XAM (eXtensible Access Method)

In eDiscovery and Litigation Support Repositories

MPeterson, June 2009

SNIA

XAM Initiative

Contents

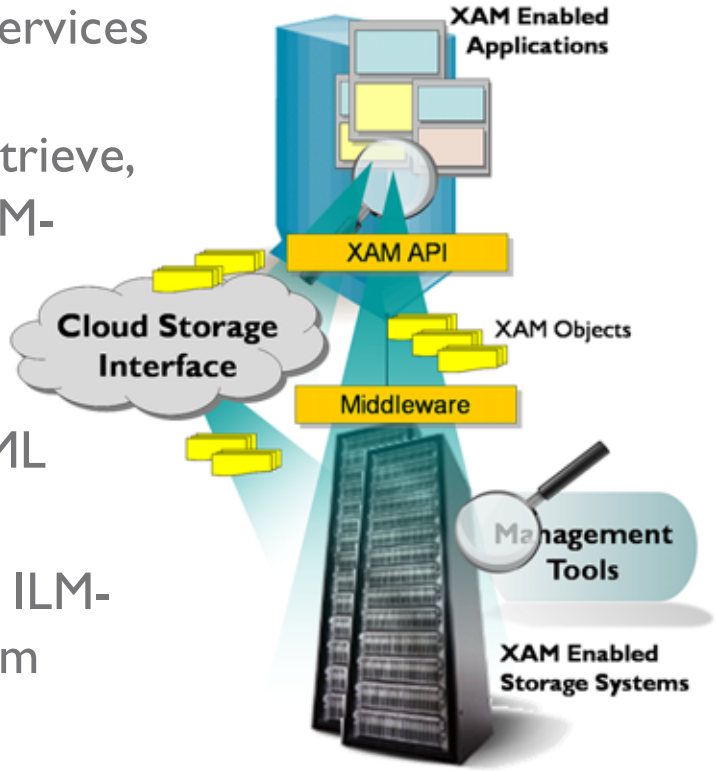
SNIA
Advancing storage & information technology

- XAM Introduction
- XAM Value Propositions
- XAM Use Cases
  - Digital Libraries and Preservation Stores
  - eDiscovery
  - Medical Records, Electronic Healthcare Records
  - Cloud-services
- Reference Materials

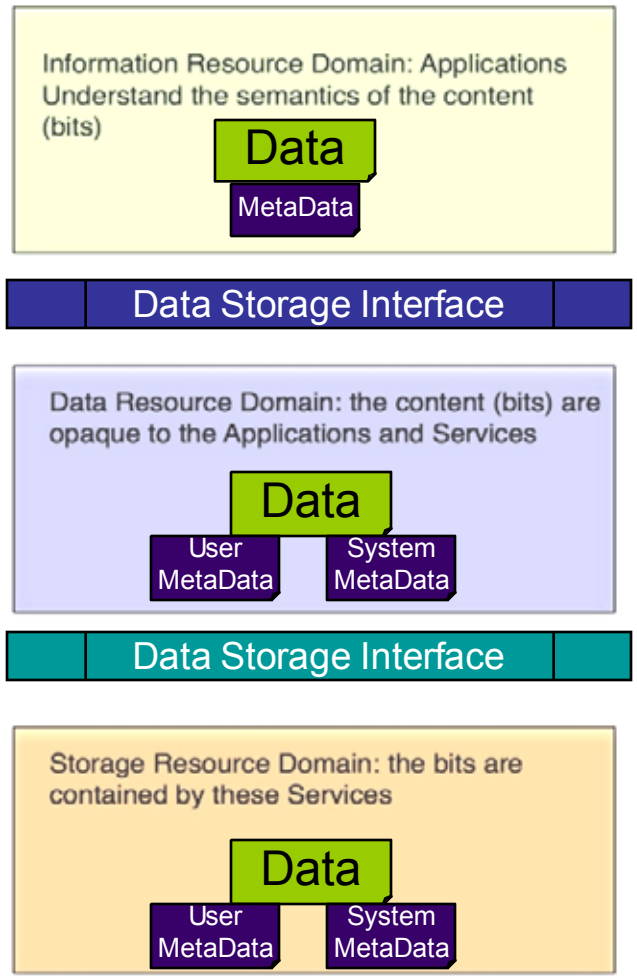
XAM Introduction

SNIA
Advancing storage & information technology

- The open XAM API specification
  - Defines the programming interface and services that enable applications and information management services to define, store, retrieve, search, and manage XAM objects on XAM-enabled storage systems
- XAM objects are:
  - Exportable or importable as standard XML containers
  - Provide for extensible metadata enabling ILM-based management, eDiscovery, long-term retention and preservation
  - Portable, location independent, compliant, secure



