SACM Architecture

draft-waltermire-sacm-architecture-00
Overall Architectural Philosophy

- Leverage existing IETF and other international standards where possible
  - Leverage existing Layer 1 – 6 specifications
  - Profile or extend existing application (Layer 7) specifications
- Minimize the number of required data flows/interfaces
- Keep it simple
Data Flows / Interfaces

- DF1: Content Retrieval
  - Used to acquire content to drive data collection and analysis

- DF2: Collection Tasking
  - Used to orchestrate required data collection

- DF3: Collected Data Publication
  - Used to publish collected data to appropriate data store(s)

- DF4: Collected Data Query
  - Used to query previously collected data for analysis
Sensors
Endpoint Sensor Data – Current Scope

• Endpoint Identity
• Posture attributes
  – Hardware Inventory
  – Software Inventory
    • Operating System
    • Applications
    • Patches
  – Software Configurations
• Provenance data
  – Identification of sensor
  – Other sensor metadata/context
• Entailment information
  – Collection methods
  – Additional context
Content Repositories

• Provided standardized, secure access to data that drives:
  – Collection policies (e.g. what, when, where)
  – Data collection activities (e.g. what)
  – Analysis activities (e.g. what)

• Data agnostic, resource identification approach
Data Storage

- Enables collected and analyzed data to be persisted
- Provides standardized, secure methods to publish and retrieve data
- Decouples dependencies between data collectors and analysis components providing architectural flexibility
Controller

- Responsible for securely orchestrating data collection by sensors
- May manage other controllers
- Enables evaluators to request on-demand or scheduled data collection
Evaluator

- Performs analysis based on previously collected data
- Securely interacts with controllers to orchestrate needed data collection
Open Questions

• Sensors
  – Are network-oriented sensors in scope?
  – Are external endpoint-oriented sensors in scope?

• Controllers
  – Should controllers manage other controllers?
  – Should controllers manage content repositories or data storage?