



What's Important About Managing Cloud Storage

*The cloud computing model – with its ability to remotely “borrow” for a fee various IT components through a specialized or open interface – offers an important new technology approach, and storage is a natural fit. With its potential to allow unprecedented resources and services scalability to be accessed through global IT infrastructures and the Internet, cloud storage promises to reduce IT costs and lessen complexity. **Written by the members of the SNIA's Cloud Technical Working Group.***

An important part of the cloud model in general is the concept of a pool of resources that is drawn upon on demand in small increments (smaller than what you would typically purchase by buying equipment). This is made possible by recent innovations in virtualization. The appeal of cloud storage is similar to other cloud services: pay as you go, the illusion of infinite capacity (elasticity), and the simplicity of use/management. It is therefore important that any interface for cloud storage support these attributes, while allowing for a multitude of business cases and offerings, long into the future.

Data Storage as a Service

DaaS is delivery of virtualized storage on demand. By abstracting data storage behind a set of service interfaces and delivering it on demand, a wide range of actual offerings and implementations are possible. The only type of storage that is excluded from this definition

is that which is delivered, not based on demand, but on fixed capacity increments.

An important part of any DaaS offering is the support of legacy clients. This is accommodated with existing standard protocols such as iSCSI for block and CIFS/NFS or WebDAV for file network storage as shown below:

The difference between the purchase of a dedicated appliance and that of cloud storage is not the functional interface, but merely the fact that the storage is delivered on demand. The customer pays for either what they actually use or in other cases, what they have allocated for use. In the case of block storage, a LUN or virtual volume is the granularity of allocation. For file protocols, a filesystem is the unit of granularity. In either case, the actual storage space can be thin provisioned and billed for based on actual usage. Data services such as compression and deduplication can be used to further reduce the actual space consumed.

The management of this storage is typically done out-of-band of these standard data storage interfaces, either through an API, or more commonly, though an administrative browser based user interface. This interface may be used to invoke other data services as well, such as snapshot and cloning.

In this model we abstract the underlying storage space exposed by these interfaces using the notion of a container. A container is not only a useful abstraction for storage space, but also serves as a grouping of the data stored in it, and a point of control for applying data services in the aggregate.

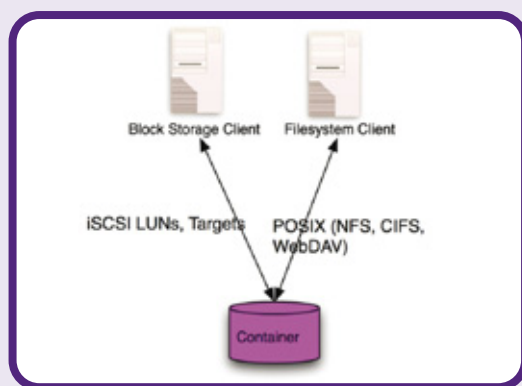


Figure 1: Existing Data Storage Interface Standards



Figure 2: Database/Table Data Storage Interfaces

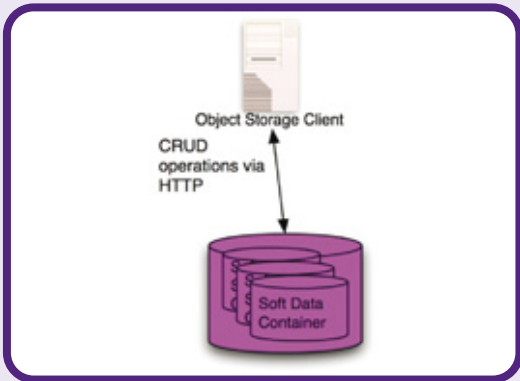


Figure 3: CRUD/HTTP Data Storage Interfaces

Another type of DaaS offering is one of simple table space storage, allowing for horizontal scaling of database-like operations needed by certain applications. Rather than virtualising relational database instances, these offerings offer a new data storage interface of limited functionality with the emphasis on scalability rather than features. This allows the tables to be partitioned across multiple nodes based on common key values, affording horizontal scalability at the expense of functions that can typically only be implemented by a vertically scaled relational database.

