Source: NIST SP 800-137



Additional Proposed Domains:
12) Digital Policy Management
13) Advanced Persistent Threat

Description of CM applied to Cybersecurity and for use with Technical Reference Architectures

Continuous security 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 measure security, ensure effectiveness of security controls, and enable prioritization of remedies.

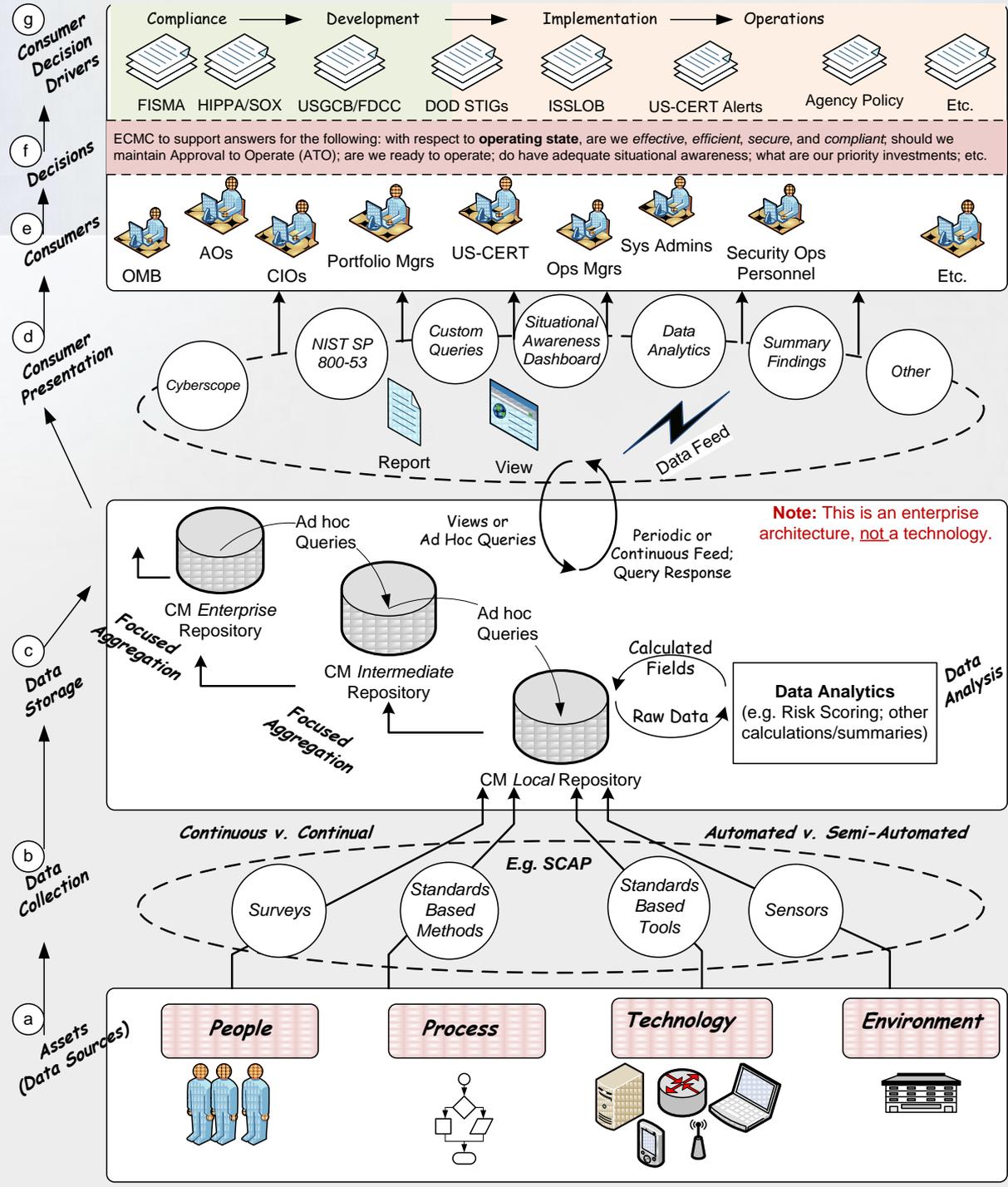
Derived CM Characteristics:

