The Cloud Data Management Interface Reference Implementation

Scott Baker
September 20th, 2010
CDMI is an interoperable & vendor neutral interface for cloud offerings consisting of:

- A Data path
- A Management / control path
The CDMI Reference Implementation:

- Facilitates client development
- Facilitates conformance testing of production implementations.

- Work-in-progress release available now from SNIA
Agenda

- CDMI overview
- CDMI Reference Implementation
  - Architecture
  - Design
- Reference Implementation Code
- Demonstration
CDMI Specification

- Developed by the SNIA Cloud TWG
- CDMI v1.0 an approved SNIA Technical Position
CDMI Protocol

- REST based interface.
- Stands for Representational State Transfer
- Attributes:
  - Resources are addressed directly by unique URIs
  - Operations constrained to a few HTTP basic verbs (GET, PUT, DELETE)
    - State accessed/manipulated in message body representation
CDMI - Model

- Provides for Create, Delete, Update, & Delete of data (CRUD)
- Model is a hierarchy of containers with data objects as leaves.
- Data objects are files, pictures, documents, etc
- Objects have attached metadata for QoS, etc
- Objects can have client/user metadata defined/attached
CDMI - Model

- Capabilities (Quality of Service) of a CDMI offering are modeled as a hierarchy of pseudo-containers

- Domains – Provide account functionality
CDMI – Model - Capabilities
Reference Implementation Architecture

- Design provides for different implementations
- Current Work-in-progress series implemented using underlying local filesystem
- Alternative for a control path implementation would be an SMI client proxy.
  - Attend the CDMI and SMI-S presentation
  - Tuesday 4:05PM
Reference Implementation Architecture

CDMI Reference Implementation Architectural Diagram

- Green: SNIA Developed Code
- Blue: 3rd Party Code

Date: July 20, 2010
RI Architecture

* File System Naming:
  Container Objects = Folders named with the container name
  Data Objects = Files named with the object name, if one was given, else the Object ID
  Metadata = Files named with the same name as the corresponding object with an additional "." in front

Examples:
  Container: /mnt/cdmi server/MyContainer
  Container Metadata: /mnt/cdmi server/.MyContainer

  Data Object's Data: /mnt/cdmi server/MyContainer/MyDataObject.txt
  Data Object's Metadata: /mnt/cdmi server/MyContainer/.MyDataObject.txt

  Data Object's Data: /mnt/cdmi server/MyContainer/0000706D0010B84FAD185C425D8B537E
  Data Object's Metadata: /mnt/cdmi server/MyContainer/0000706D0010B84FAD185C425D8B537E
RI Architecture

Mapping of CDMI Model to Filesystem:

- Containers to Directories
- Data Objects to Files
- Capabilities to hidden files
- Domains not yet implemented
RI Architecture

- Infrastructure information is stored in “.” files in parent directory
  - CDMI “Fields”
  - CDMI “Metadata”
- Stored directly in JSON format to leverage serialize/deserialize functionality of Model classes
- Capabilities hierarchy in a editable “.” file
Reference Implementation Design

- Uses the JAX-RS API Standard
- Annotations on Classes and Methods provides for:
  - URI Matching and Routing to methods
  - Parameter injection (URI segments)
  - Content-type handling (e.g.) container vs dataobject)
RI Design

- Java

- Uses Apache CXF framework implementation of JAX-RS
Framework support (web.xml). Links to Apache CXF framework servlet and specifies url pattern to route to CXF servlet.

- `<servlet>`
  - `<servlet_name>`

- `<servlet-mapping>`
  - `<url-pattern>`
RI Design

- CXF Framework support -ApplicationContext.xml
  - Identification and location of REST resource service classes including point of entry
  - Identification of injection points and current injection values
Design – Build Environment

- Maven
- Standardized directory structure
- `pom.xml` file specifies internet repositories for satisfying library dependencies
- Netbeans used by CDMI RI developers
Design – Module Breakdown

- Model classes
- REST resource classes
- DAO (Data Access Object) pattern
  - Interfaces
  - Implementation classes
Design – Model Classes

- Contains serialization/deserialization <> JSON
  - Container
  - Data Object
  - Capability
  - Domain
Design – Resource Classes

- **PathResource class**
  - `cdmi_server/{mycontainer}`

- **ObjectidResource class**
  - `cdmi_server/cdmi_objectid/{object id of mycontainer}`

- **CapabilityResource class**
  - `cdmi_server/cdmi_capabilities`
Design – DAO interfaces & impls

- ContainerDAO
- DataObjectDAO
- CapabilityDAO
- DomainDAO

- Uses java.io and java.io.File to access local filesystem
Demonstration

- OSX HttpClient
- Get CDMI containers, dataobjects, capabilities
- Create container and dataobject
- Inspect changes to underlying local filesystem
Demonstration –
Get root container

URL: http://localhost:8080/cdmi-server/
Method: GET

Header Name | Header Value
--- | ---
Accept | application/vnd.org.snia.cdmi.container+json

Body:

```
HTTP/1.1 200 OK
X-Powered-By: Servlet/3.0
Server: GlassFish v3
X-Cdmi-Specification-Version: 1.0
Date: Wed, 01 Sep 2010 04:26:50 GMT
Content-Type: application/vnd.org.snia.cdmi.container+json
Content-Length: 271
Connection: close

{
  "objectID": null,
  "capabilitiesURI": "/cdmi_capabilities/container/default",
  "domainURI": "/cdmi_domains/default_domain",
  "metadata": {
  },
  "exports": {
  },
  "objectURI": "/",
  "parentURI": "/",
  "children": ["containeralpha/", "dataobject1"]
}
```
Demonstration – Get container

