An Enterprise Continuous Monitoring Technical Reference Architecture

12/14/2010

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

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

Our CM Architecture Design Team

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

David Minge
National Security Agency

Timothy McBride
Department of Homeland Security

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

Alfred Ouyang, Mark Crouter
MITRE

Continuous Monitoring (CM) Presentation Contents

- Section 1: Conceptual Design Level
  - Definition, Essential Characteristics, and Enterprise Architecture
- Section 2: Technical Design Level
  - Subcomponent Model
  - Technical Architecture
- Section 3: Implementation Design Level
  - Subsystem communication
  - Interfaces
- Section 4: CM Maturity Models

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 Patterns
                - Functional specifications

Section 1: Conceptual Design Level

- CM Definitions
- Essential Characteristics
- Enterprise Architecture

General CM Definition

Continuous monitoring is the on-going observance with the intent to provide warning. A continuous monitoring capability is the on-going 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 could support

- Asset Management
- Configuration Management
- Content Management
- Event Management
- Incident Management
- Information Management
- License Management
- Malware Detection
- Network Management
- Patch Management
- Risk Management
- Software Assurance
- Trouble Ticketing Management
- Vulnerability Management

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 quantify risk, ensure effectiveness of security controls, and enable prioritization of remedies.

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

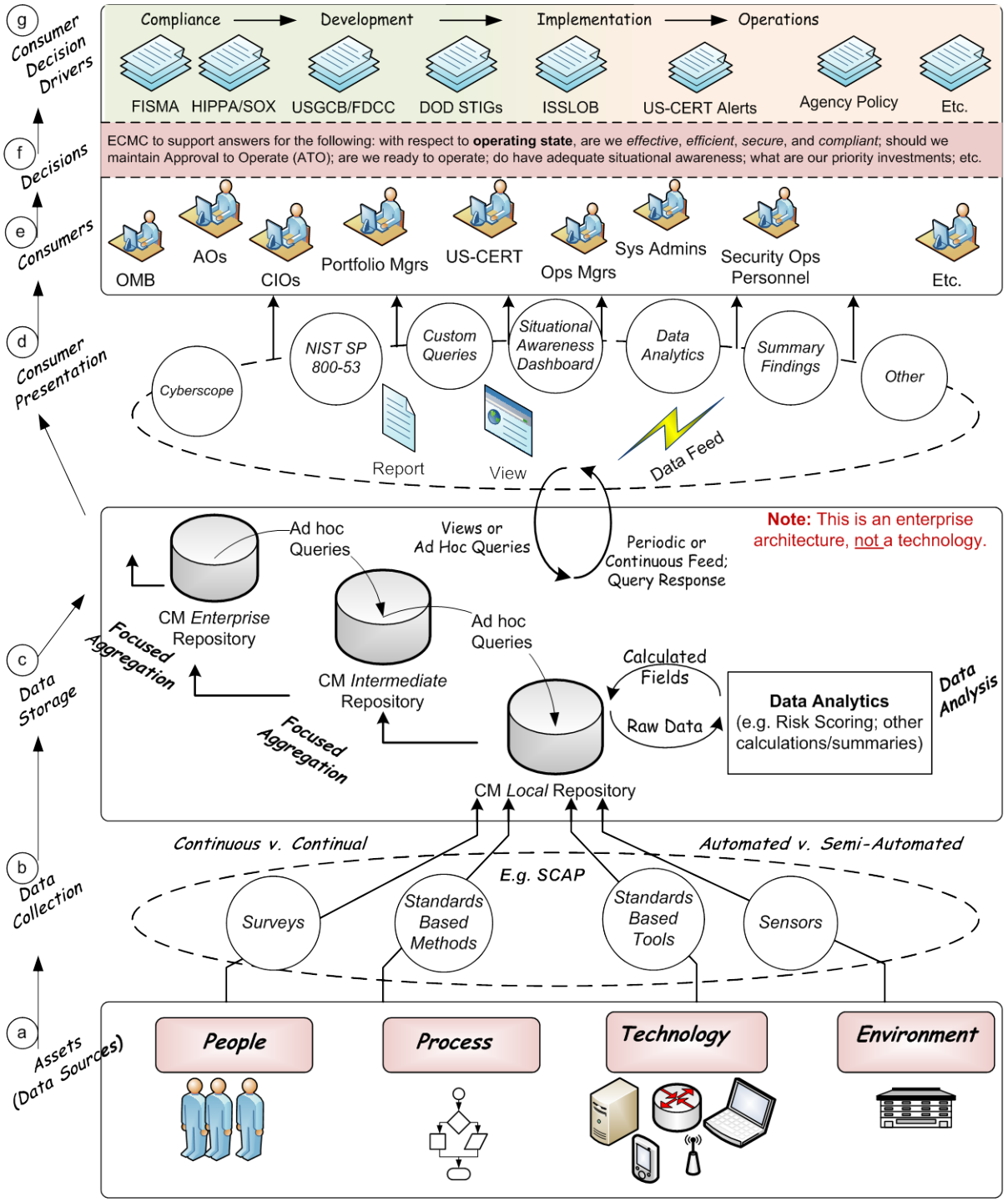
Derived CM Characteristics:

- 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

CM Enterprise Architecture

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

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



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)
  - Leverage your existing security products
  - Reduce integration costs
  - Combine best of breed solutions

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

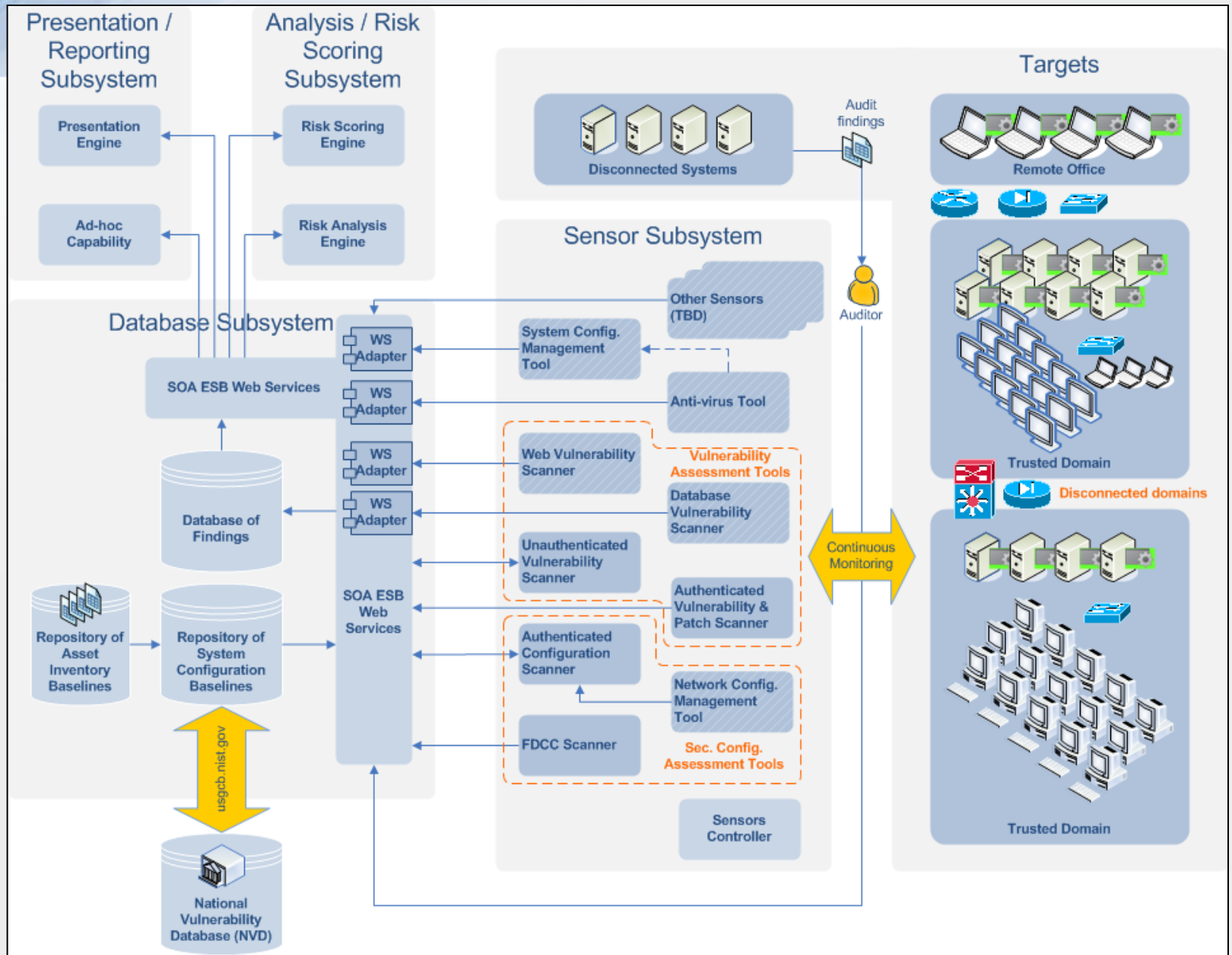
Section 2: Technical Architecture Design Level