XAM is a Data Storage Interface



Resource Domains classify services into specific areas that each deal with a different aspect of the problem

An information domain application creates data and associates metadata with it

Certain Data Storage Interfaces can accommodate both Data and MetaData (XAM, Filesystems with extended attributes)

MetaData aware Data Services interpret Data System MetaData as the requirements for it's lifecycle and implement policies for retention, placement, lifecycle, etc.

Other Data Storage interfaces (based on blocks or objects) provide virtualized Containers for the Data bits and the management of those containers

Storage services are employed to meet those requirements at this point in the data's lifecycle, however the storage services are unaware of the data's requirements

XAM Value Proposition

SNIA
Advancing storage & information technology

- XAM provides choice, flexibility, and makes information portable
  - Choice between storage devices and applications, flexibility to change vendors or use storage devices from multiple vendors
- XAM reduces costs
  - Reduces application porting, testing, and support costs, and improves utilization efficiencies & time to market
- XAM empowers ILM-based management practices
  - Retention, disposition, security, and other requirements can be embedded in the metadata allowing application-based and external services to act on XAM objects automatically based on business rules
  - Reduces cost and improves storage utilization efficiencies

XAM Value Propositions

SNIA
Advancing storage & information technology

- XAM metadata is the foundation for extended e-Discovery, analytics, and litigation support capabilities
- A standard container enabling centralized, shareable, and controlled medical and health care records
- Retention and Preservation systems utilizing XAM provide a standard preservation object format
- XAM provides cloud-based services with missing information management control capabilities

eDiscovery and Litigation Support

XAM Value Proposition

May 2009

eDiscovery and Litigation Support

SNIA
Advancing storage & information technology

- Top Challenges
  - Identification and control of information and data assets and their locations
  - Scaling to keep up with growth of capacity and number of cases
  - Litigation hold policy control
  - Avoid 'spoliating' discovered information
    - Including control of metadata
  - Controlling discovered information for long-term periods of time while minimizing cost

If you have a litigation that runs 7-10 years, you will have at least 3 migration events and will lose information and data

eDiscovery and Litigation Support

SNIA
Advancing storage & information technology

Vision

- Discovery and litigation review practices are more scalable, cost effective, automated, and controllable across time
- XAM features, benefits, and attributes
  - XAM objects are self-contained, portable, location independent preservation objects that can be searched and discovered
  - All critical preservation and litigation control services are supported – authenticity, integrity, availability, discoverability, portability, confidentiality, audit logs, ILM-base management, migration, …
  - Litigation hold controls can be embedded in each object
  - Litigation review results can be embedded in each object
  - Migration and management costs are greatly reduced