There is a great deal of innovation and change happening in these interfaces, and the offerings each have their own unique proprietary interface as shown below:

Due to the rapid innovation in this space, it is probably best to wait for further development of this type of cloud storage before trying to standardize a functional interface for this type of storage.

There is a third category of functional interface for Data Storage that has emerged. This type of interface treats every data object as accessible via a unique URI. It can then be fetched using the standard HTTP protocol and a browser can be used to invoke the appropriate application to deal with the data.

Each data object is Created, Retrieved, Updated and Deleted (CRUD) as a separate resource. In this type of interface, a container, if used, is a simple grouping of data objects for convenience. There is nothing preventing the concept of containers in this case from being hierarchical, although any given implementation might support only a single level of such. We call this type of container a “soft” container as shown below:

While there are several proprietary examples of this type of interface, they all pretty much support the same set of operations. This, then, is an area ripe for standardization.

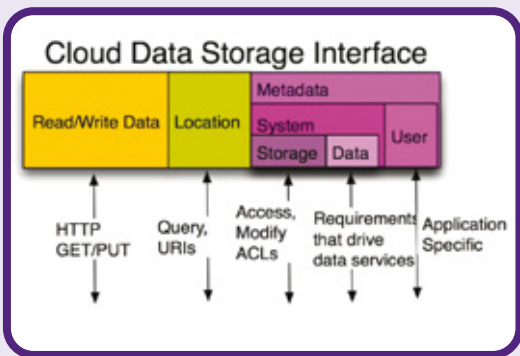


Figure 4: Using the Resource Domain Model

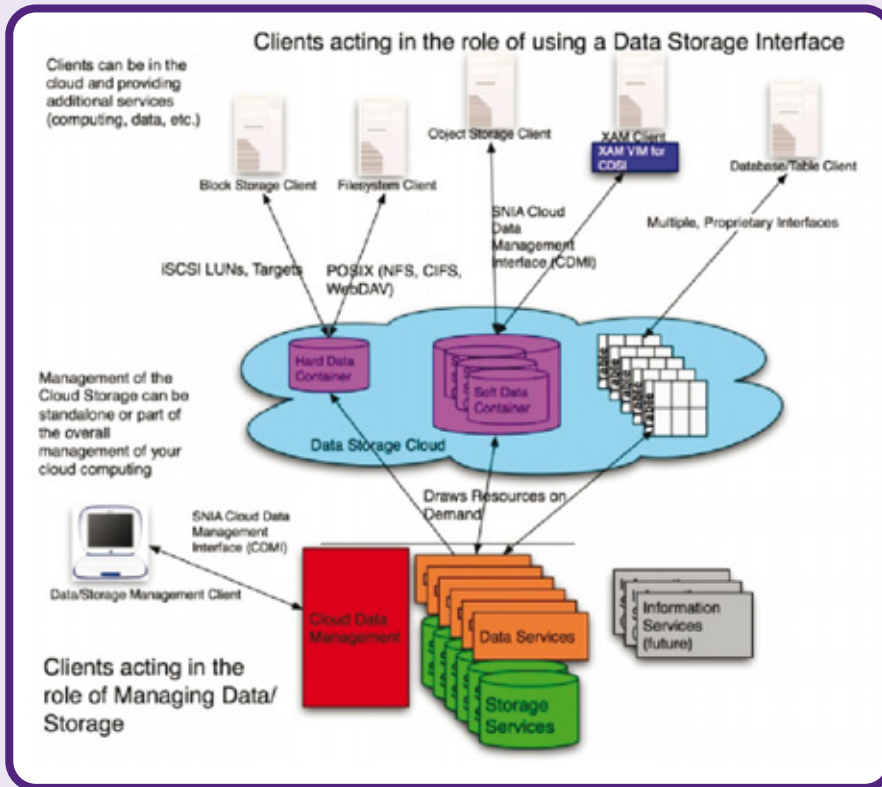


Figure 5: Cloud Storage Reference Model

Managing Data in the Cloud

Many of the initial offerings of cloud storage focused on a kind of “best effort” quality of storage service with very little offering of additional data services for that data. In order to address the needs of enterprise applications with cloud storage, however, there is increasing pressure to offer better quality of service and the deployment of additional data services.