- Maintains an accurate picture of an organization's security posture
- Provides visibility into assets
- Leverages automated data feeds
- Quantifies security measurement
- Ensures continued effectiveness of security controls
- Informs automated or human-assisted implementation of remediation
- Enables prioritization of remedies
- Identifies deviations from expected results

CM Enterprise Architecture

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

Diagram derived from NSA work (original diagram credit: Keith Willett, MITRE)



Ways to Create a Continuous Monitoring Architecture in Your Organization

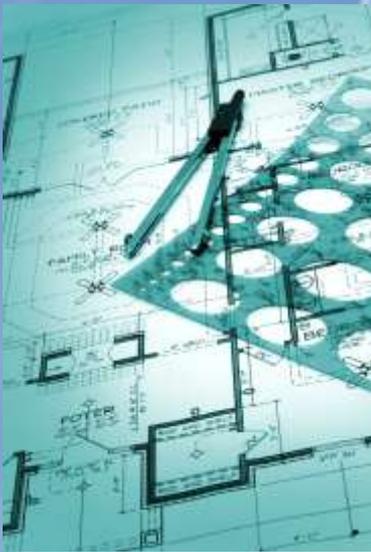
- Create ad-hoc system
 - Integrating vendor solutions to create a CM capability
 - Duplicating the work and repeating the mistakes of others
- Procure entire CM solutions from a single vendor
 - Locking into a solution that will be strong in some areas and weak in others
- Leverage a **CM technical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

Important CM architecture solution goals

- Component based approach
 - Based on a standardized reference model
 - Solutions from multiple vendors can be combined together to create a CM solution
- Standard-based for interoperability and scoring consistency
- Mathematically rigorous scoring approach
 - Risk scoring requires likelihood and system impact measurements
 - Measurement of effectiveness of security posture is more tractable