How XAM Applies

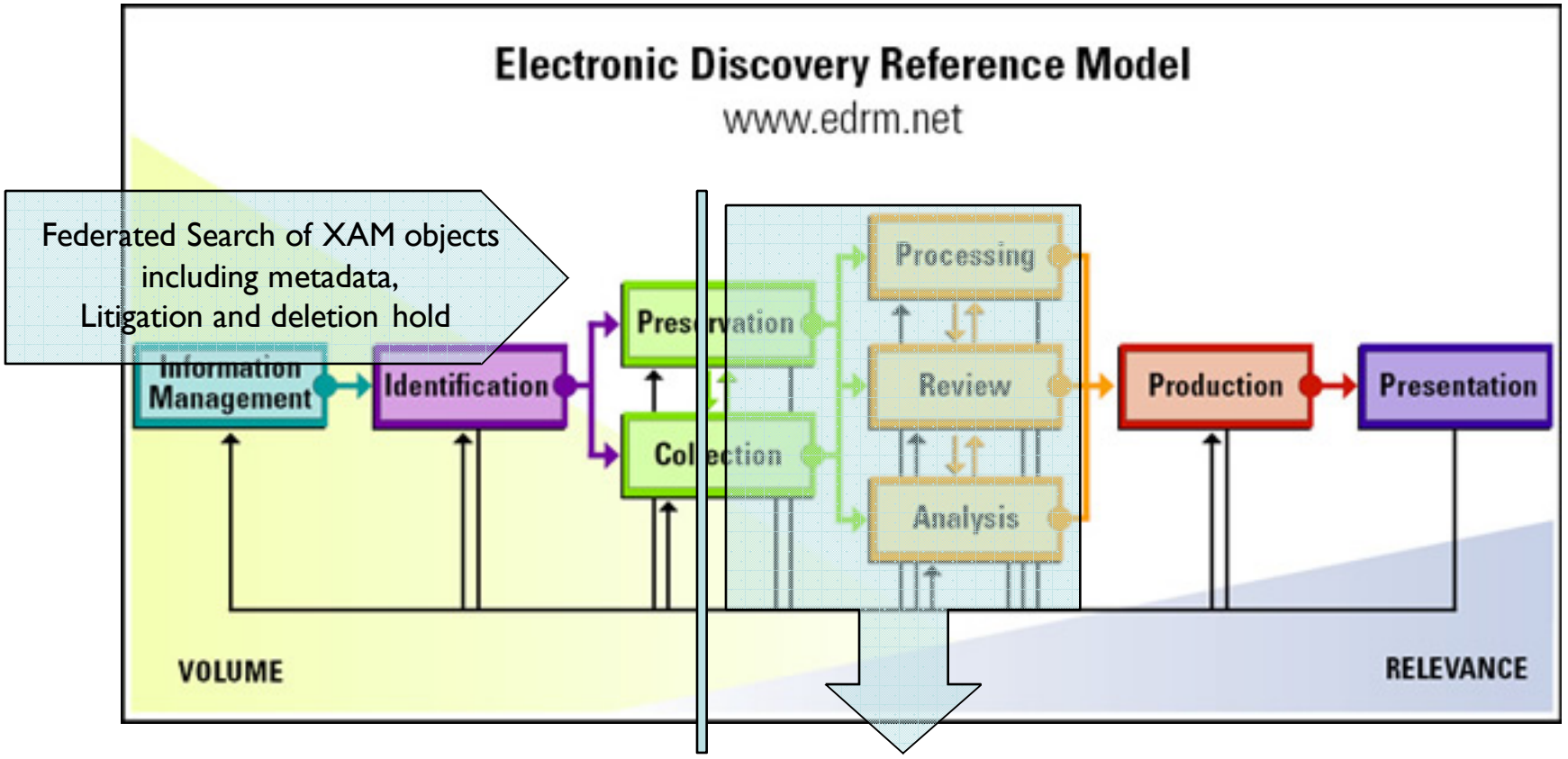
SNIA
Advancing storage & information technology

- XAM defines an updateable preservation object – it is the object container, the access method, and the application to storage interface
  - Supports federated search including expanded metadata and controls
  - Can control litigation holds
  - During litigation review, review results can be placed in each object
  - XAM objects are portable, storage and location independent reducing migration costs

Vision:

XAM objects provide preservation, litigation, and discovery controls supporting cost-effective retention, deletion, and litigation support practices in the datacenter.

