

The logo for Storage Developer Conference 2018 (SDC 18) is displayed in white on a dark blue background. It consists of the letters 'S', 'D', and 'C' in a large, bold, sans-serif font, with the number '18' inside a smaller circle to the right of the 'C'.

SDC 18

September 24-27, 2018
Santa Clara, CA

The website address 'www.storagedeveloper.org' is written in white text on a yellow-green horizontal bar.

www.storagedeveloper.org

Using CDMI to create/backup, restore and replicate storage objects and data across Storage vendors and cloud storage (Swift, S3, Ceph)

**Dhishankar and Umang
HPE**

Introduction

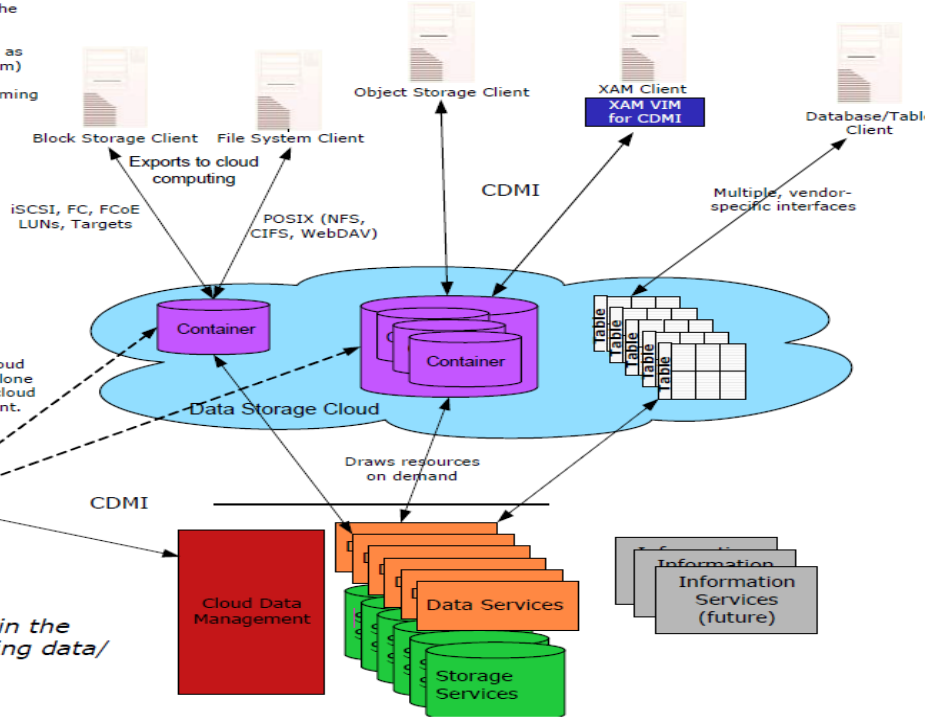
- ❑ CDMI (Cloud Data Management Interface) is an open RESTful API standard developed and governed by SNIA for managing and accessing storage via out of band interfaces.
- ❑ CDMI defines both a means to manage the data as well as a means to store and retrieve the data.
- ❑ It provides a standard data interface for cloud storage across multiple APIs like Amazon S3, native CDMI clients, Object Storage, etc.
- ❑ It also provides access via standard datapath interfaces like iSCSI (Block storage) and NFS for File access.
- ❑ CDMI standardizes CRUD Operations

The Big Picture

Clients can be inside the storage cloud (i.e., providing storage resources to the cloud as well as consuming them) or outside the storage cloud (i.e., only consuming resources).

