Storage Industry Resource Domain Model

Version 1.0

Technical Council of the Storage Networking Industry Association

Publication of this SNIA Technical Proposal has been approved by the SNIA. This document represents a stable proposal for use as agreed upon by the SNIA Technical Council. The SNIA does not endorse this proposal for any other purpose than the use described. This proposal may not represent the preferred mode, and the SNIA may update, replace, or release competing proposals at any time. If the intended audience for this release is a liaison standards body, the future support and revision of this proposal may be outside the control of the SNIA. Suggestion for revision should be directed to snia-tc@snia.org

Table of Contents

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.

Introduction

When discussing management technology and standards, it is important to distinguish the types of things that are being managed from the types of management that are applied to those things. We call the categories in each of these spaces, *Domains*. We also define *Resource Domains* to be: the categories of things that are being managed, and *Management Domains:* the categories of management applied to those things.

Another important concept explored in this paper is the concept of *Services*. We categorize services into *Data Path Services* and *Control Path Services*. Of course any given storage industry product can contain any number of these services, but by describing these services in a common fashion, we hope to better understand their interrelationship.

A service is itself an abstract notion, useful in describing what functions are being performed and allowing for the easy categorization of those functions. A disk drive for example, essentially offers a Data Path Service for storing and retrieving blocks of data. All services have a Service Interface through which their functions are offered to consumers of the service. The focus is on services, rather than devices, as the resources that need management because any given device or storage product can provide any number of the described services.

With this terminology as a foundation, we describe taxonomy of these services that can be used to describe the functions of any storage industry product or solution. We identify existing standards where appropriate and identify areas where new standards might be needed. It is our hope that, with this model, we can better focus the efforts of the storage industry on the problems that affect our customers.

What is a Data Storage Interface?

A key concept we use in this paper is that of a Data Storage Interface. The interface can be an Application Programming Interface (API), a network (or channel) protocol, or a combination of both. A Data Storage Interface is between an application (or service) and the underlying set of services that enable reading and writing data, among other functions. The purpose of this model is to categorize and describe the services that live behind this interface and as well as those which consume it. For the most part, Applications only care about the interface they are using, and not the services that implement the interface. However, when we wish to manage those services, a model such as this may be helpful.

Example Data Storage Interface

SNIA has standardized the eXtensible Access Method (XAM[1]) data storage interface for fixed content storage. In addition to the typical functions, this standard Data Storage Interface includes metadata for retention.

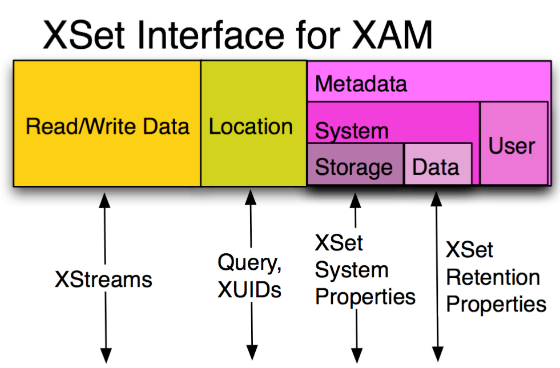

Figure1: XAM Data Storage Interface Standard

As shown above in Figure1, the XAM interface implements the basic capability to read and write data as an XStream. This data is typed by the MIME[23] standard. XAM also 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 it along with the data in the XSet. XAM both uses and produces system metadata as XSet properties. The standard storage system metadata includes access and modification (commit) times, among others. Uniquely, XAM also standardizes data system metadata for retention values – interpreted by retention data services in the system. XAM user metadata is un-interpreted by the system, but can be used in queries.

Other data storage interfaces have some of these same capabilities and can use the same elements as shown above, but XAM has defined elements in all of the above categories.

POSIX, for example, defines standard metadata for file times, permissions, owner, and group. The file system both provides this metadata and interprets the values. It also provides a means to store and retrieve data on a file-by-file basis. Lastly it also provides a location mechanism with a namespace for finding files.

A storage service implements a Data Storage Interface and provides the functions of read/write, location (with namespace) and optionally metadata. Storage services are defined as those whose primary responsibility is the storage and retrieval of data.