Section 3: Technical Architecture Design Level

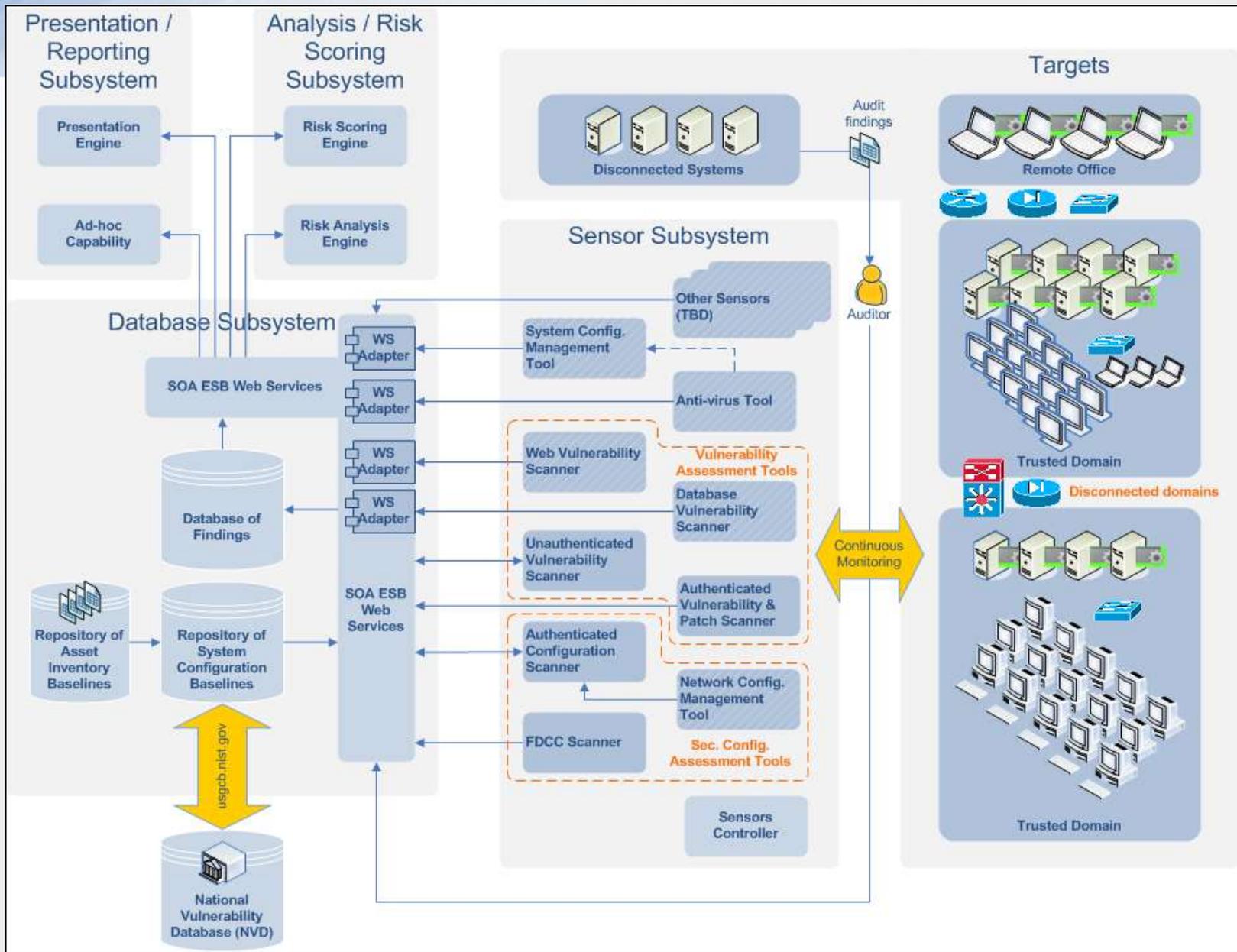
- Subsystem and Component Models



Scoping and External System Interfaces

- CM systems must leverage (not replace) existing data collection repositories from diverse domains
- This said, existing collection systems will need to be instrumented to enable them to interface with the continuous monitoring architecture

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



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-CM instance capability
6. Lack of multi-subsystem instance capability
7. CM database integration with security baseline content
8. Lack of detail on the required asset inventory
9. Requirement for risk measurement

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

CAESARS Framework Extension

- Six subsystem types
 - Presentation / Reporting Subsystem (1 or more)
 - Dashboards, reports, user queries
 - Analysis / Scoring Subsystem (1 or more)
 - Data deconfliction, scoring
 - Data Aggregation Subsystem (1)
 - Central repository
 - Content Subsystem (0 or 1)
 - Holds machine readable policy
 - Task Manager Subsystem (1)
 - Orchestrates query responses and reports
 - Collection Subsystem (0 or more)
 - **EXTERNAL SYSTEMS**
 - Provides data feeds

CAESARS FE Instance Model

(Organizations may have multiple CM instances)

Continuous Monitoring System Instance Model

Situational Awareness Capability

Analysis / Scoring

Scoring
Engine

Data
Deconfliction

Presentation / Reporting

Dashboard
Engine

Reporting
Engine

Data Aggregation

Metrics
Repository

Repository of
Findings

Metadata
Repository
(notional)

Asset
Inventory

Task Manager

Query
Orchestrator

Collection
Controller
(advanced)

Inter-tier
Reporting

Inter-tier
Queries
(advanced)