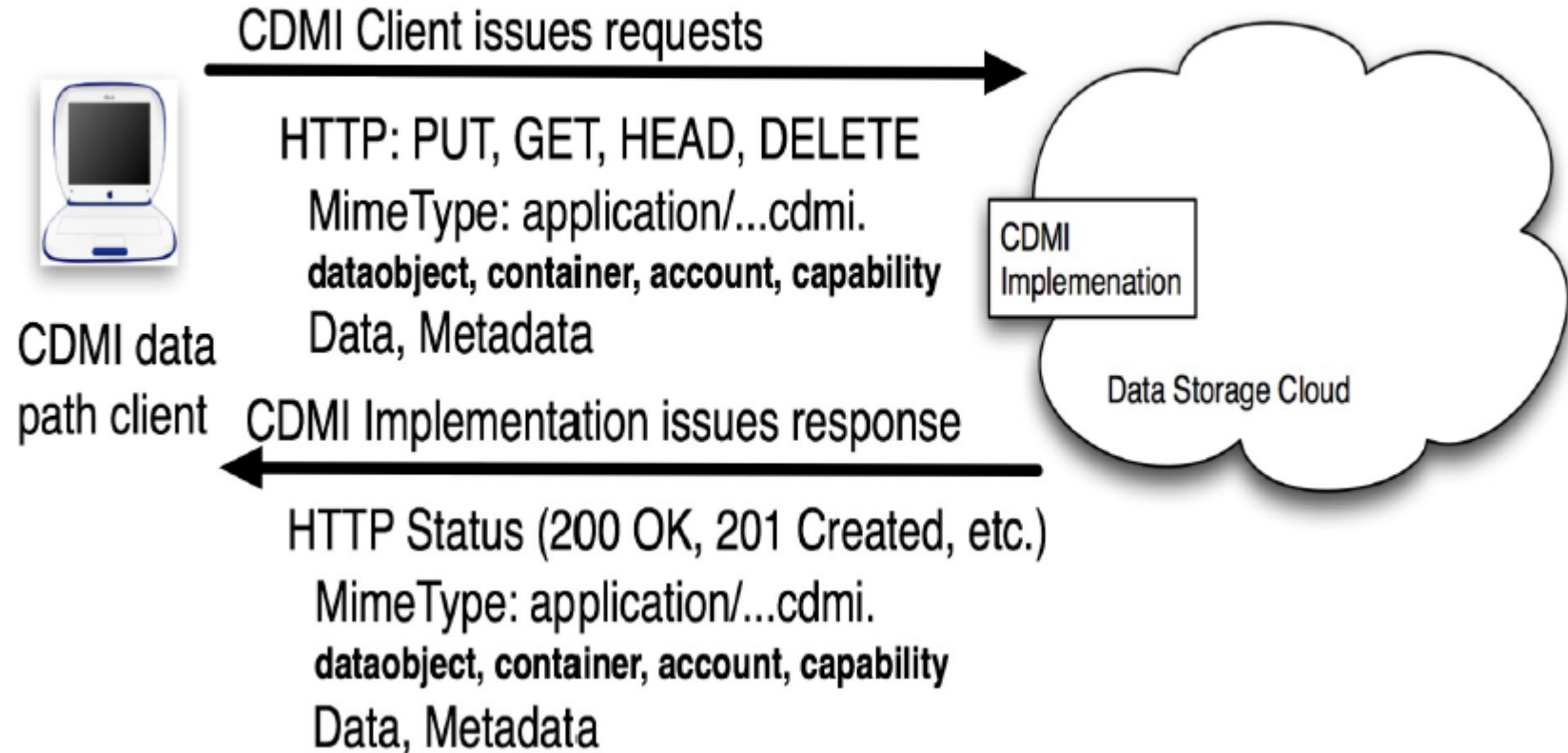
Management of the cloud storage can be standalone or part of the overall cloud computing management.

