Storage Industry Resource Domain Model

A Technical Proposal
from the SNIA Technical Council
Topics

- Abstract
- Data Storage Interfaces
- Storage Resource Domain
- Data Resource Domain
- Information Resource Domain
- Management of Resources
- Definitions
- Resource Domain Model
- Mapping to existing products
- Future standards
Abstract

- Today’s IT environment is composed of various products that are intended to store, protect, secure and make available the information used by businesses and business processes. These products encompass elements used in both the data path and control path between the user and the eventual location of that information. Standards exist and are emerging for interoperability between these elements, however, what is missing is a comprehensive description of where interoperability is needed and where standards can best be applied.

- This paper sets out a model of these elements that describes a logical view of their functions and capabilities using a descriptive taxonomy. The purpose of this model is to form a basis upon which industry efforts can be organized, needed standards identified and vendor products can be described by vendor independent terminology.
Data Storage Interfaces
What is a Data Storage interface?

- Interface can be an Application Programming Interface (API) or a network (channel) protocol (or both)
- The interface is to a device and/or software that implements one or more “services”
  - To store and retrieve the data, among other functions
- We propose a model where any number of services from different domains can sit behind such an interface
  - The purpose of the model is to describe and categorize these services
- The Application creating and using the data doesn’t care (for the most part) about “services” per se
- But a model helps when we need to manage these services
XAM API: an example Data Storage Interface

- XAM is the first interface to standardize system metadata for retention of data
- XAM implements the basic capability to Read and Write Data (through Xstreams)
- XAM has the ability to locate any XSet with a query or by supplying the XUID
- XAM allows Metadata to be added to the data and keeps both in an XSet object
- XAM 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)
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**
Data vs. Control Path

- Data storage interfaces are used for a mixture of Data Path and Control Path functions.
- Data Path functions are those which implement reading and writing data, locating (addressing) the data, and gaining access to the data.
- Control Path functions are those which implement control over the underlying storage or data services.
  - Can be in-band or out-of-band.
- Control Path examples:
  - File System Metadata (permissions, owner, etc.) \((in-band)\)
  - IOCTLS (in a filesystem interface) \((in-band)\)
  - SCSI Mode Page commands \((in-band)\)
  - SMI-S \((out-of-band)\)
Some example data storage interfaces

- Block Interfaces
  - SCSI, ATA, IDE
- Local File Interfaces
  - POSIX, NTFS
- Network File Interfaces
  - NFS, CIFS, SMB2, Appletalk, Novell, AI
- Object Based
  - OSD, XAM
- Database
  - JDBC, ODBC
Storage Resource Domain
Rather than talk about storage devices, we abstract the storage functions of those devices and talk about *storage services* instead.

This allows us to categorize the different services in these devices and understand (and standardize) the different points of interoperability needed.

This categorization of services of a particular type is called a *resource domain*.

Any given device or subsystem can implement services from any of the resource domains.

In particular we are concerned with *managing these resources interoperably*.
Layering of Storage Services

- The SNIA shared storage model shows a layering of storage services with associated data services.
Metadata and Data Storage Interfaces

- 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.
Data Resource Domain
The data resource domain is the category of services (*data services*) that treat data absent of any context, but whose primary purpose is not to store and retrieve the data itself.

- This is a useful categorization of services that requires a different view of how to manage these services.

The interfaces to the data services are also quite different from those of the storage services.

- In fact, data services can be deployed totally transparent to the actual user of data and the consumer of the data storage interfaces.

Data Services *manage data* in some manner, adding value over and above simple storage and retrieval.

- Historically data services are viewed as point products.
- Examples include: backup/restore, archiving and security.
Elements of a Data Service Interface

- Data services may, in fact, be a consumer of one or more data storage interfaces in order to add this value
  - Backup software, for example, adds value to the data being stored in a disk-based storage service by copying that data to another disk or tape-based storage service and retrieving it when needed.
  - It does this function by consuming the appropriate data storage interfaces.
- A key concept that data services understand is a quantization of data.
  - They can apply differentiated value to individual data elements and groups of data elements.
  - The data element can be a block, volume, file, file system or object.
  - Data services may be able to group data elements and treat all members of the group in the same way.
 Namespace of Data Elements

- A namespace is a context for identifiers, and thus a Data Namespace is a context for the data element identifiers.

- Data services understand the namespace for the data elements they work with.
  - While the Data Namespace is logically part of the data storage interface location concept, a Data Service may make use of that namespace in its functions of enhancing the value of the data.
  - Namespace virtualization allows Data Services to differentiate the value they provide to data elements.

- The namespace is really just a convenient handle for the application (or user) to find the data.
  - Although the growth of data will lead to search as a primary means of locating your data.
Metadata for Data Services

- 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
Information Resource Domain
The Information Resource Domain

- The information resource domain, then, is the category of services that treat data in a context.

- These services are able to treat the data, not just as an opaque set of bits, but also as information within a context.

