Storage Technology and Standards Trends

Raymond A. Clarke
Sr. Enterprise Storage Solutions Specialist,
Sun Microsystems - Archive & Backup Solutions
SNIA Data Management Forum, Board of Directors
100 Year Archive Task Force

Objectives

- Produce a “best practices for long-term digital information retention” reference model similar to the Sedona project or OAIS
- Solve the storage-side of physical and logical migration
- Teach ILM-based practices for long-term digital information retention
What is SNIA Doing About All This?

//www.snia.org

- Educates, Defines and Taking Action to Address Industry Challenges
- Specific Activities
  - 100 Year Archive Requirements Survey
    - http://www.snia.org/forums/dmf/programs/ltacsi/100_year/
  - XAM – eXtensible Access Method
  - Self-Contained Information Retention Format (SIRF)
    - Rationale & Objectives
    - Requirements & Use Cases
  - Bridging Terminology
    - http://www.snia.org/forums/dmf/knowledge/term_bridge/
  - Green Storage Initiative
  - Cloud Storage Initiative (CSI)
Introducing the New Cloud Data Management Interface

Standardizing the Cloud for Interoperability

Mark A. Carlson
Sun and SNIA Technical Council
Chair, SNIA Cloud Storage TWG
Agenda

• Some background on cloud storage
• The Cloud Storage Reference Model
• CDMI – the interface
  – Data Objects
  – Containers
  – Accounts
  – Capabilities
  – Queues
  – Data System Metadata
SNIA Cloud Storage TWG

• Launched April 2009
  – 140 Technical Work Group members (50 active)
  – Google group for broader community (276 members):
    http://groups.google.com/group/snia-cloud
• Published first documents June 2009
  – Use Cases/Requirements, Reference Model
  – Public web page http://snia.org/cloud
• Draft of Cloud Data Management Interface (CDMI)
  – Targeted at ANSI and ISO certification
• Working on a CDMI Reference Implementation
  – Portable, works on any filesystem
SNIA Cloud Storage Initiative

- Launching at Fall SNW 2009
  - Planning press release listing charter members
  - Cloud Pavilion on show floor

- Supporting the development and adoption of CDMI, Cloud Storage

- Marketing, Outreach, Education on Cloud Storage

- Requirements gathering

- Premier
A look at some existing Cloud APIs

• What are some of the offerings and their Data Storage Interfaces?
Leveraging the Storage Industry Resource Domain Model

All of these interfaces support some or all of this model. The key to retaining the simplicity of the cloud, however, is in the use of metadata to drive the underlying services so that users need not manage the services themselves.
Clients can be in the cloud or enterprise and provide additional services (computing, data, etc.)

Management of the Cloud Storage can be standalone or part of the overall management of your cloud computing.

Clients acting in the role of using a Data Storage Interface

*Block Storage Client*
*Exports to Cloud Computing*
*iSCSI, FC, FCoE LUNs, Targets*

*Filesystem Client*
*POSIX (NFS, CIFS, WebDAV)*

*Object Storage Client*

*SNIA Cloud Data Management Interface (CDMI)*

*Database/Table Client*

*SNIA Cloud Data Management Interface (CDMI)*

*Data Storage Cloud*

*Draws Resources on Demand*

*Data/Storage Management Client*

*Data Services*

*Storage Services*

*Information Services (future)*

*Cloud Data Management*
Cloud Data Management Interface

- Applicable to three types of Cloud Storage:
  - Cloud Storage for Cloud Computing
    - Whitepaper at snia.org/cloud – the management interface for the lifecycle of storage in a compute cloud
  - Public Storage Cloud
    - Both a Data Path for the Cloud and a Management Path for the Cloud Data
  - Private Cloud Storage
    - As well as hybrid clouds
    - An API for Storage Vendors selling into Cloud based solutions

- Semantics
  - Simple Containers and Data Objects with tagged Metadata
  - Data System Metadata expresses the data requirements

- Protocol
  - RESTful HTTP as “core” interface style
  - JSON (JavaScript Object Notation) – format of the representations are extensible
CDMI Specification

• First public draft posted today:
  – Version 0.8 – work in progress for public comment
• Entire specification is 90 pages
  – Intent is simplicity!
• Cloud Storage TWG is working towards a 1.0 release next year
  – Join us…
Overview

- CDMI Basic flow:
  - CDMI Client issues requests
    - HTTP: PUT, GET, HEAD, DELETE
      - MimeType: application/...cdmi.
      - dataobject, container, account, capability
      - Data, Metadata
  - CDMI Implementation issues response
    - HTTP Status (200 OK, 201 Created, etc.)
      - MimeType: application/...cdmi.
      - dataobject, container, account, capability
      - Data, Metadata
Model for the Interface