XAM and EDRM



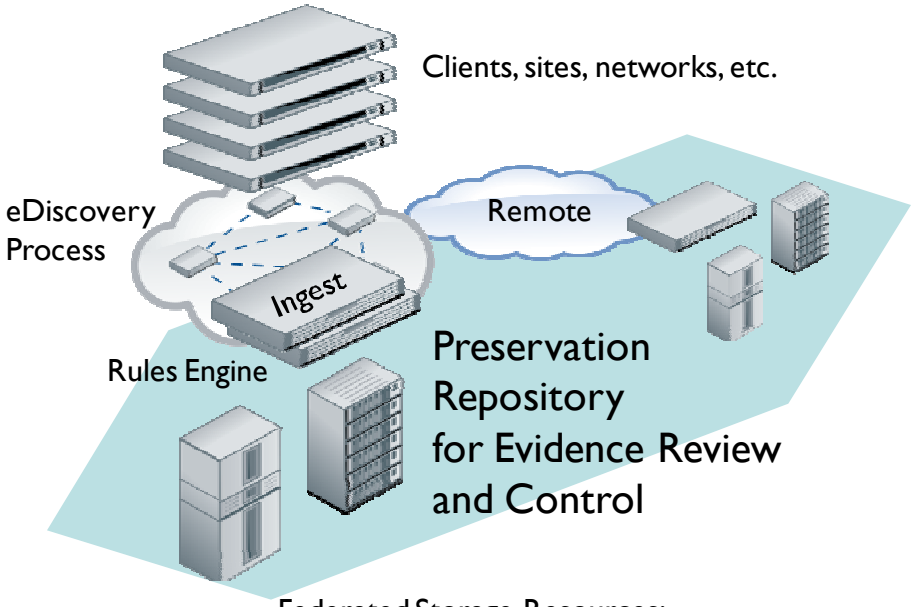
Ingest, Index,
Create XAM objects

Litigation Review Repository:
Utilize XAM to preserve and protect from change, control authenticity, to record review/analysis results in the object metadata, etc.

XAM Use Case

- A litigation support service runs eDiscovery and ingests discovered content
  - During ingestion, XAM objects are created with metadata based on policies
  - Objects are hashed, audit logs started, indexed for discovery, and storage policies initiated based on classification rules
  - Preservation services are now operating and in control

Typical Litigation Support Preservation Store



Federated Storage Resources:
- Tiered and Virtualized
- Distributed if needed for business continuity
- Multiple vendors
- Multiple architectures

XAM Value Proposition

SNIA
Advancing storage & information technology

- XAM metadata and object containers create the foundation for extended e-Discovery, analytics, and litigation support capabilities
  - Rich XAM metadata can be searched and analyzed
  - Extensible XAM container provides for security, integrity, authenticity, audit logs, chain of custody and litigation hold controls, and can hold litigation review metadata
  - XAM provides the capability of preservation services that will mitigate the cost, complexity, and potential spoliation caused by logical and physical migrations

SNIA and XAM

SNIA
Advancing storage & information technology

- For more information on XAM
  - SNIA XAM Home
    - http://www.snia.org/xam
- Contact information:
  - Michael Peterson, SNIA Chief Strategy Advocate
    (805) 201-3178 or mpeterson@snia-dmf.org

XAM BRIEFING

MP1

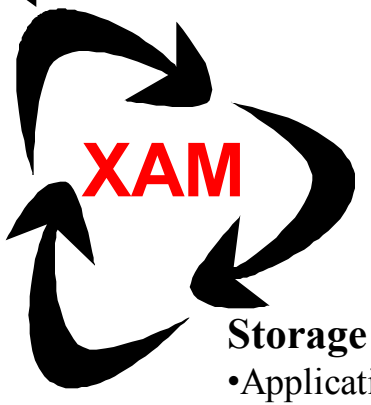
Why XAM?

SNIA
Advancing storage & information technology

The industry will benefit from a standardized access method to storage systems

End Users want:
- Choices between Application Vendors
- Choices between Storage Vendors
- Easy migration between vendors/technology
- Compliance, Scalability, Performance, $/GB, TCO

Application Vendors want:
- Annotate Data with associated Metadata
- Indicate basic Storage Management Policies
- Speak same language to all types of Devices
- Manipulate billions if not trillions or "records"



Storage Vendors want:
- Application Support for their Products
- Efficiently Store Application Data and Metadata
- Integrate Basic Storage Management Capabilities
- Manage billions if not trillions of "records"