The Storage Resource Domain

The storage resource domain then consists of the services which have functions primarily oriented toward the storage and retrieval of data. This includes services for disk storage, tape storage and their virtual counterparts. It includes services with block, file and object data storage interfaces.

Data Path and Control Path

The Data Storage Interfaces for these storage services may have a mixture of data path and control path functions. Data path functions are those used to actually transfer the data to and from the service without concern for anything other than the location of the data. Control path functions are those concerned with aspects of how the service is configured and what quality of service the storage provides such as performance, security and protection. Control path functions may also be used for the device that offers the service, and to monitor and control the storage services on the device.

Note that any given Data Storage Interface may comprise both data path and control path functions and commands. We treat these separately for purposes of this taxonomy. Also, the interfaces may be an API (such as a library), a channel protocol or a network protocol without affecting the taxonomy as well. Lastly, management is recognized as available either or both as an in-band control path (using the same interface or protocol as the data path) or an out-of-band control path (using a separate interface or protocol).

Elements of a Data Storage Interface

A fundamental aspect of a Data Storage Interface is that it allows the consumer of that interface to store and retrieve data. All data storage interfaces have a concept of location that is integral to this capability. Block data storage interfaces typically have a Logical Block Address (LBA) that indicates the location of a block of data. File and object data storage interfaces typically have a handle or descriptor to locate the data. The interface may also have a notion of the current location within a file or object that is exposed through the interface.

A data storage interface will always have functions for reading and writing the data, of course. It may also have functions for moving the current location (such as a seek).

Databases provide a special type of interface for providing data storage, oriented toward tables of data elements with special functions for manipulating the relationships between data elements and doing queries based on the actual contents of the data elements.

All of these examples, however, share the common feature of being primarily used to store and retrieve data from the service.

Layering of Storage Services

We have previously discussed the layering of various storage services in the SNIA Shared Storage model as seen below:

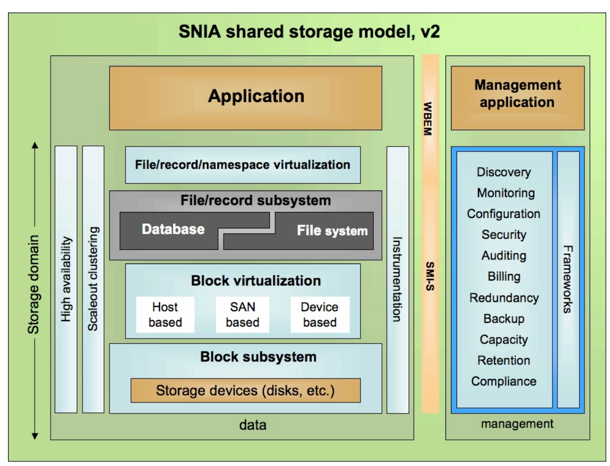

Figure2: The SNIA Shared Storage Model

Any given Data Storage Interface can be *virtualized* to provide not the actual storage function itself, but a point of abstraction behind which, further services (such as location independence and transparent data services) can be added.

Management of Storage Services

Orthogonal to this model for resource domains are categories for management services used to manage the various resource domains. The typical enumeration of services in the Management Domain is known as FCAPS[4] (Fault, Configuration, Accounting, Performance and Security Management). These management services are applied to the storage services through management interfaces in the control path to the storage services. SNIA has standardized a majority of these interfaces with the Storage Management Initiative Specification (SMI-S[5]). This is out-of-band management that involves standard protocols (CIM-XML[6] and WS-Man[7]) and the standard Common Information Model (CIM[8]) from the Distributed Management Task Force (DMTF).

New standards are being created both in the management interfaces to resource domain services, and in the management domain itself for common services that can form a framework for management applications. Services in the Management Domain for storage services are not further discussed in this model.