- Subsystem Design
- Technical 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 CM 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

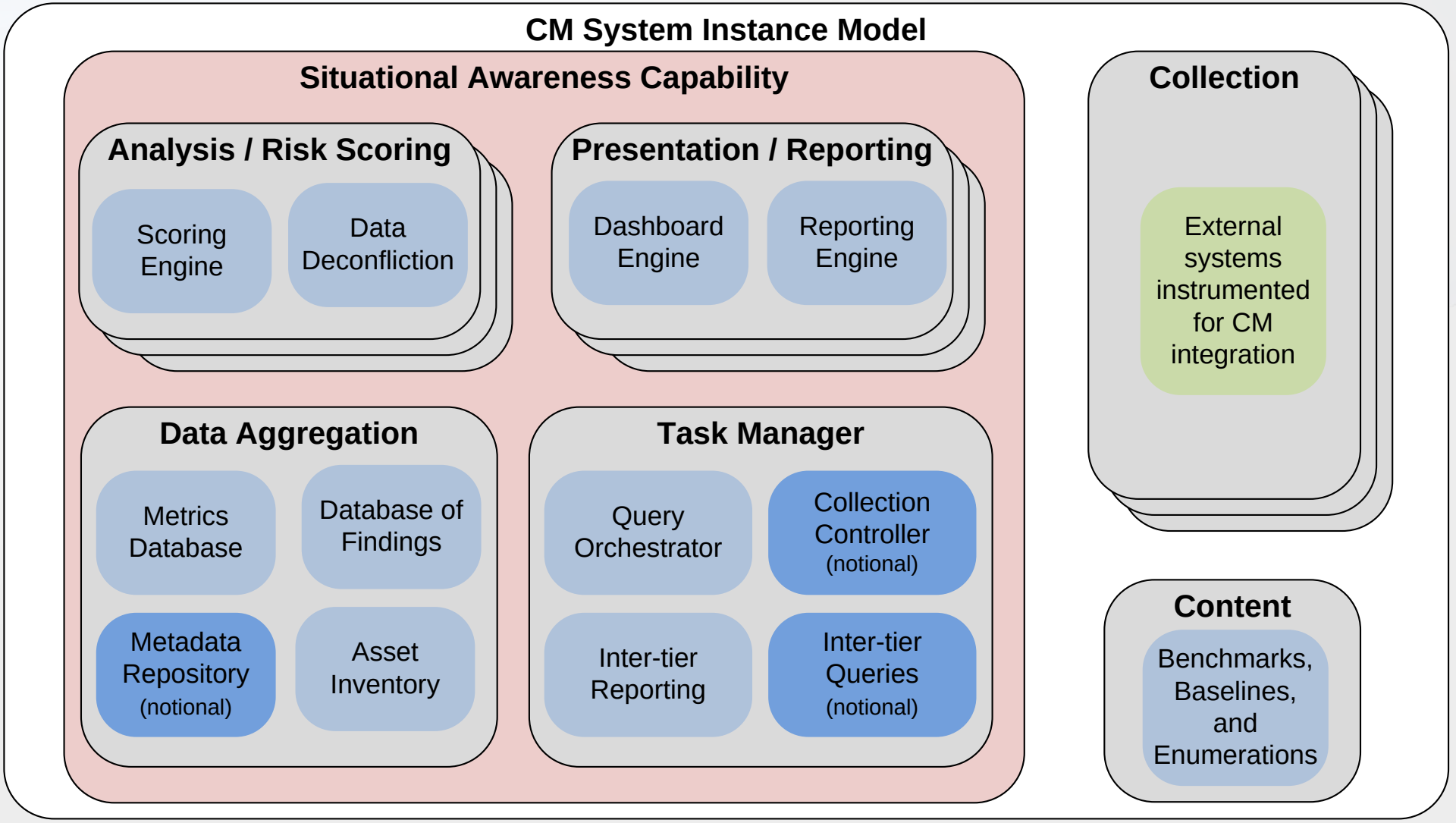
CAESARS is a good foundation. We need to expand upon its framework to address the limitations and add additional capabilities