Slide 16

MP1 Needs updating with the new value props
Michael Peterson, 5/13/2009

A New Storage API

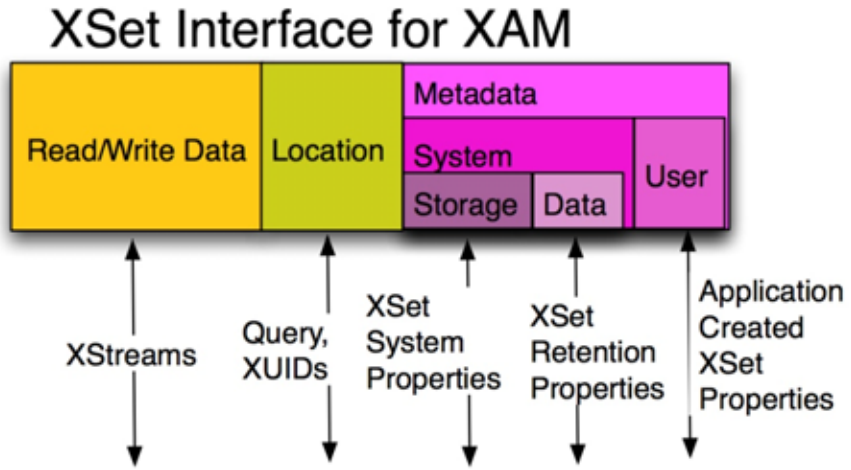
SNIA
Advancing storage & information technology

- XAM Provides:
  - Interoperability: Applications can work with any XAM conformant storage system; information can be migrated and shared
  - Compliance: Integrated record retention and disposition metadata,
  - ILM Practices: Framework for classification, policy, and implementation
  - Migration: Ability to automate migration process to maintain long-term readability
  - Discovery: Application-independent structured discovery avoids application obsolescence.
- XAM is a SNIA Architecture
  - The XAM Architecture spec defines the normative semantics of the API for use by applications and implementation by storage systems
- XAM is an Application Programming Interface (API)
  - The XAM Java and C API specs define bindings of the XAM Architecture to the Java and C Languages
- XAM is SNIA Software – open source
  - The XAM SDK provides a common library and reference implementation to promote widespread adoption of the standard

XAM Interface

- First interface to standardize system metadata for retention of data
- Implements basic capability to Read and Write Data (through Xstreams)
- Locates any XSet with a query or by supplying the XUID
- Allows Metadata to be added to the data and keeps both in an XSet object
- Uses and produces system metadata for each Xset
- For example Access and Commit times (Storage System Metadata)
- But it also uniquely specifies Data System Metadata for Retention Data Services

- XAM User metadata is un-interpretable by the system, but stored with the other data and is available for use in queries
- Given this we can see that XAM is a data storage interface that is used by both Storage and Data Services (functions)



XAM System Metadata

SNIA
Advancing storage & information technology

- XAM specifies *property fields* that are interpreted by an Xsystem as System Metadata
- It does this by using a reserved field namespace
  - .xam.*, .xsystem.* and .xset.* are reserved
- Either the Content Application or a separate Information Management application provides preservation and/or ILM metadata to cause the underlying system to treat the data according to the specified requirements
  - i.e. Retention

Data Storage Interface Standards

SNIA
Advancing storage & information technology

- Other standard data storage APIs have the ability to deal with metadata as well (POSIX filesystems)
- POSIX specifies standard system metadata as part of the data storage interface:
  - File times, Permission (including ACLs), owner, group, etc.
- This metadata is maintained and used (interpreted) by the storage services that implement the API
  - Thus we call it ***storage system metadata***
- The functions that are controlled by this metadata govern the storing and retrieval of the data through the interface
- These functions are described in the abstract as ***storage services***

Metadata

Storage services may provide functions for metadata as part of the data storage interface.