The danger, of course, is that cloud storage loses its benefit of simplicity and the abstraction of complexity as additional data services are applied, and the implication that these services need to be managed.

Fortunately, the SNIA Resource Domain Model gives us way to minimize this complexity and address the need of cloud storage to remain simple. By using the different types of metadata discussed in the model for a cloud

storage interface, we can create an interface that allows offerings to meet the requirements of the data without adding undo complexity to the management of that data.

By supporting metadata in a cloud storage interface standard and proscribing how the storage and data system metadata is interpreted to meet the requirements of the data, we can retain the simplicity required by the cloud storage paradigm, and yet still address the requirements of enterprise applications and their data.

Managing Data and Containers

There is no reason that the management of data and the management of containers should involve different paradigms. As a result, the SNIA has proposed extending the use of metadata from applying to individual data



elements into applying to containers of data as well. Thus any data placed into a container essentially inherits the metadata of the container it was placed into. Creating a new container within an existing container would similarly inherit its parent's metadata settings. Of course the metadata can be overridden at the container or individual data element level as desired.

Even if the functional interface provided by the offering does not support this type of metadata on individual data elements, it can still be applied to the containers even though it cannot be overridden on the basis of individual data elements. For file-based interfaces that support extended attributes (i.e. CIFS, NFSv4), these extended attributes can be used to specify the Data System Metadata to override that specified for the container. The mapping of extended attribute names and values to individual file data requirements as supported by cloud storage will be done as a follow on effort.

This model shows multiple types of cloud data storage interfaces able to support both legacy and new applications. All of the interfaces allow storage to be provided on demand, drawn from a pool of resources. The capacity is drawn from a pool of storage capacity provided by storage services. The data services are applied to individual data elements as determined by the data system metadata. Metadata specifies the data requirements on the basis of individual data elements or on groups of data elements (containers).

SNIA Cloud Data Management Interface

As shown in Figure 5, the SNIA Cloud Data Management Interface (CDMI) is the functional interface that applications will use to create, retrieve, update and delete data elements from the cloud. As part of this interface the client will be able to discover the capabilities of the cloud storage offering and use this interface to manage containers and the data that is placed in them. In addition, metadata can be set on containers and their contained data elements through this interface.

It is expected that the interface will be able to be implemented by the majority of existing cloud storage offerings today. This can be done with an adapter to their existing proprietary interface, or by implementing the interface directly. In addition, existing client libraries such as eXtensible Access Method (XAM) can be adapted to this interface as show in Figure 5.

This interface is also used by administrative and management applications to manage containers, accounts, security access and monitoring/billing information, even for storage that is accessible by other protocols. The capabilities of the underlying storage and data services are exposed so that clients can understand the offering.

Conformant cloud offerings may offer a subset of either interface as long as they expose the limitations in the capabilities part of the interface.

SNIA Cloud Storage Activities

SNIA has created the Cloud Storage Technical Work Group (TWG) for the purpose of developing SNIA Architecture and Software related to system implementations of Cloud Storage technology.

The Cloud Storage TWG is creating a standard called the Cloud Data Management Interface (CDMI) and will also create an open source reference implementation of the standard. In addition, SNIA has approved the formation of a Cloud Storage Initiative (CSI) to further promote and educate about Cloud Storage technologies and standards and is actively soliciting membership in preparation for a launch of the group later this year.

The SNIA CSI will also coordinate its activities with other industry Cloud efforts in order to drive a broader adoption and consistency of Cloud technologies and standards.

For more information on the SNIA's Cloud Storage, CDMI, including the Object Model, please see the full Reference Model draft at www.snia.org/cloud