Notional CAESARS Framework Expansion

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

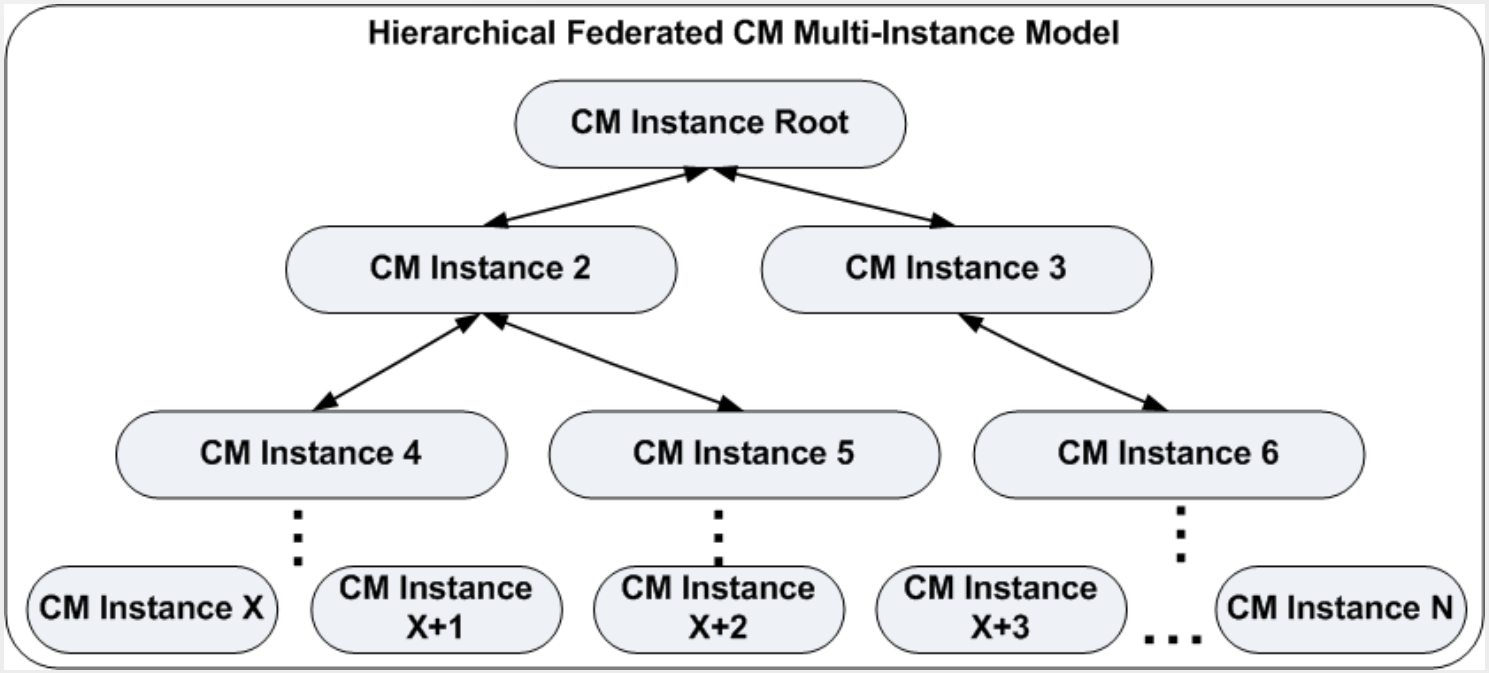
Continuous Monitoring Instance Model

(Organizations may have multiple CM instances)