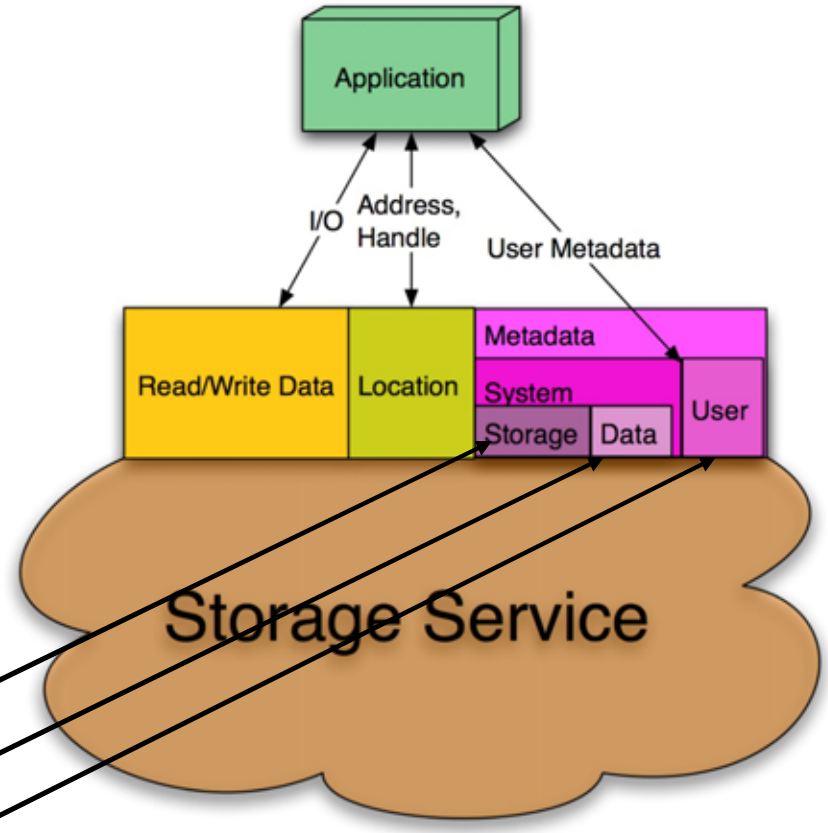
- This is an important capability for managing Data Resources (as opposed to managing Storage Resources).

The metadata may be managed by the storage service, managed by data services, or un-interpreted by either.

- System metadata that is managed by storage services are those properties of a data element that pertain to the primary functions of storing and retrieving the data.

We call this *storage system metadata*, as it is used and managed by storage services.

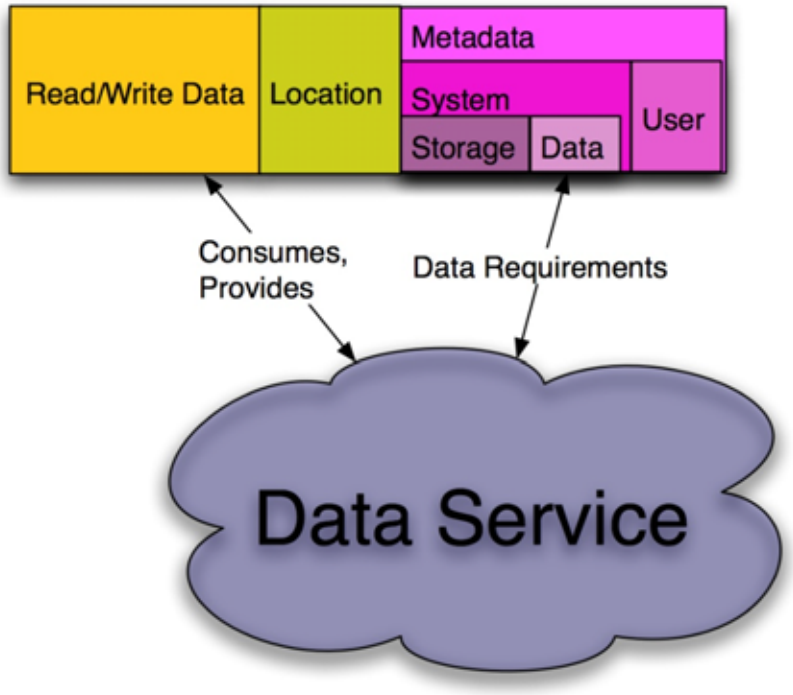
- Other system and user metadata may be preserved on the basis of individual data elements, but is not interpreted by the storage services