Metadata in Data Storage Interfaces

Data storage interfaces may provide functions for metadata. 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. This can include time stamps as well as access control lists and properties. This is illustrated graphically below:

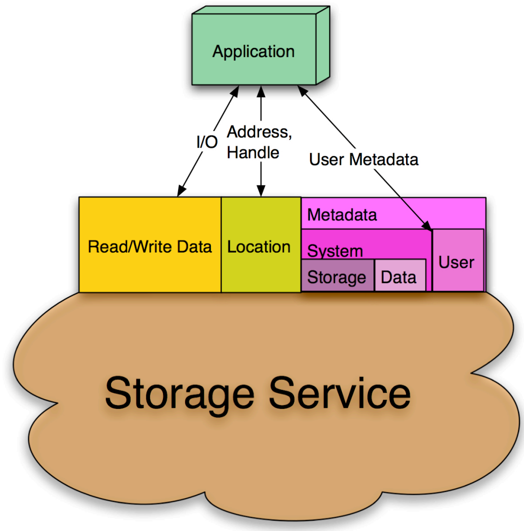

Figure3: Storage Service use of Data Storage Interface

We call this *storage system metadata*, as it is system metadata 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 storage services.

The Data Resource Domain

The data resource domain is the category of 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 data services' use of data storage interfaces 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.

Elements of a Data Storage Interface

Data services are typically used to add some value to the data stored by storage services. They may, in fact, be a consumer of one or more storage services 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.

Data services may be managed through a management interface that allows them to be configured, monitored and to direct them to operate on specific data or groups of data. This management can take the form of administrative user interfaces or they can even be driven from policy.

Namespace for 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 individual data elements.

Metadata in Data Services

Metadata available through the Data Storage Interface may also be managed by data services. This data system metadata can be used by data services to provide differentiated value to individual data elements. The model or schema for data system metadata may be defined by each data service and may be standardized. This is illustrated in the figure below:

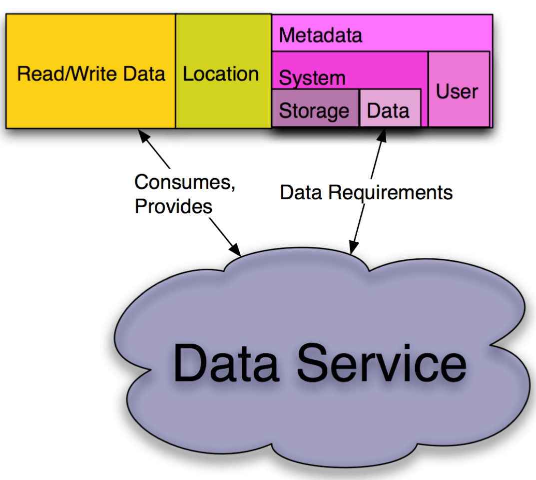

Figure4: Data Service use of Data Storage Interface

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 associated 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.

Metadata in Information Services

The role of metadata in information services can be as a communication mechanism with the underlying storage services and data services. Information services are primarily concerned with the data 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. This is illustrated below:

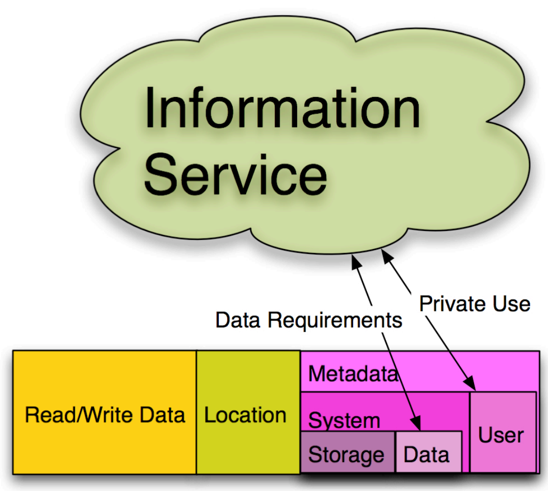

Figure5: Information Service use of Data Storage Interface

Policies

Policy based management is used to help reduce the complexity and help automate the management of resources. Policies are rules that express the intent of an administrator on how the resource should be managed. Those rules are composed of conditions that must be satisfied and actions that should result.

Policies, like services, may be part of any given storage industry product. The expression of policies may be standard or proprietary and may act against standard or proprietary management interfaces. Policies can be grouped according to the resource domain they operate against as described below.