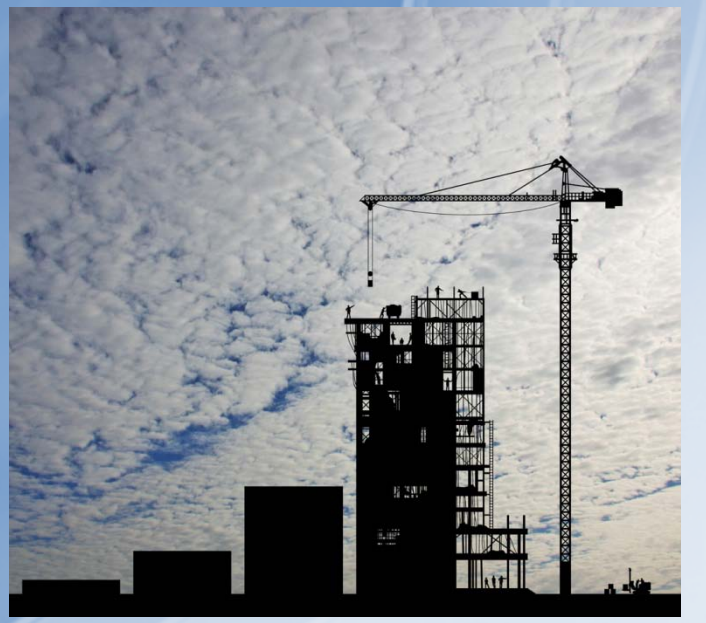


Hierarchical Federated Architecture

- 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

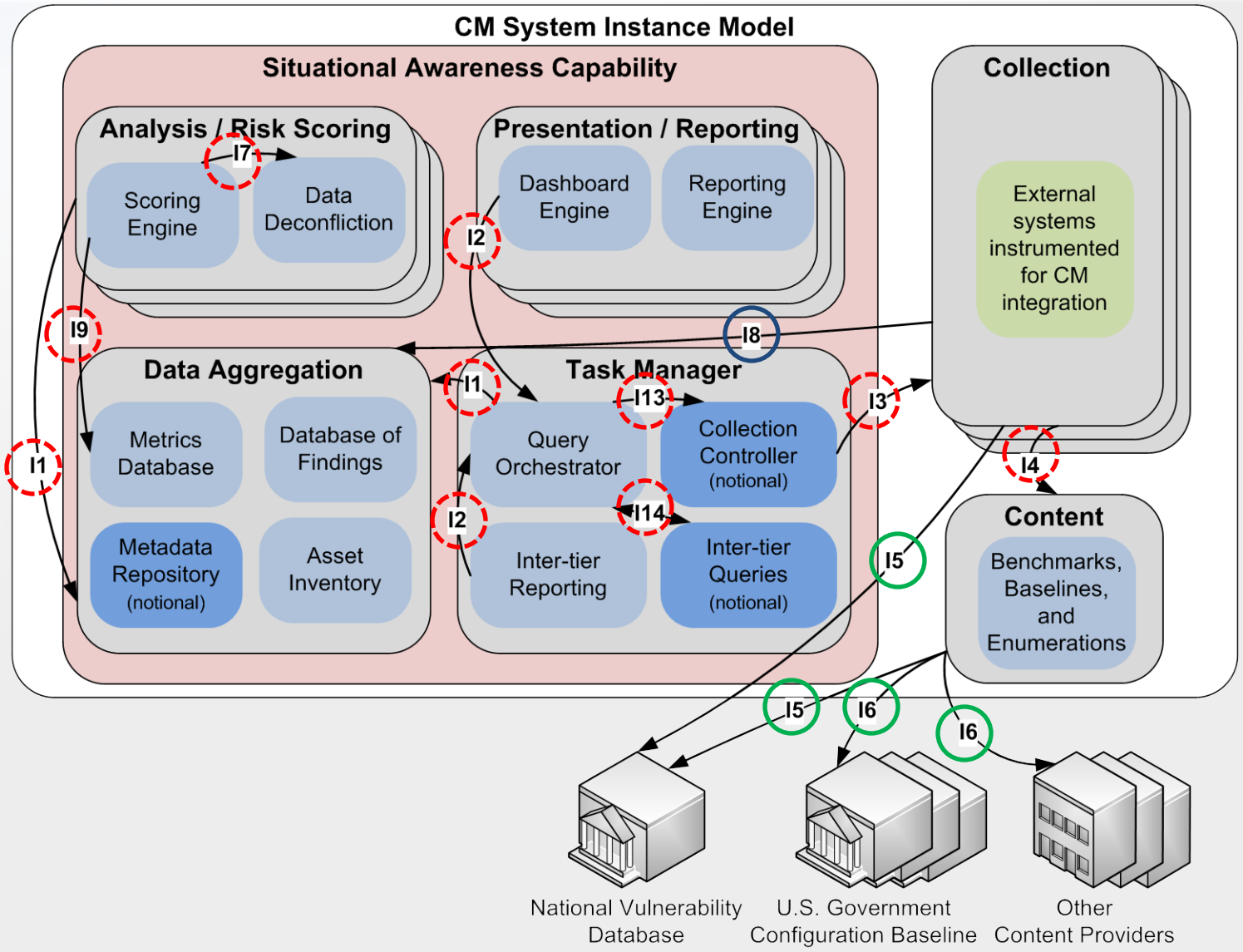


Section 3: Implementation Design Level