Clients acting in the role of managing data/storage



- Provides the concept of a single namespace for different data types (S3, File systems, LUNs, Swift, etc) which allows this interoperability and manageability across the tree based hierarchies.
- Provides both a data path to the cloud service and a management path for the cloud data,
- CDMI is the functional interface that applications will use to Create, Retrieve, Update and Delete (CRUD semantics) data elements in the cloud.
- Discover the capabilities of the cloud storage offering and use this interface to manage containers and the data that is placed in them

□ CDMI Basic flow:



PUT to the container URI the data object name and contents

```
PUT: /MyContainer/MyDataObject.txt
Host: cloud.example.com
Accept: application/vnd.org.snia.cdmi.dataobject+json
Content-Type: application/vnd.org.snia.cdmi.dataobject+json
X-CDMI-Specification-Version: 1.0
{
  "mimetype" : "application/txt",
  "metadata" : [ ],
  "value" : "This is the Contents",
}
```

The response looks like:

```
HTTP/1.1 201 Created
Content-Type: application/vnd.org.snia.cdmi.dataobject+json
X-CDMI-Specification-Version: 1.0
{
  "objectURI" : "/MyContainer/MyDataObject",
  "objectID" : "AAAAFAAo7EFMb3JlbSBpcHN1bSBkb2xvciBzaXQgYW1ldCBhbWV0Lg==",
  "parentURI" : "/MyContainer",
  "accountURI" : "/cdmi_accounts/MyAccount",
}
```

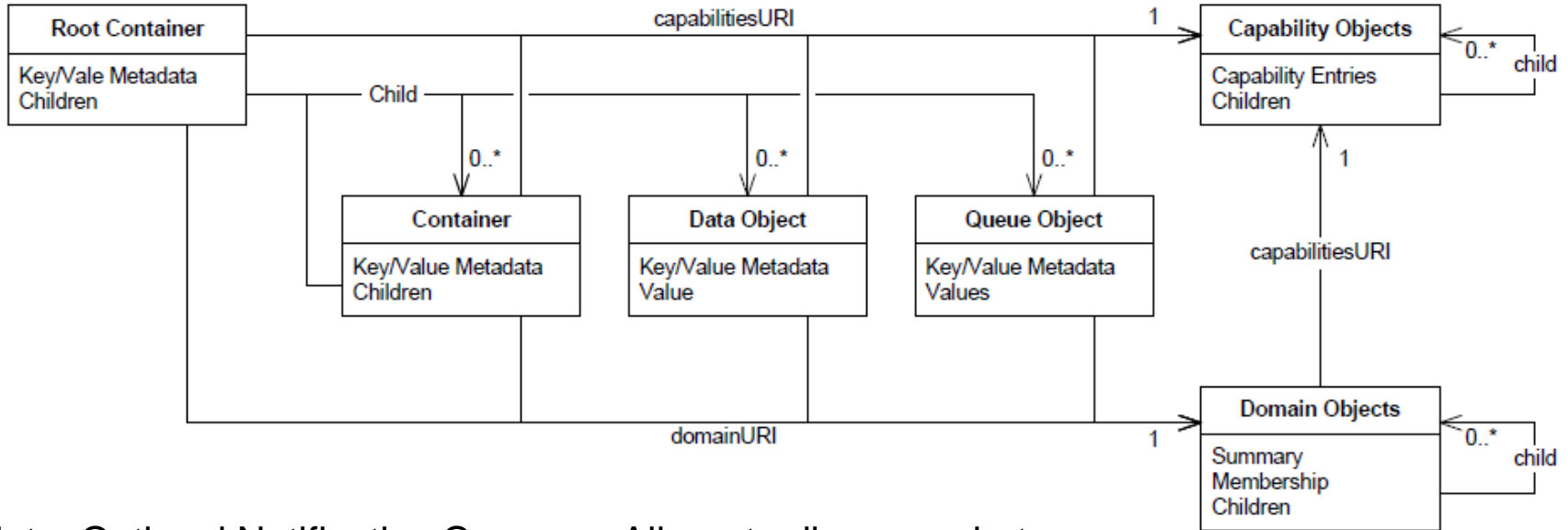
GET from the data object URI

```
GET: /MyContainer/MyDataObject.txt  
Host: cloud.example.com
```

The response looks like:

```
200 OK  
Content-Type: application/txt  
This is the Contents
```

CDMI Object Models



Note: Optional Notification Queues – Allows to discover what changes have occurred in the system

What is a Container

- ❑ Containers are STORAGE abstraction to put data.
- ❑ A container serves as a grouping of the data stored in it and a point of control for applying data services in the aggregate.
- ❑ Can be modified later to add additional exports
- ❑ Note: The specification defines a list of container capabilities.