Storage Policies

Storage policies are used to manage storage services. The conditions can be based on any event or instrumented property in the storage resource domain. The actions may be any storage service function exposed through a management interface. Storage policies may be specific to a management domain or may be used to achieve results in multiple management domains. For example, a policy to correct a failed storage network link would support both the Fault and Performance management domains.

Storage policies' primary purpose is to ensure the correct and reliable operation of storage services in their primary purpose of storing and retrieving data.

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 system 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 system metadata corresponding to the requirements of the data over its lifetime 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).

Data Storage Interface Standards

A number of standards exist for data storage interfaces, but more are being standardized as additional capabilities are added to storage systems and solutions. In the block storage space, the SCSI[9] standard is widely used for enterprise storage and ATA[10] is used for desktop and consumer storage. These standards combine a data storage interface with a small amount of in-band management of the storage device. The semantics of these interfaces are leveraged by storage networking protocols (iSCSI[11], Fibre Channel[12] and SATA[13]) and are reflected in operating system interfaces. The Object-based Storage Device (OSD[14]) standard extends these semantics to handle objects instead of blocks.

There are also file system API standards as well such as POSIX[15] that allow applications to create, read and update files.

Ontology of storage industry concepts

It is important to define the terms we will use in our ontology. The key high-level terms we use are Storage, Data and Information. These terms as nouns and their use as adjectives in other terms are defined in the following sections.

Data

We start with the term data, as it is foundational to the other concepts we wish to define here. According to the [SNIA Dictionary definition of data]() it's defined as:

`Data`

`The digital representation of anything in any form.`

This is a very comprehensive definition for data as a noun, but when we use it as adjective, we will want to further discriminate its use.

`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.`

We have thus defined the set of services that manipulate data as if it was an opaque sequence of bits. We have also taken care to differentiate these services from both the Information and Storage services.

Information

Building on the above definitions, the definition of Information can also be taken from the SNIA Dictionary:

`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

And next, we define the corresponding storage related terms:

`Storage`

`a function that records data and supports retrieval.`

`Storage Service`

`A storage service is a set of functions that provide storage.`

`Storage Resource Domain`

`The category of resources that exclusively encompass storage services.`

Metadata

Finally, we wish to make clear the definition of Metadata as used in this paper. Again, taking from the SNIA Dictionary:

`Metadata`

`Information about data.`

We distinguish between *System* Metadata and *User* Metadata, however, and define *system metadata* as any metadata that is managed or interpreted by a storage system (composed of any of the above defined services). User metadata is defined as metadata that is un-interpreted by the services in a storage system.

Resource Domain Model

The resource domain model is illustrated below:

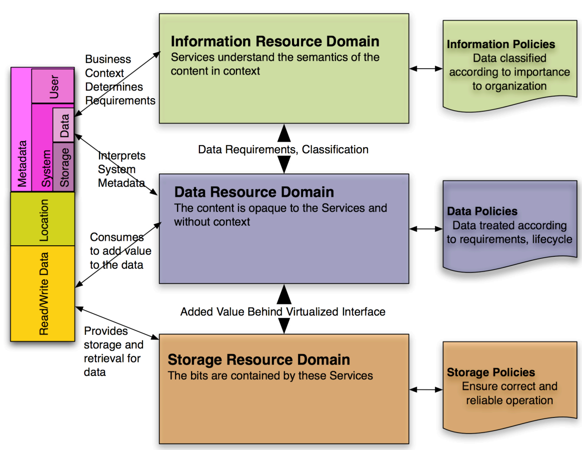

Figure6: Resource Domain Model

This 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.

The data storage interface model is on the left (shown vertically) and the interactions between domains and with the data storage interface are called out. The storage resource domain implements the basic read and write of data from the data storage interface as well as the location part of the interface. Storage policies are primarily deployed to manage the storage services (ideally through the SMI-S interface) and endure their correct and reliable operation. Storage services can thus better meet a consistent level of service that may be expected by the data services.

The data resource domain has the data services that add value to the data, keeping it secure, available and performant. Data policies manage the data services and act to ensure that each data element has its requirements met. This will, over time, be more driven from the data system metadata such as that standardized in XAM for retention.