Metadata for Data Services

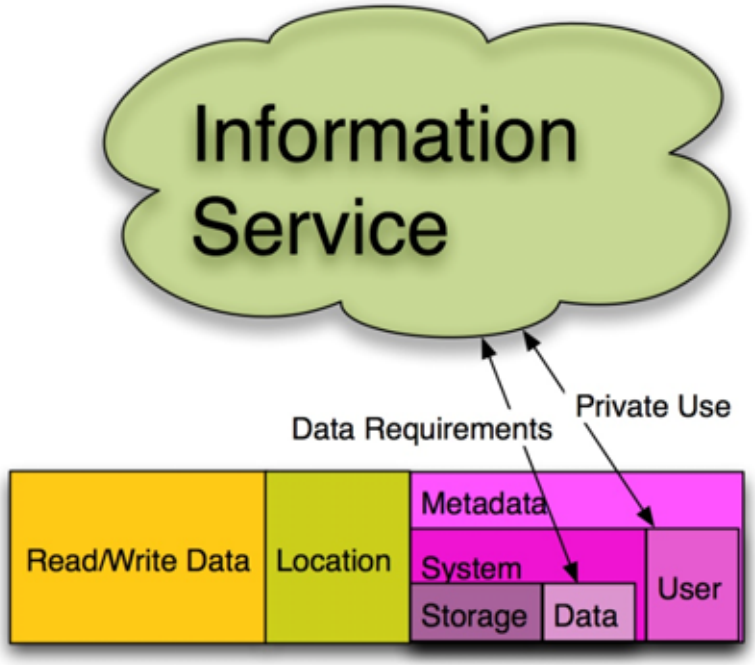
SNIA
Advancing storage & information technology

- Metadata available through the data storage interface may also be managed by data services
- This data service metadata can be used by data services to provide differentiated value to individual data elements
- The model or schema for data service metadata may be defined by each data service and may be standardized