The resources which are accessed through the RESTful interface

- Root https://<offering>:
  - Container A https://<offering>/containerA:
    - Key Value
    - Key Value
    - ...
  - DataObject1 https://<offering>/containerA/databoject1:
    - Key Value
    - Key Value
    - ...
  - DataObject2 https://<offering>/containerA/databoject2:
    - Key Value
    - Key Value
    - ...
- Capabilities https://<offering>/Capabilities:
  - Key Value
  - Key Value
  - ...
- Accounting https://<offering>/Accounting:
  - Key Value
  - Key Value
  - ...
CDMI Overview

• Chapter 5 – executive overview, based on earlier reference model
• Chapter 6 – some quick examples to get you started on coding
• Chapter 7 – the actual interface
  - 7.3 – Data Objects, 7.4 – Containers, 7.5 – Accounts, 7.6 – Queues, 7.7 – Capabilities, 7.8 – Import/Export Serialization
• Chapter 8 – the metadata
• Appendices - Implementation Compliance, Clients
Data Object Example

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": "AAAAFAA07EFMb3ULbSBpcHN1bSBkb2xvciBzaXQgYW1ldCBhbWV0Lg==",
    "parentURI": "/MyContainer",
    "accountURI": "/cdmi_accounts/MyAccount"}
Get the Data

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
Getting the Metadata

HEAD from the data object URI

HEAD: /MyContainer/MyDataObject.txt
Host: cloud.example.com
Accept: application/vnd.org.snia.cdm.dataobject+json
Content-Type: application/vnd.org.snia.cdm.dataobject+json
X-CDMI-Specification-Version: 1.0

The response looks like:

HTTP/1.1 200 OK
Content-Type: application/vnd.org.snia.cdm.dataobject+json
X-CDMI-Specification-Version: 1.0
{
    "objectURI" : "/MyContainer/MyDataObject.txt",
    "objectId" : "AAAAFAA07EFMb3JlbSBpcHN1bSBkb2xvciBzaXQgYW1ldCBhbWV0Lg==",
    "parentURI" : "/MyContainer",
    "accountURI" : "/cdmi_accounts/MyAccount",
    "capabilitiesURI" : "/cdmi_capabilities/dataobject",
    "percentageComplete" : "Complete",
    "mimetype" : "application/txt",
    "metadata" : []
}
CDMI Containers

• Containers are an abstract place to put “data”
• A Container can contain Data Objects and/or can be “exported” as a block based volume or filesystem
• Containers have data system metadata that specifies the requirements for the data contained in them (rather than an explicit “configuration”)
• Data system metadata in CDMI is inherited from parent containers to child containers and data objects.
• Containers can also be serialized and stored into a data object.
Create a Container

PUT to the URI the container name and other metadata

```plaintext
PUT: /MyContainer
Host: cloud.example.com
Accept: application/vnd.org.snia.cdmis.container+json
Content-Type: application/vnd.org.snia.cdmis.container+json
X-CDMI-Specification-Version: 1.0
{
    "metadata": [],
}
```

The response looks like:

```
HTTP/1.1 201 Created
Content-Type: application/vnd.org.snia.cdmis.container+json
X-CDMI-Specification-Version: 1.0
{
    "objectURI": "/MyContainer",
    "objectId": "AAAAFAAo7EFM3J1bSBpcHN1bSBkb2xvciBzaXQgYW1ldCBhbWV0Lg==",
    "parentURI": "/",
    "accountURI": "/cdmi_accounts/MyAccount",
    "capabilitiesURI": "/cdmi_capabilities/container",
    "percentageComplete": "Complete",
    "metadata": [],
    "children": [],
}
```
Exporting Containers

• The export of a container, via data path protocols other than CDMI, is done by creating or updating a container and supplying one or more export structures, one for each such protocol.
  • The elements of the export structure include:
    • The protocol being used
    • The identify of the container as standardized by the protocol
    • The list of who can access that container via that protocol, identified as standardized by that protocol (may leverage the CDMI accounting for this)
• CDMI standardizes several export structures for various protocols. Export structures can also be defined for proprietary and vendor extensions of
CDMI in Cloud Computing

A single cloud computing infrastructure can implement both the OCCI and CDMI interfaces.

The infrastructure abstracts the configuration of the networking and virtual machine details and uses the standard interface merely to define connectivity.

A cloud computing client can then utilize the interfaces to both specify the data requirements and then use that data for guests.

Get white paper at snia.org/cloud
CDMI Capabilities

• Capabilities define what storage operations a CDMI provider is capable of providing.

• Contrast with permissions, which define what storage operations a CDMI provider will permit a user to perform.

• Capabilities are static for a given cloud storage system, but different sets of capabilities may be present for different URIs.
Questions

• Thank you!