

CDMI Extensions

David Slik
NetApp, Inc.

Session Agenda

- ❑ The Cloud Storage TWG at SNIA
- ❑ The Standardization Process
- ❑ Extensions published for public review
- ❑ Areas of future work

CDMI Specification Status

- ❑ Technical Working Group founded in 2009
 - ❑ Published TWG Charter and Use Cases

- ❑ CDMI 1.0 ratified as a US Technical Architecture in late 2010
 - ❑ CDMI 1.0.1 errata released in late 2011
 - ❑ CDMI 1.0.2 errata released in mid 2012

- ❑ The SNIA has submitted CDMI 1.0.2 to the ISO for international standardization as ISO/IEC 17826
 - ❑ Final DIS ballot in progress

- ❑ CDMI 1.1 is under active development
 - ❑ 13 CDMI Extensions submitted and published



The Cloud Storage TWG Charter

The Cloud Storage Technical Work Group (TWG) is created for the purpose of developing SNIA Architecture related to system implementations of cloud storage technology. The Cloud Storage TWG:

1. Acts as the primary technical entity for the SNIA to identify, develop, and coordinate systems standards for cloud storage
2. Produces a comprehensive set of specifications and drives consistency of interface standards and messages across the various cloud storage related efforts
3. Documents system-level requirements and shares these with other cloud storage standards organizations under the guidance of the SNIA Technical Council and in cooperation with the SNIA Strategic Alliances Committee

The TWG will also assist and cooperate with other SNIA TWGs in their efforts to incorporate cloud storage into their standards work.

- Updates to the CDMI standard
 - Errata releases (e.g. 1.0.2) are intended to correct minor issues in the CDMI standard and do not break backwards compatibility.
 - Minor releases (e.g. 1.1) are intended to refine existing functionality and include new extensions while not breaking backwards compatibility.
 - Major releases (e.g. 2.0) are intended to add significant new functionality to the standard, which may break backwards compatibility.

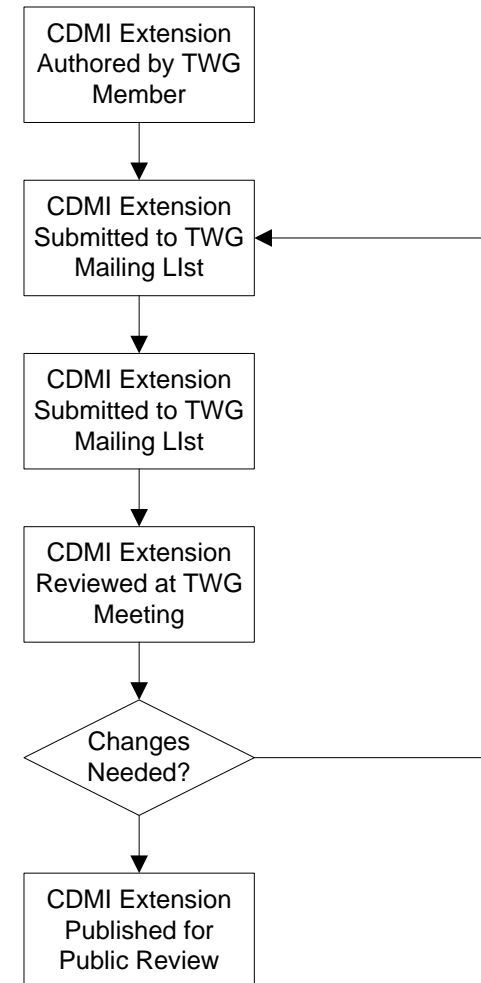
- ❑ When errata or changes to the standard text that improves clarity without changing functionality:
 - ❑ Tickets are opened in the TWG Trac server,
 - ❑ Changes to the specification text is proposed,
 - ❑ Proposed changes are voted on by the TWG,
 - ❑ Changes are then incorporated into the next release of the specification.

- ❑ When new functionality or changes to existing functionality is proposed:
 - ❑ TWG members bring forward “CDMI Extensions”

- ❑ A CDMI extension document is a “diff” of text changes to the specification
- ❑ A CDMI extension shall be sufficiently detailed to be able to be directly incorporated into the spec
- ❑ Recommended best practices for CDMI extensions include:
 - ❑ Introducing a single or a closely related group of functionality
 - ❑ Not depending on any other extensions