The information resource domain has the services which understand the semantics of the information and its relationship to the environment in which it is created and used. Information policies will be able to use this context to classify data elements according to their individual requirements, setting the data system metadata accordingly.

Storage Industry Product Examples

As mentioned above, many of the existing storage industry products will have services that fall into one or more of these resource domains. The model can be used in a descriptive manner to highlight the different services in each product.

Backup Software

Backup software can be mapped to the Resource Domain Model as shown below:

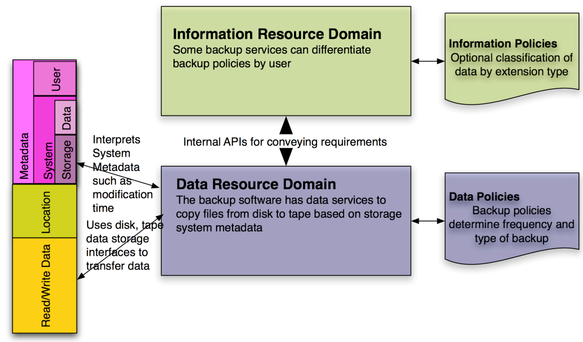

Figure7: Resource Domain Model of Backup Software

Backup may contain services from both the Information Resource Domain and the Data Resource Domain. To the extent that backup software can classify data based on the type of file it is (file extension, for example) it has Information Policies as part of it. It may have information services that can identify each user's files and apply different backup policies as a result. Usually this classification is conveyed to the data services via internal APIs as opposed to marking the files with data system metadata. The Data Services in the backup software copy data (using the Read/Write and Location parts of the data storage interface) to tape or other media and maintain an index and log of its operation. Data Policies are driven off of time periods and events, along with storage system metadata (such as modification time) to initiate and control the backup data services.

Array with Snapshot and Remote Replication

An array with added data services can also be mapped into the Resource Domain Model. The basic capability of the array to store and retrieve data from virtual volumes is a storage service and is part of the Storage Resource Domain. Logical block addresses are used to locate any give block in each volume, and Target and LUN addresses are used to locate any given volume. The storage services can be configured to achieve the desired protection and performance qualities using parity and striping. Storage policies can be used to automatically deploy a spare drive in the event of failure, or to activate a passive controller.

The deployment of snapshot and remote replication capabilities to the array introduce data services in the Data Resource Domain as well as shown below:

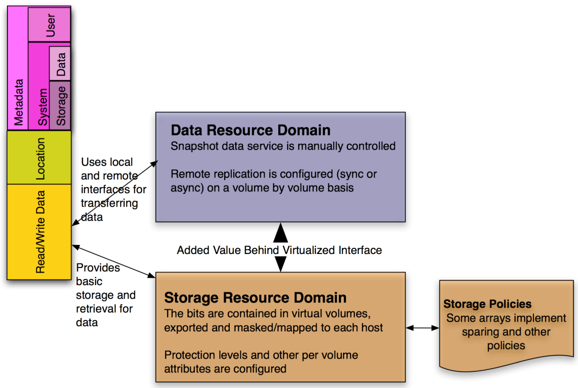

Figure8: Resource Domain Model of an Array with Snapshot and Remote Replication

The snapshot data service allows virtual copies of the data at a point in time to be taken and then exposed through a data storage interface. The remote replication data service copies the data to a remote location and can usually be configured for synchronous or asynchronous operation. There is no current standard that would allow data system metadata to be set on each volume to control these data services, but SMI-S does have a standard interface for configuration and control.

