CDMI Specification for Developers

Mark Carlson
SNIA TC and Oracle
Agenda

- Some background on cloud storage
  - Why Cloud Storage?
  - What Is Cloud Storage Used For?

- SNIA Cloud Efforts
  - Cloud Storage TWG
  - Cloud Storage Initiative

- The Cloud Storage Reference Model
  - Existing Cloud APIs
  - Leveraging the Storage Industry Resource Domain Model
  - The Big Picture
  - Using a RESTful protocol
  - Why an Industry Standard?

- CDMI Status

- Object Model, Deployment possibilities
What Is Cloud Storage Used For?

- Elastic demand for web based media (video, eBooks, audio)
- Backup to the cloud
- Internet “Drive” secondary storage
- Sync of files to the cloud and multiple devices
- Archive to the cloud
  - Including Compliance, Retention and eDiscovery
- Storage for Cloud Computing
- Cloud Application storage
SNIA Cloud Storage TWG

- Launched April 2009
  - 175 Technical Work Group members (50 active)
  - Google group for broader community (350 members): [http://groups.google.com/group/snia-cloud](http://groups.google.com/group/snia-cloud)
- Published first documents June 2009
  - Use Cases/Requirements, Reference Model
  - Public web page [http://snia.org/cloud](http://snia.org/cloud)
- Cloud Data Management Interface (CDMI) v 1.0
  - Targeted at ANSI and ISO certification
- Working on a CDMI Reference Implementation
SNIA Cloud Storage Initiative

- Launched at Fall SNW 2009
  - 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 Organization promoting Cloud Storage and associated Standards
What is Cloud Storage?

- The use of the term **cloud** in describing these new models arose from architecture drawings that typically used a cloud as the dominant networking icon.

- The cloud conceptually represented any to any connectivity in a network, but also an abstraction of concerns such as the actual connectivity and the services running in the network that accomplish that connectivity with little manual intervention.

![Diagram of data storage in the cloud](image)
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.
Cloud Storage Container

- Cloud Storage may be used similar to a volume/filesystem
- DSI Protocols include: WebDAV, NFS, CIFS, iSCSI, OSD, others
- Existing Management interfaces: proprietary, Web UI
- Billing based on allocated space, Data Requirement (DR) parameters
- Resource guarantee (desired and required), consumption
- Configuration of DR is an object oriented hierarchy from containers on down to individual data elements
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 Managing Data/Storage

The Complete Picture

Clients acting in the role of using a Data Storage Interface

- Block Storage Client
- Filesystem Client
- Object Storage Client
- XAM Client
- Database/Table Client

SNIA Cloud Data Management Interface (CDMI)

POSIX (NFS, CIFS, WebDAV)

Multiple, Proprietary Interfaces

Draws Resources on Demand

iSCSI, FC, FCoE LUNs, Targets

Data/Storage Management Client

SNIA Cloud Data Management Interface (CDMI)

Cloud Data Management

Data Services

Storage Services

Information Services (future)
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
    - Whitepaper at snia.org/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
REST

- **Representation State Transfer**
  - Started with [Dissertation by Roy Fielding](#) outlining the principles

- **Addressability**
  - Every object (resource) is addressable through a unique identifier

- **Uniform, Constrained Interface**
  - Use only HTTP verbs and model other semantics in the data model
  - Allows for Familiarity (low learning curve), Interoperability and Scalability

- **Representation Oriented**
  - Complexity is in the representations

- **Communicate Statelessly**
  - No persistent client-server connections
Why a RESTful approach for a Cloud Storage Standard

- Simplicity Rules!
- Common Infrastructures in many Languages on many Platforms
- Low learning curve leads to developer adoption
- Developer adoption creates eco-system around API
- Eco-system eases adoption by vendors and customers
Cloud Peering

Models for Cloud Ecology

Cloud Federation

Computing Cloud

Object Storage Cloud

Object Storage Cloud

Multiple Distribution Points

Cloud Peering

Data Usage

Data Usage

Cloud Peering
Why not just a common library?

- There are several common Cloud Libraries available
  - Libcloud (python), Jclouds (Java), Simple Cloud (PHP), etc.
- They all write adapters from the common library to each of the proprietary interfaces
- Adapters must be maintained as interfaces evolve
- Library is under control of 3\textsuperscript{rd} party, so vendors not likely to support directly
- Each language ends up propagating it’s own common library with no common semantics between them
Why not just adopt one of the existing interfaces?

- Despite the “open” licensing of several existing cloud storage interfaces, they all remain under the change control of a single vendor.
- No cloud vendor wants to have a competitor have change control over their interface.
  - Thus they release their own interface which they do have change control over.
- This leads to the propagation of multiple interfaces, each essentially locking developers/customers into that service.
- CDMI is under change control of a standards body, accommodates requirements from multiple vendors and can be extended for proprietary functions.
CDMI Specification

- CDMI is now a SNIA Architecture (standard):
  - [http://snia.org/cloud](http://snia.org/cloud)

- CDMI reference implementation
  - Java source code working draft available
CDMI Basic flow:

CDMI Client issues requests

HTTP: PUT, GET, HEAD, DELETE
  MimeType: application/...cdmi.
  dataobject, container, queue, account, capability
  Data, Metadata

CDMI Implementation issues response

HTTP Status (200 OK, 201 Created, etc.)
  MimeType: application/...cdmi.
  dataobject, container, queue, account, capability
  Data, Metadata
Model for the Interface

The resources which are accessed through the RESTful interface

- Root
  - https://<offering>/
- Capabilities
  - https://<offering>/Capabilities
  - Key Value
  - Key Value
- Container A
  - https://<offering>/containerA/
  - Key Value
  - Key Value
  - ......
- Container B
  - https://<offering>/containerB/
  - Key Value
  - Key Value
  - ......
- Accounting
  - https://<offering>/Accounting
  - Key Value
  - Key Value
  - ......
- DataObject1
  - https://<offering>/containerA/databoject1/
  - Key Value
  - Key Value
  - ......
- DataObject2
  - https://<offering>/containerA/databoject2/
  - Key Value
  - Key Value
  - ......
- Queue
  - https://<offering>/containerB/queue1/
  - 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
- Chapters 7-15 – the core interface
  - 8 – Data Objects, 9 – Containers, 10 – Accounts, 11 – Queues, 12 – Capabilities, 13 – Exported Protocols, 14 – Snapshots, 15 – Serialization/Deserialization
- Chapter 16 – the metadata
- Chapter 17 – logging
CDMI deployment

- Small Private Cloud
  - Deployed as a layer above NAS box, or may also be embedded
- CDMI Containers and Objects are mapped to a mounted filesystem’s directories and files
- CDMI can also be used to configure NAS storage not available through CDMI data path
CDMI deployment

- **Large Scale out Cloud**
  - Deployed as a horizontal set of parallel filesystem clients with requests balanced across them
  - Storage is implemented by a set of data servers with a common metadata server
For More information

- One Web Site to Remember: [http://snia.org/cloud](http://snia.org/cloud)
- Large Cloud Storage Community
  - [http://groups.google.com/group/snia-cloud](http://groups.google.com/group/snia-cloud)
  - [http://twitter.com/SNIAcloud](http://twitter.com/SNIAcloud) (@SNIAcloud)
  - [http://www.google.com/profiles/SNIAcloud](http://www.google.com/profiles/SNIAcloud)