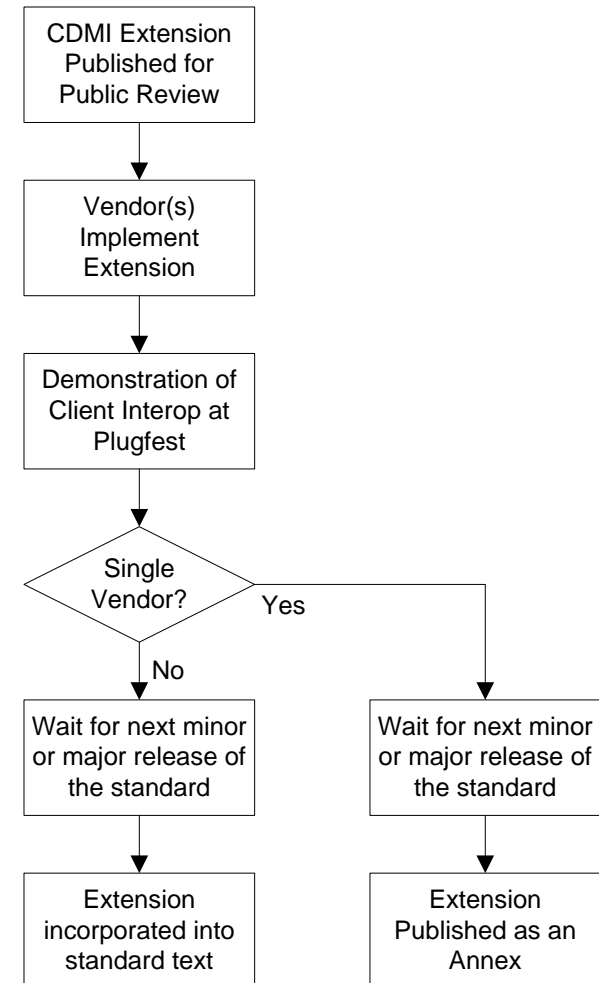
CDMI Extension Process (1/2)

- ❑ Any TWG member can introduce an extension
- ❑ The draft extension is uploaded to the SNIA TWG mailing list for review
- ❑ After two weeks of review, the extension is discussed at a meeting (either on our weekly teleconference or at a face-to-face meeting)
- ❑ The TWG reviews each draft extension, and recommends changes.
- ❑ When a consensus is reached that the extension is ready, the TWG votes to publish the extension for public review.



CDMI Extension Process (2/2)

- ❑ Once published, a CDMI extension must be implemented by at least one vendor
- ❑ When client interoperability with a vendor's implementation of the CDMI extension is demonstrated at a plugfest, the extension is published as an annex in the next minor or major release of the CDMI standard.
- ❑ When client interoperability of two or more vendors' implementations is demonstrated at a plugfest, the extension is directly incorporated into the text of the next minor or major release of the CDMI standard.



Extensions Published for Review

http://snia.org/tech_activities/publicreview/cdmi

- ❑ CIMI Extension
- ❑ Copy Range Extension
- ❑ Data Affinity Extension
- ❑ Domain Auth Methods Extension
- ❑ Domain Summary Extension for Bandwidth
- ❑ Expiring ACE Extension
- ❑ Fixity Metadata Extension
- ❑ Group Storage System Metadata Extension
- ❑ Jobs Extension
- ❑ Maximum Value Transferred Extension
- ❑ Multi-part MIME Extension
- ❑ OVF Extension
- ❑ Versioning CDMI Extension

- ❑ Defines a CDMI export type for CIMI systems, such that a CIMI-based VM can easily discover and access to LUNs stored on a CDMI storage system
- ❑ Similar to what is provided for OCCl systems in the 1.0.2 specification.

Copy Range Extension

- ❑ Specifies how a client can initiate a server-side copy of a range of a data object to another data object.
- ❑ Allows source and/or destination ranges

```
PUT /MyContainer/DestinationDataObject.txt?value:0-9
HTTP/1.1 Host: cloud.example.com
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.1
{
  "copy" : "/MyContainer/SourceDataObject.txt?value:10-19"
}
```

- ❑ Introduces a new data system metadata item for client authentication methods for domain objects
 - ❑ Allows a client to discover what authentication methods are supported by the system
 - ❑ Allows a client to request which authentication methods should be allowed for a given domain
 - ❑ Allows a client to discover which authentication methods are provided for a given domain
- ❑ Currently defined authentication types are “anonymous”, “basic”, and “digest”

Bandwidth Domain Summary

- ❑ Introduces new domain summary attributes for public and private network data transfers
- ❑ Permits reporting on a cumulative, daily, monthly and yearly basis of:
 - ❑ Total bytes sent and received
 - ❑ Private and public network bytes sent and received
 - ❑ Private and public network byte transfer rates (min/max/average)

Expiring ACEs Extension

- ❑ Introduces the ability to have an ACE within an ACL have a specified lifespan
- ❑ Each ACE can optionally have a point in time when it will no longer be effective

- ❑ Still some debate within the TWG over how this should be expressed to ensure compatibility with other protocols
 - ❑ Part of ACL or separate metadata item?