What is a Data Object

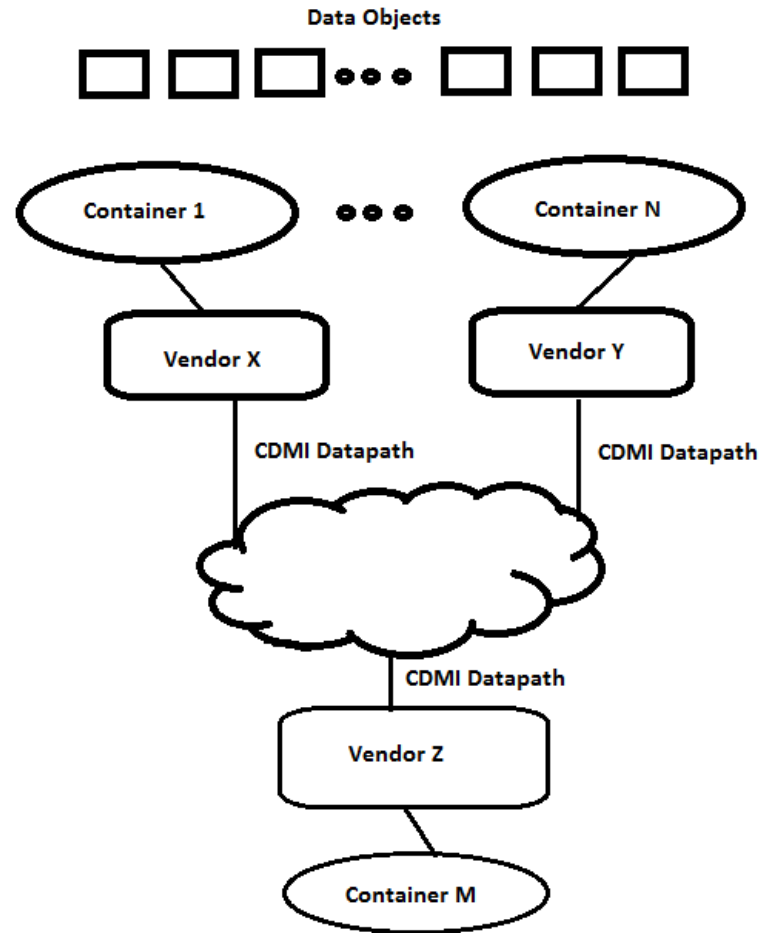
- ❑ Data objects are the fundamental storage component within CDMI™ and are analogous to files within a file system. Each data object has a set of well-defined fields that include:
 - ❑ a single value; and
 - ❑ optional metadata that is generated by the cloud storage system and specified by the cloud user.
- ❑ Data objects are addressed in CDMI in two ways:
 - ❑ by name (e.g., <http://cloud.example.com/dataobject>); and
 - ❑ by object ID (e.g., http://cloud.example.com/cdmi_objectid/00007ED90010D891022876A8DE0BC0FD).
- ❑ Example – LUN, File, ObjectStore, Directory, etc