Database Software

Database software may contain services from any or all of the Resource Domains in the model as shown below:

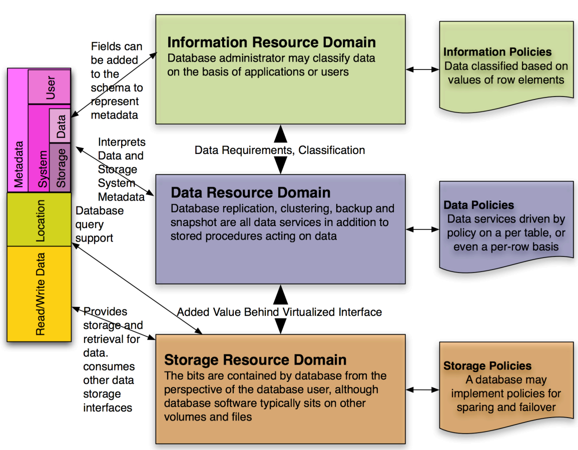

Figure8: Resource Domain Model of an Array with Snapshot and Remote Replication

The database administrator can classify data on the basis of applications or users by using information services int the Information Resource Domain. Information policies might be able to classify data based on the value of elements in each row (basically user metadata). Additional columns can be added to the schema to represent data system metadata for use by retention and other data services.

Data services are typically part of databases as well to protect, secure and ensure the data reliability. Stored procedures may act on the data as well and are considered data services and part of the Data Resource Domain. Data Policies can be driven on the basis of each table or even row of data.

A database also acts as a storage service from the point of view of the database user, storing and retrieving data through standard interfaces such JDBC[16] an ODBC[17]. Storage policies for redundancy and failover can maintain the desired quality of this storage service. A database will typically consume the data storage interface of other storage services for block and file access.

Future Standardization

It's clear from this 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 and the services within them.

The SNIA will look for opportunities to standardize this type of metadata for the various Data Storage Interfaces in the industry as appropriate. This will not only lead to greater interoperability between vendors, but also lead to better alignment of data services with business policies. Best of breed information services will be able to be deployed against data services driven off of standard data system metadata.

The metadata standards can also apply to data formats themselves when the metadata is kept with the data in such a format. XAM already standardizes this with the XSet export/import format. The Long Term Retention Technical Work Group may also define similar metadata standards in the future.

Longer term, it may be possible to define standard Data and Information policies and the languages they are written in. This might look something like the DMTF standard CIM-SPL[18] language that can be used for storage policies based on SMI-S.

Summary

The Storage Industry Resource Domain Model described herein is based on a decomposition of the services contained in storage products today, and a classification of services of similar type into domains. It is hoped that this will lead to a better understanding of the needs of each domain and identification where interoperability standards need to be defined.

The SNIA encourages feedback on this model and its use in the industry.

References

[1] XAM Architecture Specification Version 1 (http://www.snia.org/forums/xam/technology/specs/XAM_Arch_v1.0_Apr6.pdf)

[2] [RFC2046] Freed, N, et al., "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

[3] [RFC2387] Levinson, E. "The MIME Multipart/Related Content-type", RFC 2387, August 1998.

[4] ITU-T Recommendation M.3400 (2000), TMN management functions.

[5] "Storage Management Initiative Specification," - ISO/IEC 24775:2007, http://webstore.ansi.org/RecordDetail.aspx?sku=ISO%2fIEC+24775%3a2007

[6] DMTF DSP0214:2004 CIM Operations over HTTP

[7] DMTF DSP0226 Web Services for Management (WS Management)

[8] DMTF CIM Schema 2.15.0 http://www.dmtf.org/standards/cim/cim_schema_v215

[9] ISO/IEC 14776-413, SCSI Architecture Model - 3 (SAM-3) [ANSI INCITS 402-200x]

[10] INCITS 370-2004 (1510D): ATA Host Adapter Standards

[11] RFC 3720 iSCSI April 2004

[12] ANSI/INCITS 374:2003, Information technology - Fibre Channel Single - Byte Command Set-3 (FC-SB-3)

[13] Serial ATA 2.6 Specification (released February 2007) http://www.sata-io.org

[14] T10 Object-Based Storage Devices - 2 (OSD-2) http://t10.org

[15] ISO/IEC 9945:2003 POSIX (identical to IEEE Std 1003.1™-2003)

[16] JSR 221: JDBC™ 4.0 API Specification (http://www.jcp.org/en/jsr/detail?id=221)

[17] ISO/IEC 9075-3:1995, Call-Level Interface (SQL/CLI)

[18] DMTF DSP0231 CIM Simplified Policy Language (CIM-SPL)