Collection

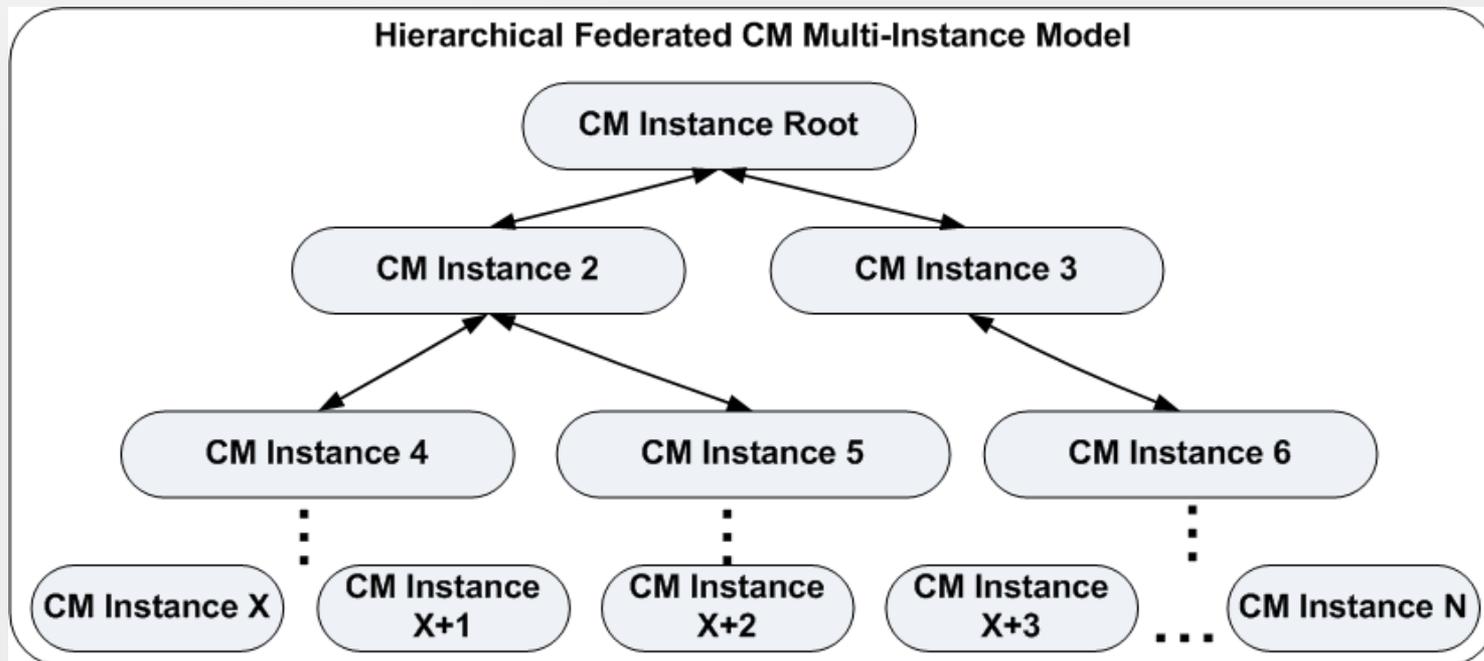
External
systems
instrumented
for CM
integration