Data Object Capabilities

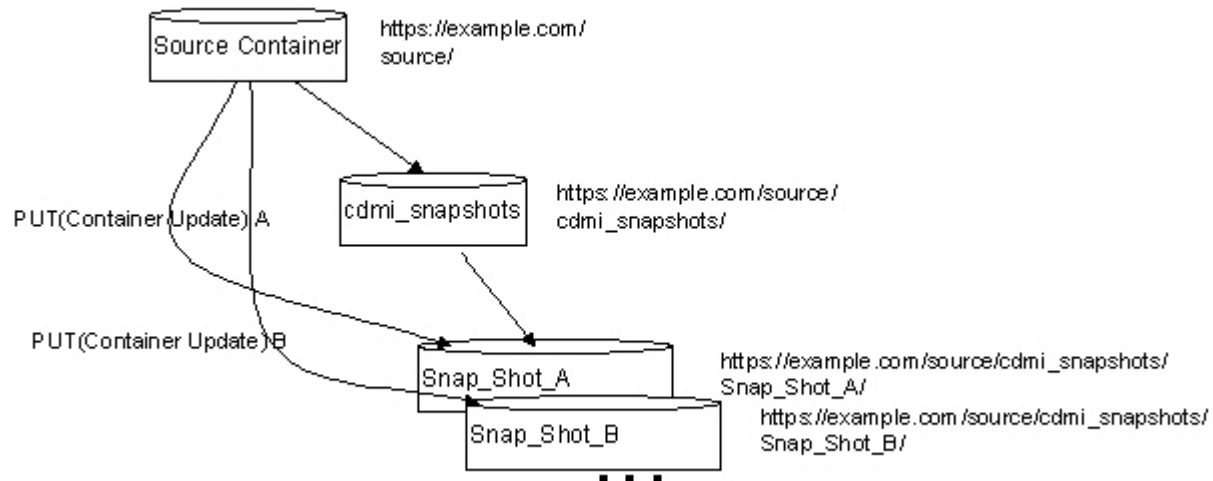
Capability name	Type	Definition
cdmi_read_value	JSON string	If present and "true", this capability indicates that the cloud storage system shall support the ability to read the object's value.
cdmi_read_value_range	JSON string	If present and "true", this capability indicates that the cloud storage system shall support the ability to read the object's value with byte ranges.
cdmi_read_metadata	JSON string	If present and "true", this capability indicates that the cloud storage system shall support the ability to read the object's metadata.
cdmi_modify_value	JSON string	If present and "true", this capability indicates that the cloud storage system shall support the ability to modify the object's value.
cdmi_modify_value_range	JSON string	If present and "true", this capability indicates that the cloud storage system shall support the ability to modify the object's value with byte ranges.
cdmi_modify_metadata	JSON string	If present and "true", this capability indicates that the cloud storage system shall support the ability to modify the object's metadata.
cdmi_modify_deserialize_dataobject	JSON string	If present and "true", this capability indicates that the cloud storage system shall support the ability of the data object to deserialize a serialized data object into the data object as an update.
cdmi_delete_dataobject	JSON string	If present and "true", this capability indicates that the cloud storage system shall support the ability to delete the object.

Migration Use-case

- The ability to move data between heterogeneous storage devices addresses use case of data migration.
- To develop a migration tool one would need to know the management & data interface of all the heterogeneous storage devices.
- The tools currently available are not standard tools and don't provide the ability to migrated data between on-prem and cloud and visa-versa
- CDMI interface not just provides the ability to manage the storage it also provides the ability to access the data in the storage.
- CDMI supports Block, File and Object storage, using its data path interface data can be moved back and forth.



Taking a Snapshot of the Source Data



Copying/Migrating the Data object from Snapshot container to Destination container:

- Listing/Discovering all data objects from Snapshot Data container.

EXAMPLE Perform a GET to the container URI:

```
GET /SnapContainer/ HTTP/1.1
Host: cloud.example.com
Accept: */*
X-CDMI-Specification-Version: 1.0.2
```

- Reading the Contents of each Data Object from Snapshot Data container (Data+Metadata)(GET operation)

EXAMPLE GET from the data object URI:

```
GET /SnapContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
X-CDMI-Specification-Version: 1.0.2
```