- Interface Specifications
- Communication Models
- Derived Test Requirements

CM Instance Interfaces

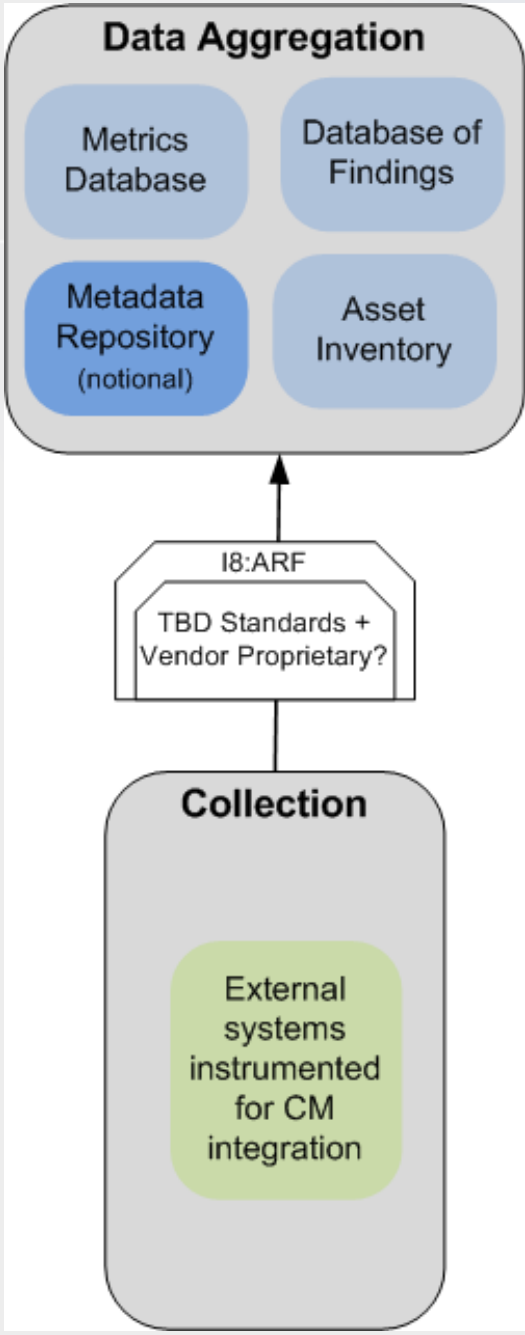


Interface and Payload Specifications:

- Existing
- Current Focus
- Proprietary/ Future Focus

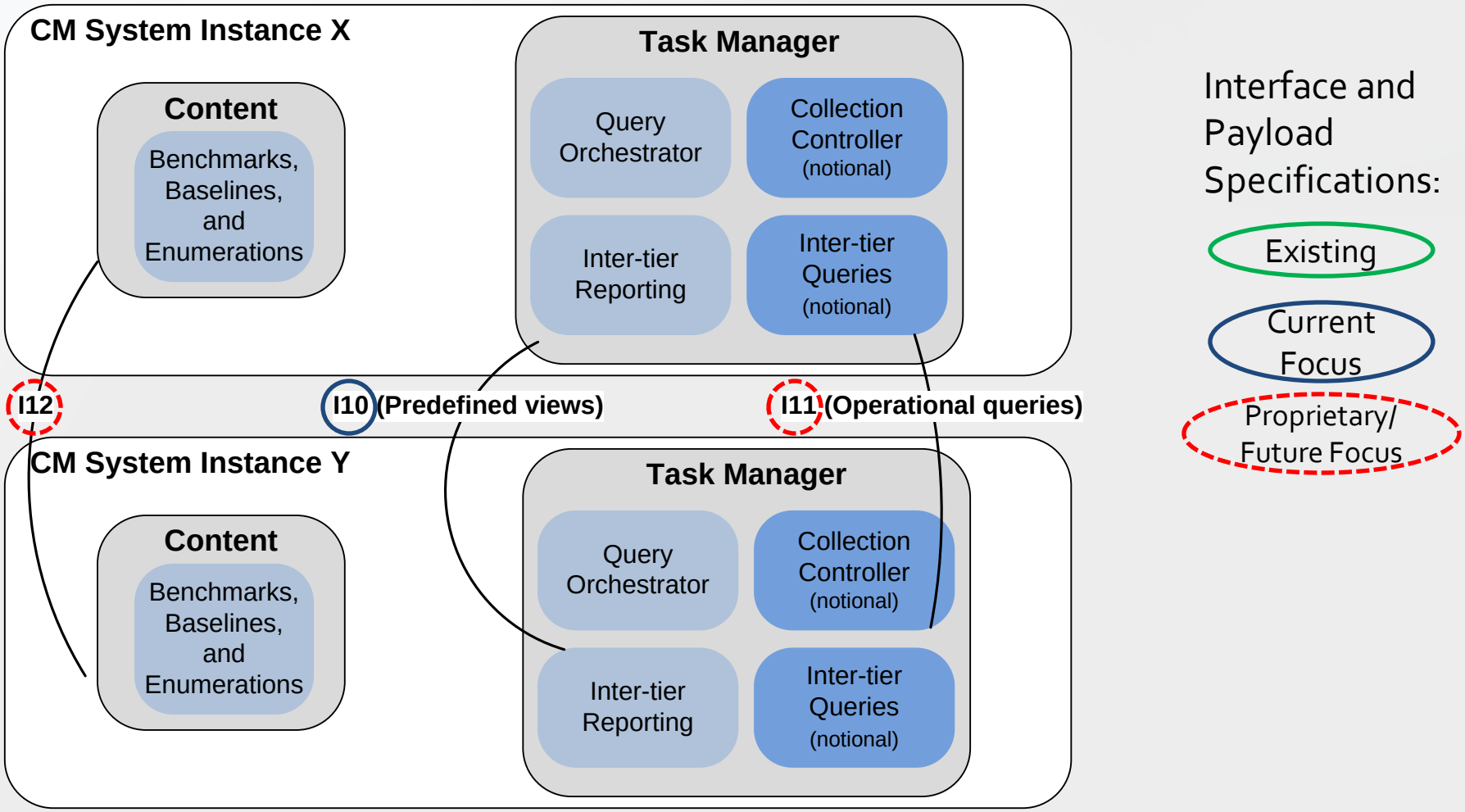
Notional Interface Overview: I8

- 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 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??