Content

Benchmarks,
Baselines,
and
Enumerations

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

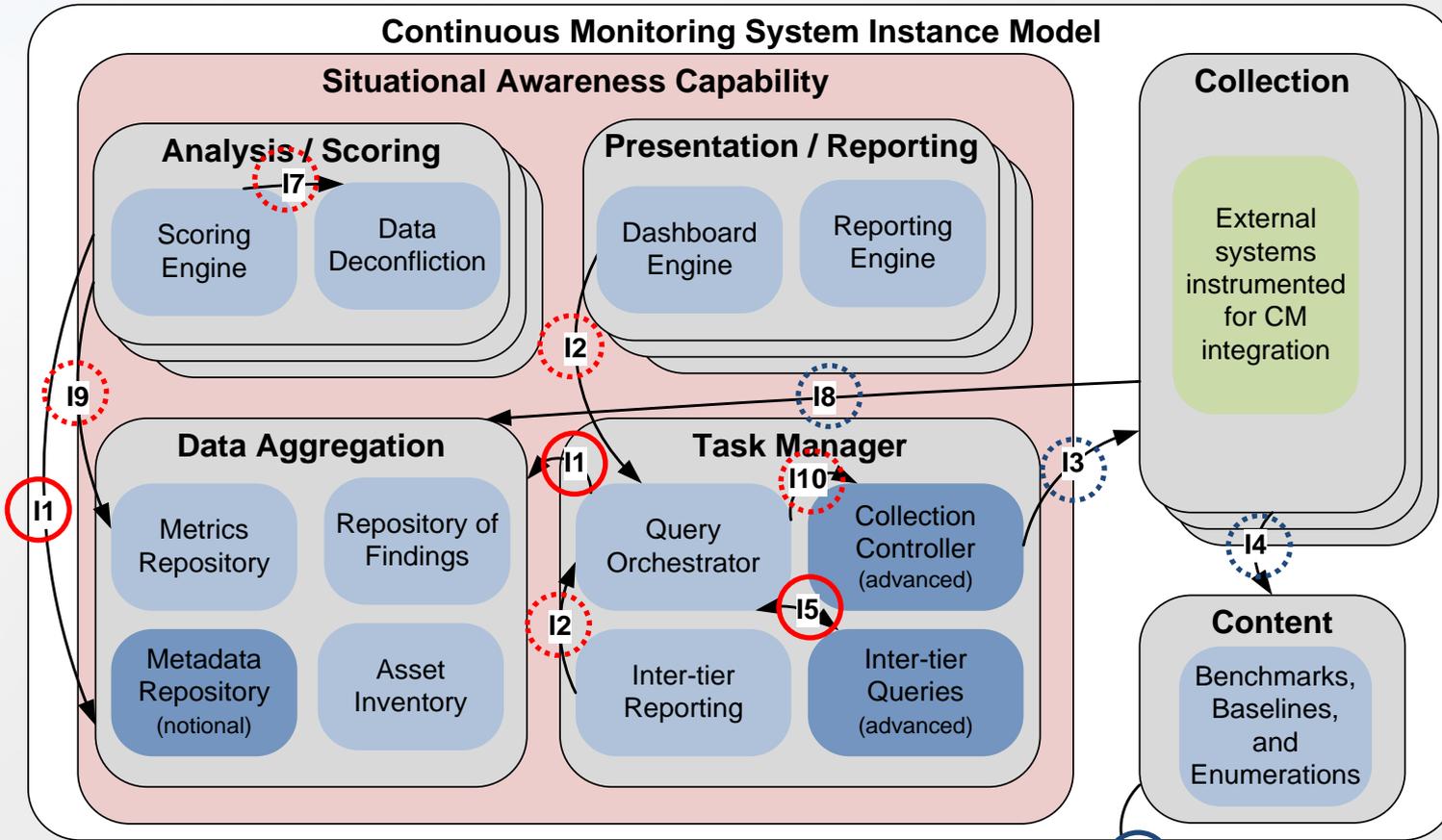


Section 4: Communication Pattern Level



- Interface Specifications
- Communication Models

CM Instance Interfaces



Interface and Payload Specifications:

Existing/ Standardized

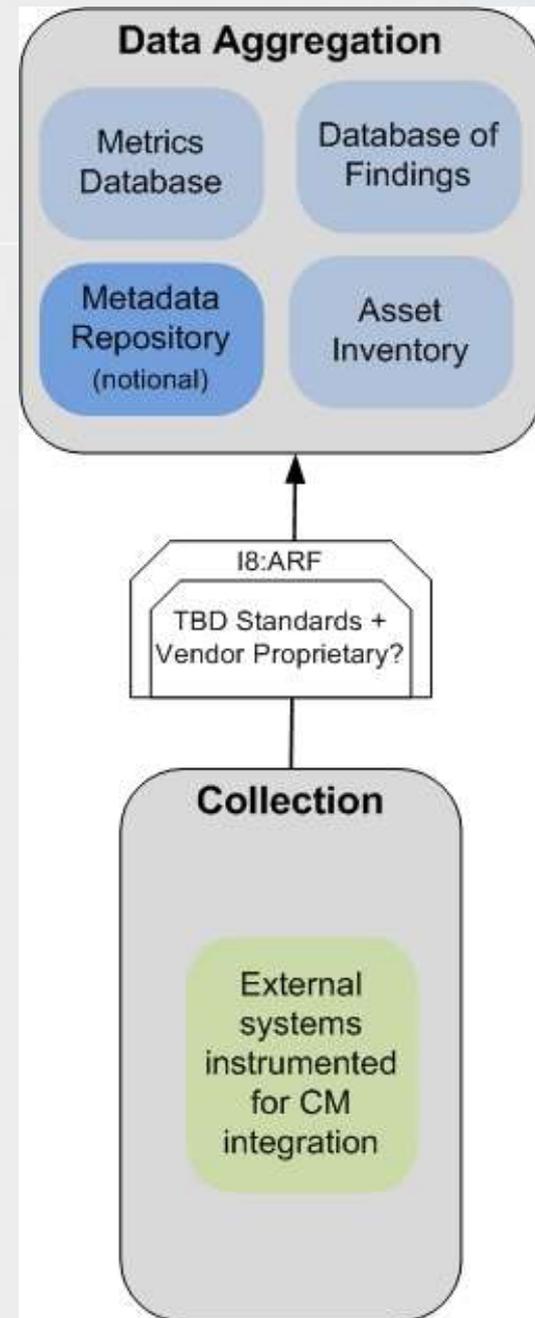
Current focus/ Standardized

Partial focus/ Proprietary

Future Focus/ Proprietary

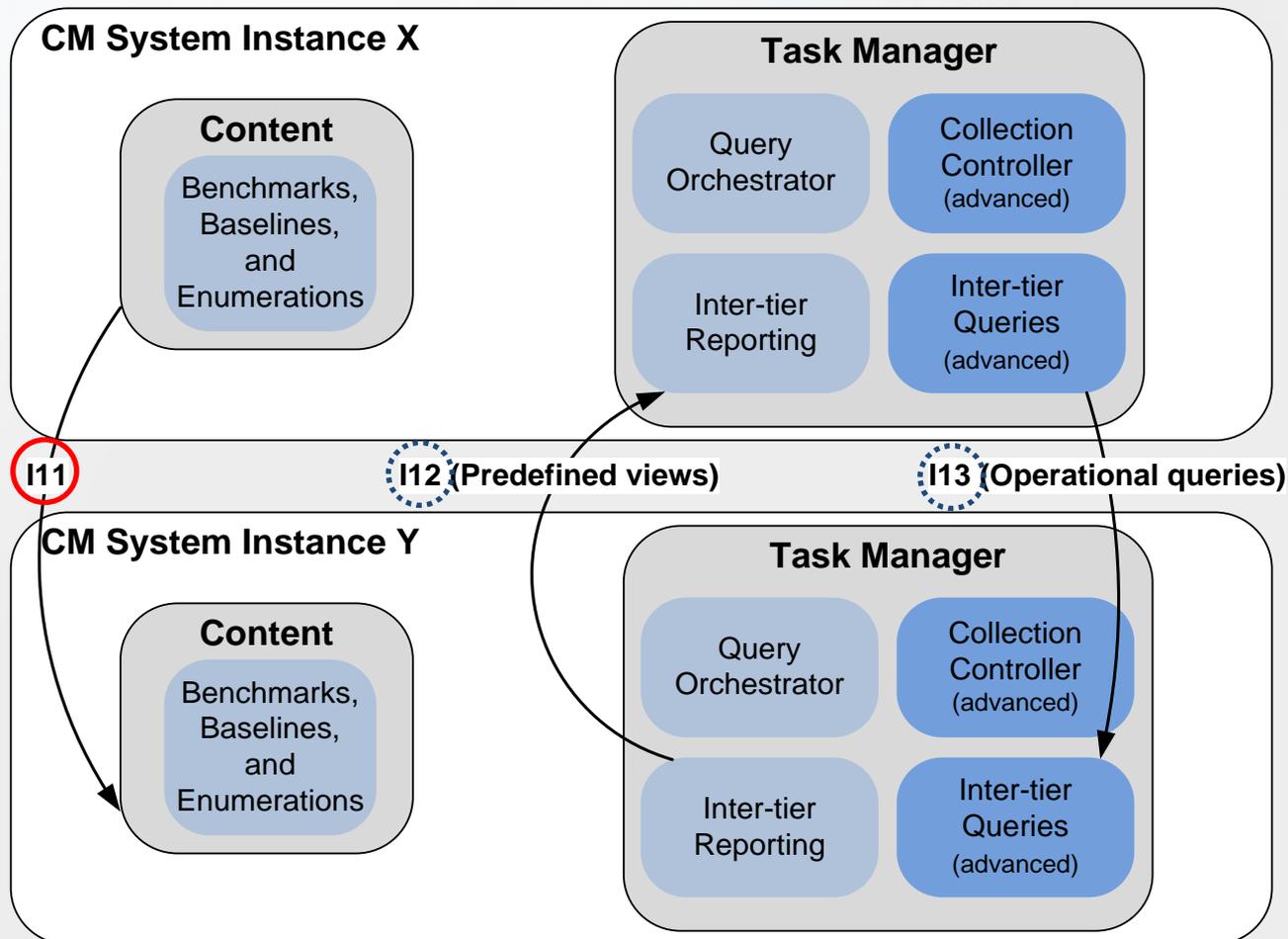
Notional Interface Overview: I8

- 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 collector 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??



Multi-instance CM Interfaces

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



Interface and Payload Specifications:

Existing/
Standardized

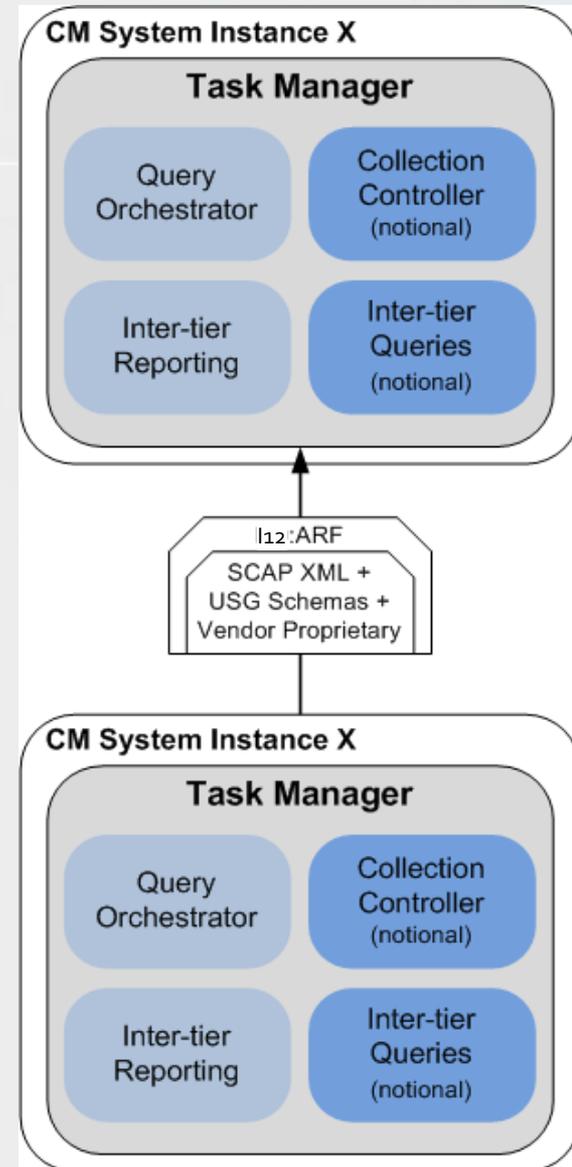
Current
focus/
Standardized

Partial
focus/
Proprietary

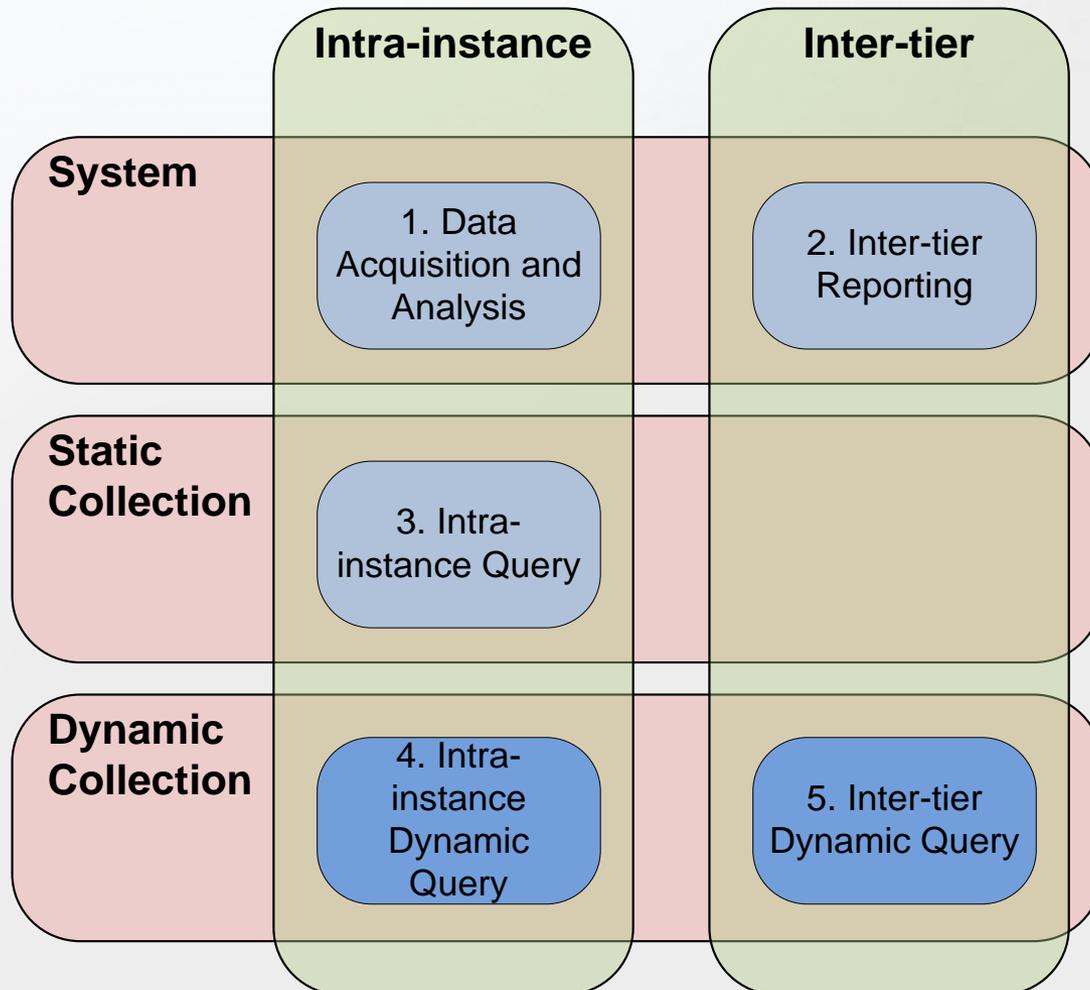
Future
Focus/
Proprietary

Notional Interface Overview: I12

- 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



CAESARS FE Abstract Use Cases and Workflow



Proposed Workflow:
Dynamic Scoring
Adjustment

Scope:
Dynamic
Intra-instance

Section 5: Specification Level



- How do we specify requirements to enable use of the architecture:
 - Product development
 - Procurement
 - Product validation

Section 6: CM Maturity Models



- How do we grow up?
- Transitioning to more effective approaches

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

Closing Thoughts

- There exists great momentum surrounding continuous monitoring (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 continuous monitoring solution
 - Enhance their own capability and meet upcoming reporting demands
- Reference models
 - 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 continuous monitoring
- 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) Continuous Monitoring working group
 - DHS Federal Network Security (Cyberscope and CAESARS)
 - NSA Information Assurance Directorate (IAD)
 - NIST Security Content Automation Protocol (SCAP) team
 - NIST Risk Management Framework (RMF) team
 - MITRE McLean CAESARS team
 - MITRE Bedford SCAP team

Summary and Questions



Presenter:

Peter Mell

NIST Senior Computer Scientist

301-975-5572

peter.mell@nist.gov

<http://twitter.com/petermmell>