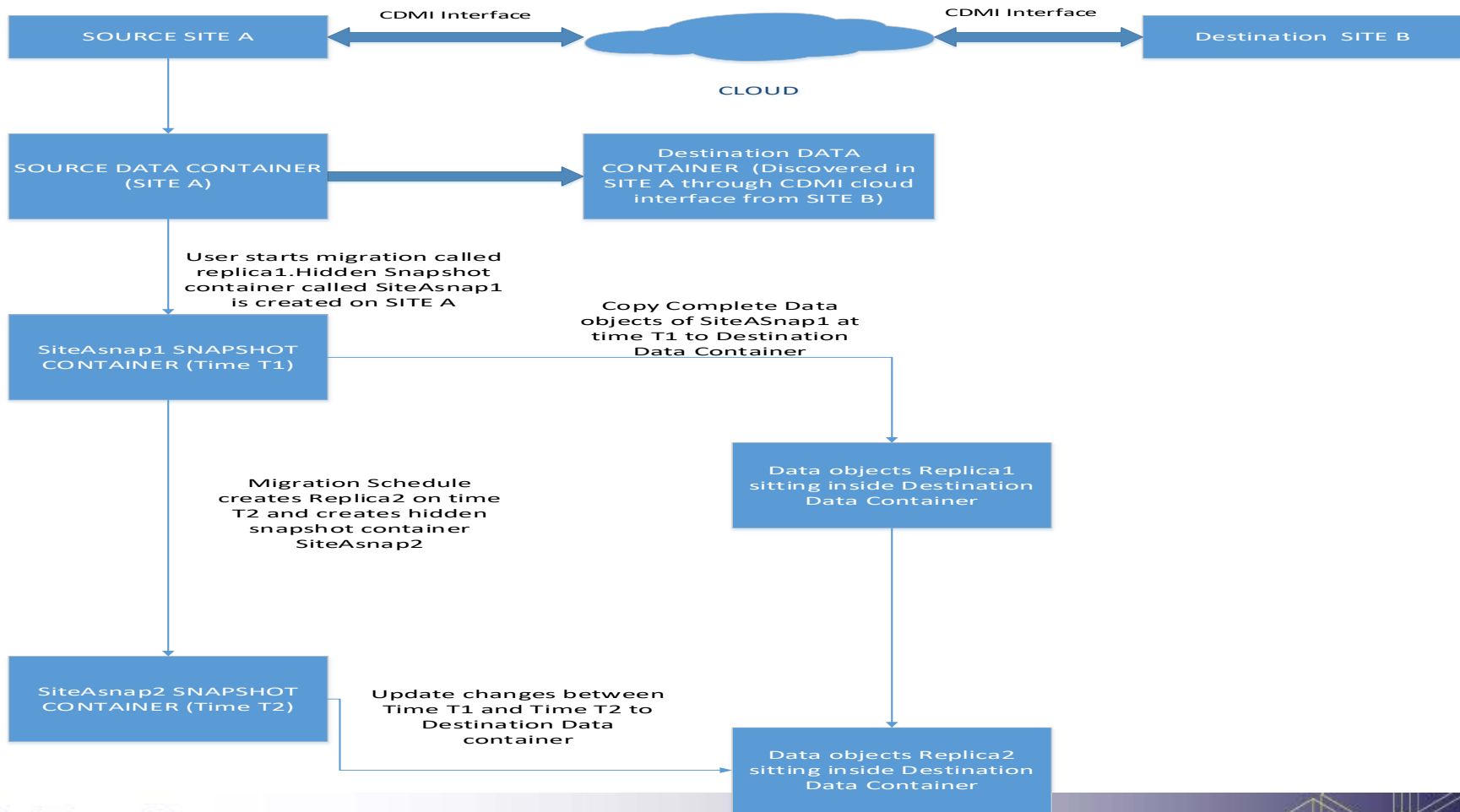
Writing each Data object read before to Destination Container (Data+Metadata)(PUT OPERATION)

EXAMPLE Perform a PUT to the new data object URI:

```
PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.2
```