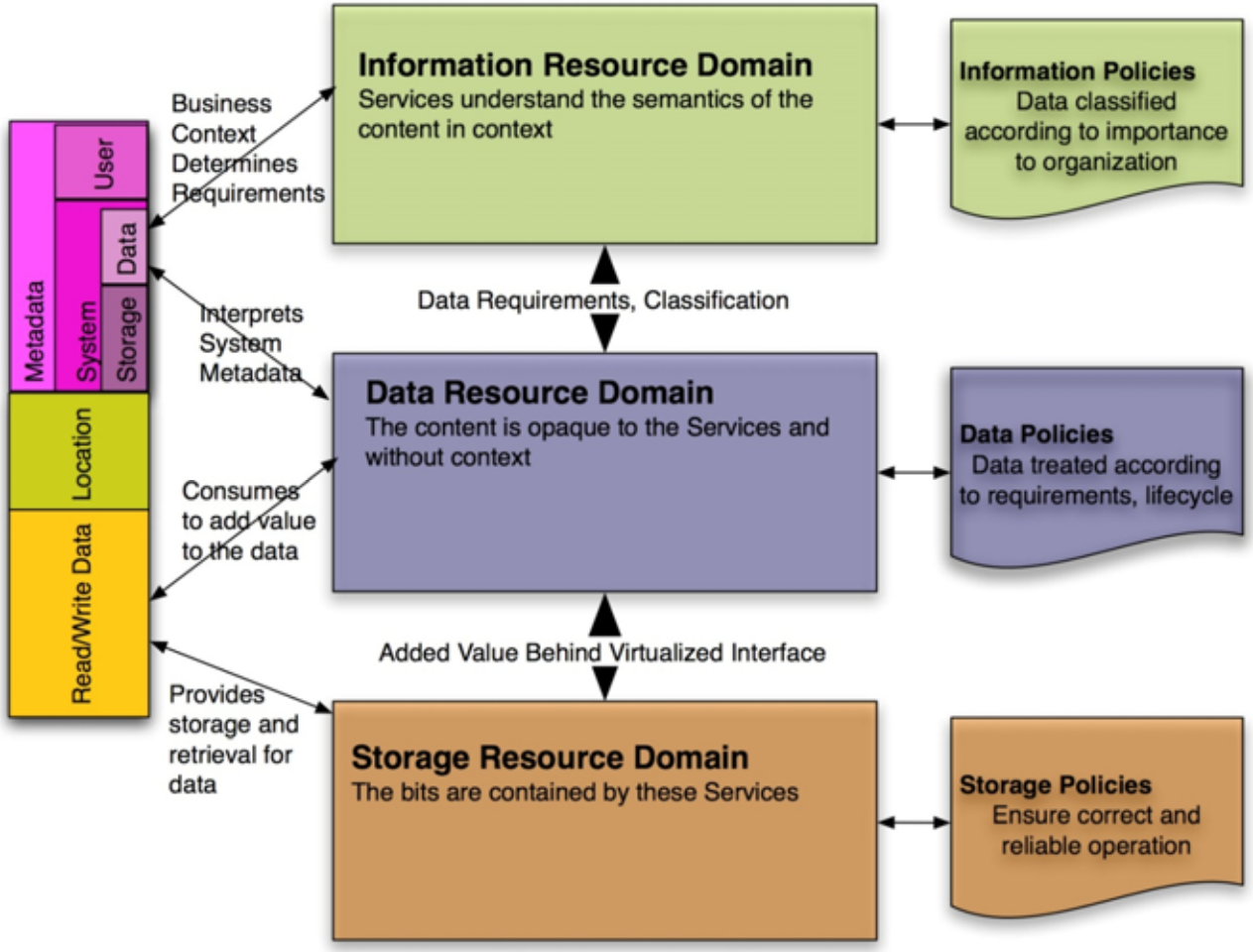
Metadata in Information Services

- The role of metadata in information services is as a communication mechanism with the underlying storage services and data services.
- Information services are primarily concerned with the data service system metadata as a means to convey the data's requirements to the underlying data services.
- An information service may also interpret user metadata for purposes of data classification.
- An information service can create its own user metadata that is un-interpreted by the underlying services for its own use.

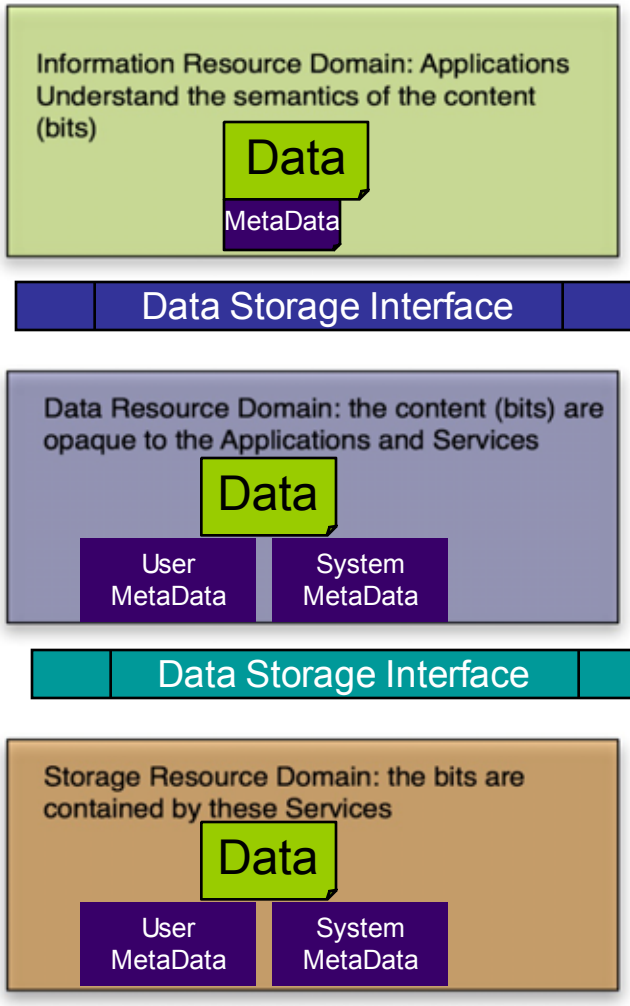


The Resource Domain Model

- This model shows the logical layering of the different domains and the role of policies for each domain. The services in each domain play a different role, but leverage common, standard interfaces



Resource Domains



Resource Domains are a way of classifying services into specific areas that each deal with a different aspect of the problem

An information domain application creates data and associates MetaData with it

Certain Data Storage Interfaces can accommodate both Data and MetaData (XAM, Filesystems with extended attributes)

MetaData aware Data Services interpret Data System MetaData as the requirements for it's lifecycle and implement policies for retention, placement, lifecycle, etc.

Other Data Storage interfaces (based on blocks or objects) provide virtualized Containers for the Data bits and the management of those containers

Storage services are employed to meet those requirements at this point in the data's lifecycle, however the storage services are unaware of the data's requirements