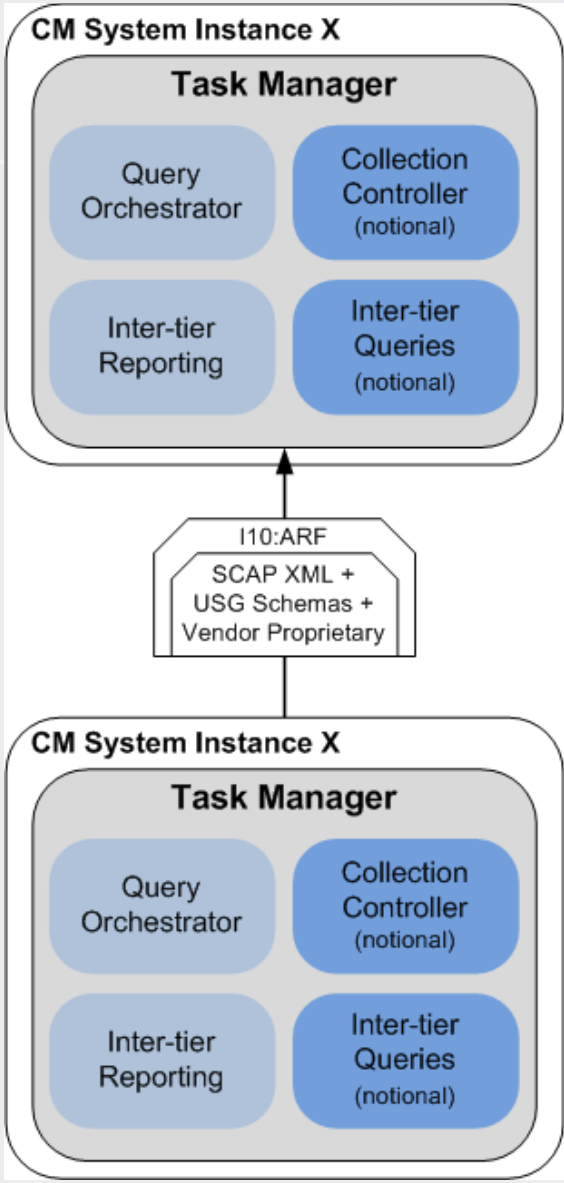
Multi-instance CM Interfaces

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



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

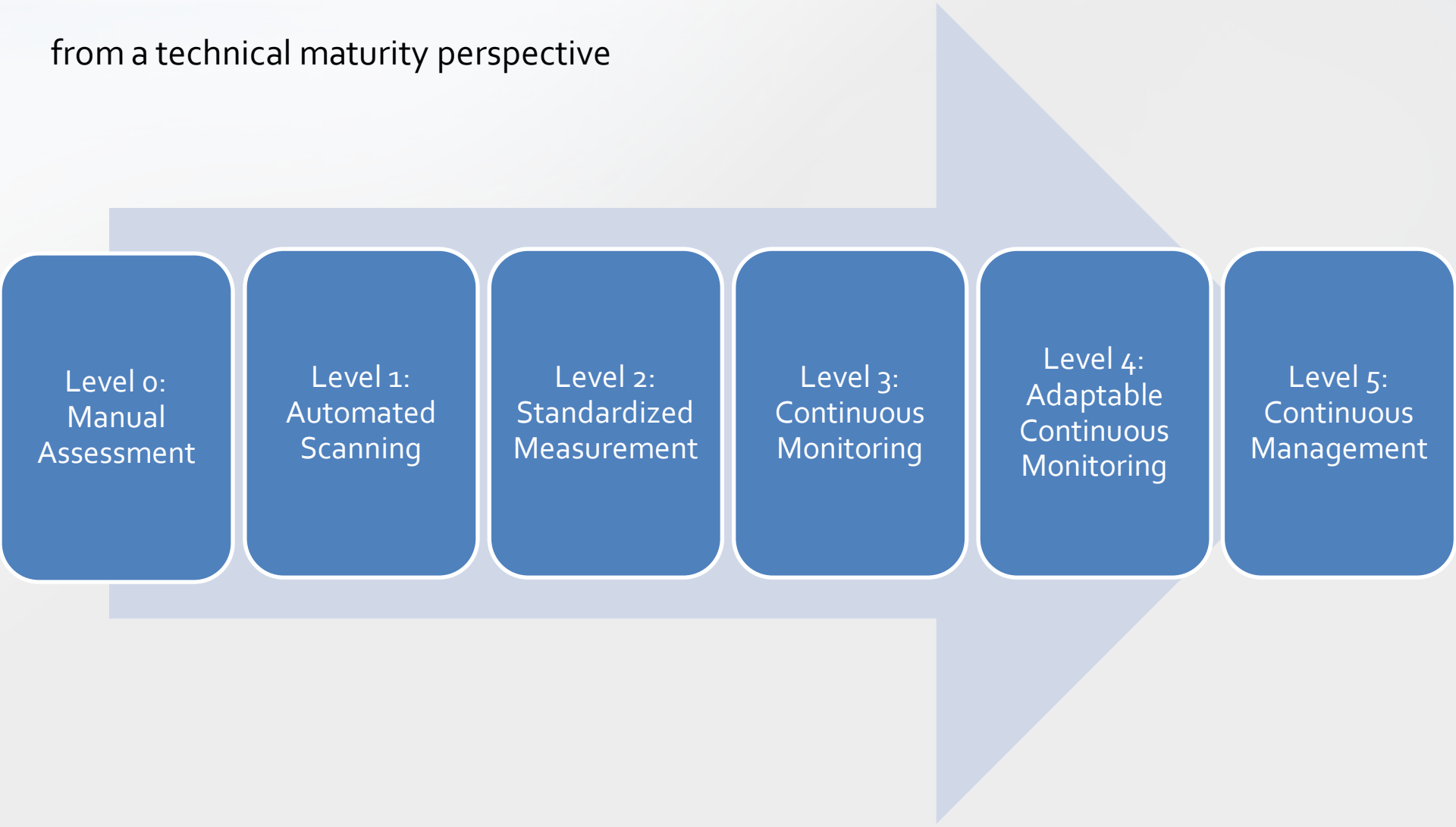


Section 4: CM Maturity Models

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

Notional Maturity Model for Continuous Monitoring

from a technical maturity perspective



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 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

Presenter:

Peter Mell
NIST Senior Computer Scientist
301-975-5572
peter.mell@nist.gov
http://twitter.com/petermmell