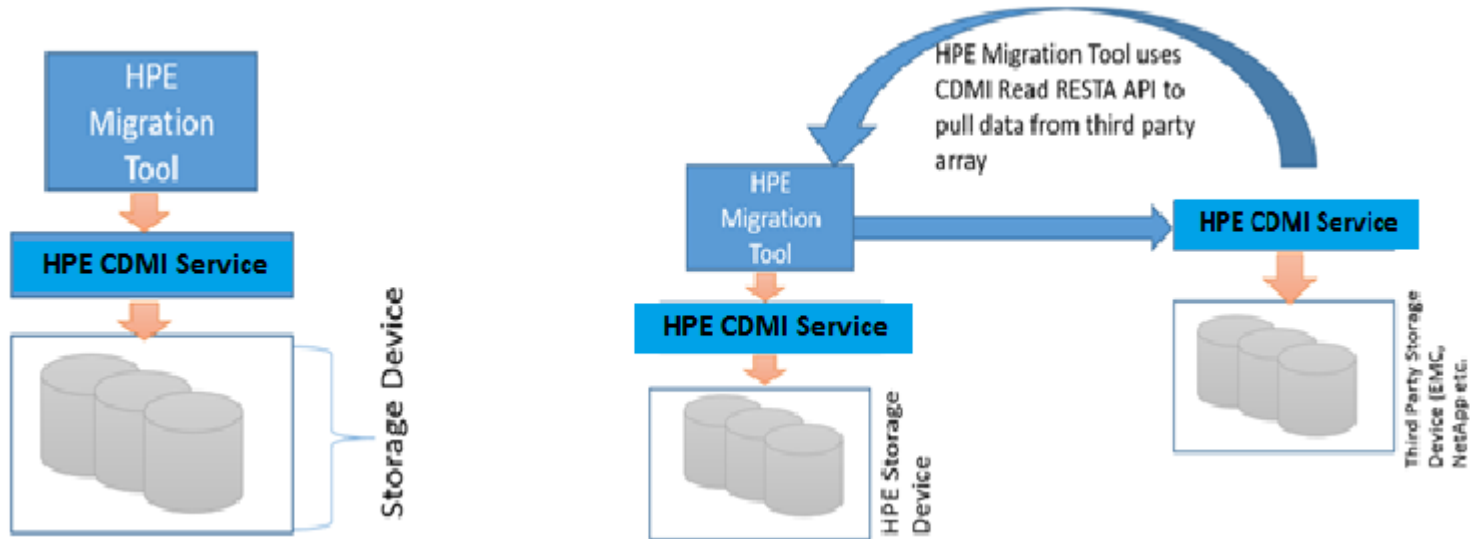
```
{
  "mimetype" : "text/plain",
  "metadata" : {

  },
  "value" : "Hello CDMI World!"
}
```

