Compliance Testing of CDMI:
Overcoming challenges & best practices

Udayan Singh, Nishi Gupta, Hansi Agarwal
Tata Consultancy Services
Abstract

Over the past years, interoperability has become more and more of a necessity in standardization and Industry demands delivery of products/services which are interoperable. SNIA addresses lack of interoperability issue of Cloud Storage by Cloud Data Management Interface specification. It tags users data with special metadata (data system metadata) that tells the cloud storage provider what data services (backup, archive, encryption etc) to provide that data. It helps in moving users data from one cloud vendor to another cloud vendor without the pain of recoding of different interfaces. There is a growing trend where organizations are adopting CDMI in their products and hence interoperability becomes extremely important. TCS is working on `CDMI Automated Test Suite`, which focuses on testing compliance to CDMI specifications. In this proposal we will share our observations/challenges during the development of test suite and challenges while testing products for CDMI compliance. This will benefit the companies to adopt best practices while developing CDMI compliant products.
Session Agenda

- Cloud Storage Key challenges / SNIA Focus
- CDMI Automated Test Suite
- Challenges in Development of CDMI Test Suite
- Challenges while testing products for CDMI compliance
- Best Practices
- Conclusion
## Cloud Storage Key challenges / SNIA Focus

<table>
<thead>
<tr>
<th>Current Challenges</th>
<th>SNIA Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>❑ No Standard method for a cloud storage service provider to directly migrate customer data to another provider</td>
<td><strong>SNIA introduces</strong> CDMI to overcome Cloud Storage challenges</td>
</tr>
<tr>
<td>❑ Vendor specific API’s</td>
<td>❑ Through standard interface for clients to communicate with storage clouds</td>
</tr>
<tr>
<td>❑ Unique financial methods of offering Cloud Storage Services to end-customer’s by each vendor</td>
<td>❑ Through a standard approach for adding vendor specific functionality without breaking client compatibility</td>
</tr>
<tr>
<td></td>
<td>❑ Through a standard approach for comparing cloud capabilities &amp; services</td>
</tr>
</tbody>
</table>
# CDMI Automated test Suite

**Overview:** Provides System Compliance Testing of Cloud Storage Server for conformity to SNIA CDMI specifications for different use-cases

## Functionality –

- Automated Compliance testing of CRUD Operations for following:
  - Data Object
  - Container Object
  - Capability Object
  - Domain Object
  - Queue Object

- Supports both TLS/Non-TLS CDMI and Non-CDMI objects
- Platform Independent
- Support for multiple versions of CDMI spec
- Artifacts – Test Plan, Test Spec, Report and log files
Changes in Development of CDMI Test Suite
Challenges in development of CDMI Test Suite

- CDMI Specifications (Continuous Evolving Features)
- CDMI Specifications Challenges
- Requirement of CDMI Server’s / Simulators
- Scenario Creation
Challenges in development of CDMI Test Suite

Requirement of CDMI Server’s / Simulators:

1. Requirement of CDMI test server to test successful implementation of test script
2. Requirement of at least two CDMI servers to test successful implementation of test script for interoperability testing
Challenges in development of CDMI Test Suite

CDMI Specifications (Continuous Evolving Features):

3. Multiple version of CDMI spec
   - Supporting different CDMI specification versions with a single test suite.
   
   Example:
   - Reading Object. Verify that a 200 Ok status code is returned.
   - ‘Accept’ header is mandatory in CDMI_spec_ver 1.0.1, while it is optional in CDMI_spec_ver 1.0.2

4. Incorporation of CDMI extension in the test suite.
Challenges in development of CDMI Test Suite

CDMI Specifications Challenges:

5. Ambiguities in CDMI Specification document

   Example:
   - Validation of request header.
     - Example: Creating data object by passing invalid Content type

6. Implementation of some scenarios due to lack of Examples.

   Example 1:
   - Updating value of Queue object. Verify that 204 No Content is returned.
   - No example to differentiate between enqueue new value / updating existing value

   Example 2:
   - To test creation of big size data object
Challenges in development of Test Suite

CDMI Specifications Challenges (cont.):

7. Vendor specific implementations not specifically mentioned in CDMI Specification document

Example:
- It has not been mentioned whether the creation of snapshot feature is vendor specific or not.
Challenges in development of Test Suite

Scenario Creation:

8. Challenge in testing same scenario with multiple combination of Input parameters

   *Example:*
   - Creating Container by passing invalid fields. Verify that a 400 Bad Request status code is returned.

9. Challenge in identifying Multiple capabilities check in single test scenario:

   *Example:*
   - Deleting Data object with the help of object id. Verify that a 204 No Content status code is returned.

   Following Capabilities should be supported to perform this operation:
   - `cdmi_dataobjects` system-wide capability
   - `cdmi_delete_dataobject`
   - `cdmi_object_access_by_ID` system capability
Challenges in testing CDMI Compliant product
Challenges while testing products for CDMI compliance

**Capability not matches implementation:**

1. Testing of CDMI objects where capability object is not implemented properly

   *Example:*
   
   - “cdmi_dataobjects” capability is not listed while reading capability object. But data objects are supported by CDMI servers.
Challenges while testing products for CDMI compliance

Different interpretations of CDMI Standards:

2. Incorrect response status in response of invalid header fields.
   
   \textit{Example}:
   
   - Some server returns 406 and some returns 404

3. Vendor Specific implementation
   
   \textit{Example 1}:
   
   - Server specific response of Reading Capability object in two ways
     
     Server A: “cdmi_queues” : true
     Server B: “cdmi_queues” : “true”

   \textit{Example 2}:
   
   - Validation of single slash(\/) and double slash(//) in URI, taken differently by servers.
     
     \textit{Example}:
     
     Request URI: GET <root URI>//<container name>//<data object name> is valid for Server A, while invalid for Server B
Challenges while testing products for CDMI compliance

Multi Version implementation:

4. Same server implemented with two versions

   Example:

   - Server implemented with 1.0.1, also implemented some part of CDMI spec 1.0.2.
Best Practices
Best Practice

- Capability object should be properly implemented by CDMI servers
  - This would ensure execution of all required test cases based on capability.

- Common bugs should be taken care of while implementing CDMI Server
  - Response Body Error:
    - Presence of mandatory parameters and their correct values
  - Directory name:
    - Should be followed by a "/"
Vendor’s should adopt methods so to have a similar type of implementation of CDMI server.
  - Each vendor should return response in JSON format as per cdmi spec

Use Simulator’s
Conclusion

- Cloud Storage is a hot Industry trend and as we move forward CDMI is the right step towards addressing end customer’s requirements.
- There are challenges in Testing CDMI Compliance, e.g. due to evolving nature of CDMI, these can be mitigated by adopting some of the proposed best practices.
- Use of Vendor agnostic Compliance Test Tool for testing of Cloud Storage Products for conformance to CDMI, will enable implementation of the CDMI standard in right manner.
- This can help in building interoperable client and server.
Questions?

Mail us @
udayan.singh@tcs.com

Special Thanks to Reena Dayal, Nishi Gupta and Hansi Agarwal for making this presentation possible
Thanks 😊