- An information service may understand what application has generated the data, it may understand the format of the data, or it may understand the relationship of the data to other parts of the environment.

- An information service may examine the information in the environment, extract keywords from data elements, index the content and/or create metadata.

- Because information services are able to understand this context, only information services can be used to help classify data according to its requirements.

- This data classification can then be communicated to data services so that those requirements can be met.

- We differentiate this classification from the grouping of data elements that data services are capable of, but note that once classified, data can be treated as a group by the data 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.
Management of Resources
Data Policies

- Data policies are used to manage data services in support of achieving the data’s requirements.
  - In the absence of data classification, data policies are used by administrators on groups of data that are the target of the policies’ operation.
- Conditions may be time based and may involve the value of data service metadata.
- Events in the environment may also trigger data policies.
- The actions may be any functions of data services that are exposed through a management interface.
  - Some example actions include data placement, data movement and data transformation (such as compression and encryption).
Information Policies

- Information policies are used to ensure that data is treated according to its importance to the organization.
  - Information policies implement business processes regarding the information that applications generate and use.

- Information policy conditions depend on business related properties that are available in the environment.
  - These properties can include the position of employees in the organization (as stored in a corporate directory), properties derived from the information itself, business intelligence applications and business conditions available from financial applications.

- The actions of information policies can set data service metadata corresponding to the requirements of the data at this point in time as well as corresponding to the class of data.
Data Requirement Lifecycles

- The data’s requirements may change over the life of the data and may change based on events internal to the environment or events external to the business (such as a subpoena).

- Data policies can be used to manage the data according to pre-defined lifecycle steps for each different class of data.
  - This is known as Data Lifecycle Management (DLM).
Definitions
Data Related Terms

- **Data**
  - The digital representation of anything in any form.

- **Data Service**
  - A set of functions that treat data without any contextual interpretation. This treatment may, for example, involve copying, movement, security and/or protection, but not the actual storage of the data.

- **Data Resource Domain**
  - The category of resources that exclusively encompass data services.
Information

Information is data that is interpreted within a context such as an application or a process.

Information Service

A set of functions that treat data within an interpretation context.

Information Resource Domain

The category of resources that exclusively encompass information services.
Storage Related Terms

- **Storage**
  - A function that records data and supports retrieval.

- **Storage Service**
  - A storage service is a set of functions that provide data storage.

- **Storage Resource Domain**
  - The category of resources that exclusively encompass storage services.
Resource Domain Model
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.

- **Metadata**
  - Business Context
  - Determines Requirements

- **Location**
  - Interprets System Metadata
  - Consumes to add value to the data

- **ReadWrite Data**
  - Provides storage and retrieval for data

- **Information Resource Domain**
  - Services understand the semantics of the content in context

- **Data Resource Domain**
  - The content is opaque to the Services and without context

- **Storage Resource Domain**
  - The bits are contained by these Services

- **Information Policies**
  - Data classified according to importance to organization

- **Data Policies**
  - Data treated according to requirements, lifecycle

- **Storage Policies**
  - Ensure correct and reliable operation
Mapping to existing products
Product Mapping

- Backup software
- Array with snapshot and remote replication
- Database software
Backup Software

- Backup software may contain services from both the Information Resource Domain and Data Resource Domain
  - Information services classify the data to be backed up, may use policies for classification
  - Data services copy the data under the direction of backup policies
- Standard interfaces are used for storage system metadata, location and read/write of data
- Data classification is typically not marked as Data system metadata
  - But instead is conveyed through internal APIs today
- Information and Data Policies are in a proprietary format
  - Typically modified through a custom user interface
- Points of interoperability that might be standardized
Array with Snapshot and Replication

- Storage arrays may contain services from both the Data Resource Domain and the Storage Resource Domain
  - Snapshot data service keeps consistent, virtual copy of data
  - Remote replication copies data to a remote site
- Standard interfaces are used for location and read/write of data
- There is no per volume metadata that can be used to convey requirements
  - Configuration and control can be done through a standard SMI-S interface however
- Storage Policies can be externalized
  - Not much traction within SMI-S for this so far
Database software

- Database software may contain services from any or all of the resource domains
  - Information services to classify the data in the database instances
  - Data services to protect, secure and ensure data availability
  - Storage services to contain the data and query for its location

- Columns in the database tables can be used as metadata to express requirements of each row of data
  - Similar metadata could be created for each table

- Policies are typically driven by various administrative interfaces
  - But no de jure standard exists today
Future Standardization
Future Standardization

- It’s clear from the Model that metadata can play a key role in interoperability between the Information Resource Domain and the Data Resource Domain.
- Standardizing Data System Metadata for different types of data requirements and their implementing data services will allow for interoperability between these domains.
- The standards can apply equally to Data Storage Interfaces as well as data formats themselves.
  - XAM has both: XSet properties in the API, and an export format that encapsulates these properties.
  - Long Term Retention TWG developing a data format with associated data system metadata.