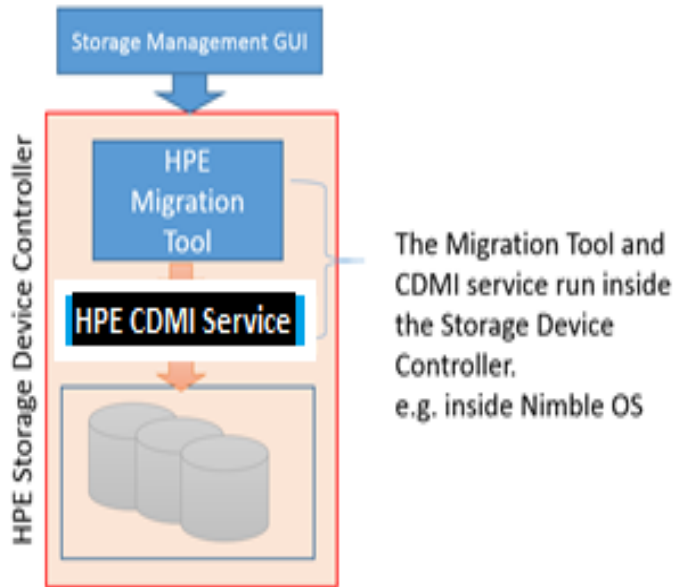


Example: Nimble to 3PAR, Use Case 1

Migration and CDMI Service outside as a software



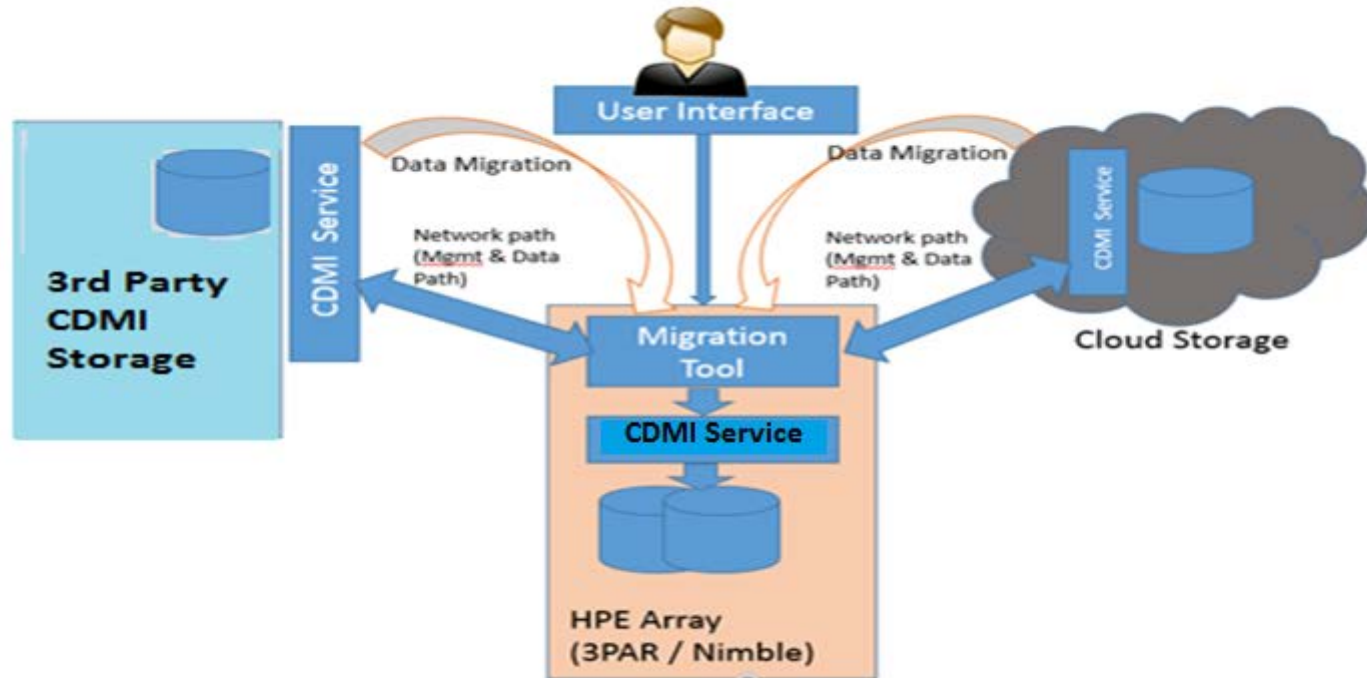
Example: Nimble to 3PAR, Use Case 2



Migration tool engine as a native component to the array. This provides a major usability advantage, saving the customer from having to dedicate a host and deploy a tool.

Storage management GUI providing the GUI options to migrate data from other storage devices, this along with the built-in tool provides major advantage to the customer.

Migrating Across Storage Boundaries



References

- ❑ CDMI Specification v1.1.1

- ❑ SNIA SDC 2014

https://www.snia.org/sites/default/files/SDC15_presentations/cloud/DavidSlik_Using_CDMI_%20Management.pdf

Thank You

Backup