Fixity Metadata Extension

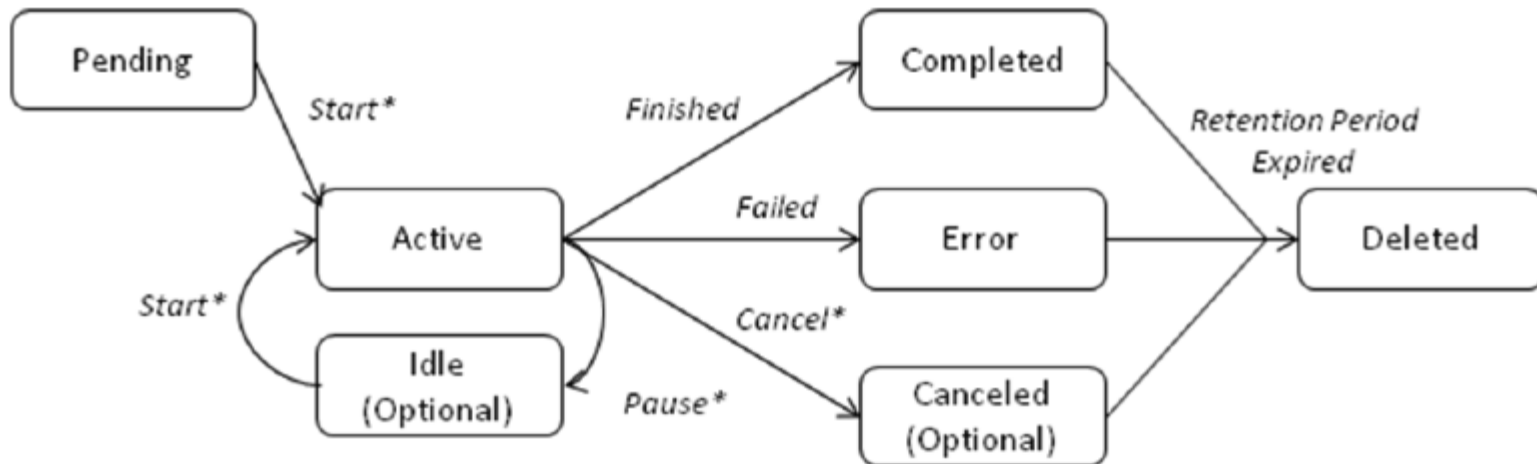
- ❑ Introduces a new data system metadata item that allows clients to specify the maximum age between integrity verification (fixity checks)
- ❑ Introduces new storage system metadata items to allow a client to determine when fixity was last checked by the cloud, and the result of that last check was.

Group Metadata Extension

- Introduces a new storage system metadata item that indicates what group the object belongs to

- ❑ Introduces specialized metadata that can be used to represent and control long-running “jobs” within the cloud.
- ❑ Examples of things that can be represented as a job include:
 - ❑ Bulk deletion, bulk metadata updates, serialization, data migration, virus scanning, search indexing, etc.

- Job objects have a specific state lifecycle:



- The Start, Pause, and Cancel transitions can be initiated by a client

- ❑ The jobs extension also introduces metadata that allows a system to designate a special container where jobs can be enumerated
- ❑ It also introduces metadata that allows a client to discover what jobs can be run, and in which containers the data objects used to manage these jobs can be created.

Max Value Transferred Capabilities

- ❑ Introduces a capability that indicates the maximum size that can be transferred in a PUT, GET, and POST request
- ❑ Introduces a capability that indicates the maximum size of the value of a data object

- ❑ In CDMI, data must be either transferred as a Non-CDMI transaction, or encoded in JSON
 - ❑ As JSON requires UTF-8 clean data, sending binary data requires Base 64 encoding
 - ❑ This is inefficient
- ❑ The Multi-part MIME extension allows CDMI operations to be split into multiple parts, the first containing the CDMI JSON, and the remaining parts containing the value

Multi-part MIME Extension

```
PUT /MyContainer/MyDataObject.txt
HTTP/1.1 Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object
{
  "domainURI": "/cdmi_domains/MyDomain/",
  "metadata": {
    "colour": "blue"
  }
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-Length: 37

<37 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p--
```

Multi-part MIME Extension

- ❑ This extension also improves the efficiency of deserialization, as the serialized representation no longer needs to be JSON escaped to be put into a JSON string
- ❑ This extension also allows multiple byte ranges to be set as part of a single client-initiated transaction
- ❑ Finally, this extension provides the same efficiency gains for queues

- ❑ This extension permits an OVF container to be deserialized into a CDMI container, such that all of the contents of the OVF container becomes CDMI data objects and containers.
- ❑ This permits an OVF file to be deserialized into a CDMI system, then easily exported to a OCCI or CIMI managed VM system
- ❑ CDMI containers with a valid OVF manifest can likewise be serialized back into an OVF file

Versioning Extension

- ❑ This extension allows a client to request that modifications to an object retains historical versions, and provides access to these historical versions.
- ❑ Versions are arranged in a tree, as with eventual consistency, multiple writes may occur at disconnected or distant nodes
- ❑ Clients can traverse and reconcile version trees

Versioning Extension

- ❑ This extension allows a client to request that modifications to an object retains historical versions, and provides access to these historical versions.
- ❑ Versions are arranged in a tree, because with eventual consistency, multiple writes may occur at disconnected or distant nodes
- ❑ Clients can traverse and reconcile version trees



Versioning Extension Demonstration

Thank you!

Questions and Answers

Contact Info:
dslik@netapp.com