HTTP/1.1 200 OK
X-Powered-By: Servlet/3.0
Server: GlassFish v3
X-Cdmi-Specification-Version: 1.0
Date: Wed, 01 Sep 2010 04:36:06 GMT
Content-Type: application/vnd.org.snia.cdmi.container+json
Content-Length: 444
Connection: close

{
  "objectID": "00000000001896A6366465616634G372D31036362D3461",
  "capabilitiesURI": "/cdmi_capabilities/container/default",
  "domainURI": "/cdmi_domains/default_domain",
  "metadata": {
    "mymetadata": "zz13",
    "cdmi_ctime": "2010-08-31T21:25:33",
    "cdmi_atime": "never",
    "cdmi_mcount": "0",
    "cdmi_acount": "0"
  },
  "exports": {
  },
  "objectURI": "/containeralpha",
  "parentURI": "/",
  "children": []
}
Demonstration – Get object

```
HTTP/1.1 200 OK
X-Powered-By: Servlet/3.0
Server: GlassFish v3
X-Cdmi-Specification-Version: 1.0
Date: Wed, 01 Sep 2010 04:31:44 GMT
Content-Type: application/vnd.org.snia.cdmi.dataobject+json
Content-Length: 531
Connection: close

{   "objectURI": "dataobject1",
    "capabilitiesURI": "/cdmi_capabilities/dataobject",
    "objectId": "0000000001895035363137303933622D613965642D3463",
    "mimetype": "text/plain",
    "metadata": { "mimetype": "text/plain",
        "cdmi_cftime": "2010-08-31T21:26:48",
        "cdmi_atime": "2010-08-31T21:31:44",
        "fileName": "/Users/Scott/snia/cdmi/cdmi-server/dataobject1",
        "cdmi_size": "7",
        "metadataFileName": "/Users/Scott/snia/cdmi/cdmi-server/.dataobject1"
    },
    "valueRange": "7",
    "value": "someval"
}
```
Demonstration – Create Container

URL: http://localhost:8080/cdmi-server/containerbeta
Method: PUT

Header Name: Content-Type
Header Value: application/vnd.org.snia.cdmi.container+json

Body:
```
{
  "metadata": {
    "mymetadata": "z213"
  }
}
```

HTTP/1.1 200 OK
X-Powered-By: Servlet/3.0
Server: GlassFish v3
X-Cdmi-Specification-Version: 1.0
Date: Wed, 01 Sep 2010 04:39:06 GMT
Content-Type: application/vnd.org.snia.cdmi.container+json
Content-Length: 443
Connection: close

```
{
  "objectID": "00000090185B3CA336D33135643465652D3162376222D3430",
  "capabilitiesURI": "/cdmi_capabilities/container/default",
  "domainURI": "/cdmi_domains/default_domain",
  "metadata": {
    "mymetadata": "z213",
    "cdmi_ctime": "2010-08-31T21:39:06",
    "cdmi_mtime": "0",
    "cdmi_atime": "never",
    "cdmi_account": "0"
  }
}
```
Demonstration – Create Object

URL: http://localhost:8080/cdmi-server/dataobject2
Method: PUT

Header Name: Content-Type
Header Value: application/vnd.org.snia.cdmi.dataobject+json

Body:

```json
{
    "value": "someval"
}
```

Request

HTTP/1.1 200 OK
X-Powered-By: Servlet/3.0
Server: GlassFish v3
X-Cdmi-Specification-Version: 1.0
Date: Wed, 01 Sep 2010 04:41:54 GMT
Content-Type: application/vnd.org.snia.cdmi.dataobject+json
Content-Length: 517
Connection: close

Response

```json
{
    "objectURI": "dataobject2",
    "capabilitiesURI": "/cdmi_capabilities/dataobject",
    "objectID": "000000000018E033030638323765362D656234352D3466",
    "mimetype": "text/plain",
    "metadata": {
        "mimetype": "text/plain",
        "cdmi_c_time": "2010-08-31T21:41:54",
        "fileName": "/Users/Scott/snia/cdmi/cdmi-server/dataobject2",
        "cdmi_a_time": "never",
        "cdmi_size": "7",
        "metadataFileName": "/Users/Scott/snia/cdmi/cdmi-server/./dataobject2"
    },
    "valueRange": "7",
    "value": "someval"
}
```
Demonstration – Capabilities

URL: http://localhost:8080/cdmi-server/cdmi_capabilities/
Method: GET

Header Name: Accept
Header Value: application/vnd.org.snia.cdmi.capability+json

Body:
```
HTTP/1.1 200 OK
X-Powered-By: Servlet/3.0
Server: GlassFish v3
Date: Wed, 01 Sep 2010 04:44:26 GMT
Content-Type: application/vnd.org.snia.cdmi.capability+json
Content-Length: 169
Connection: close

{"capabilities":
{"cdmi_metadata_maxitems":"1024","cdmi_metadata_maxsize":"4096","domains":"false","cdmi_export_occi_iscsi":"true"},
"children": ["container", "dataobject"]
}
```
Testing

- Test Client using JUnit
Current Uses

- Deployed at SNIA Tech Center
- An integral part of the joint cloud demo with OGF/OCCI
- Used as reference for the development of Clients
  - iPad
  - Java FX
  - Flex
License

- BSD 3 Clause
- Oracle & SNIA
- Right to modify & redistribute provided entire license/copyright is included
Call for involvement

- We are looking for volunteers to help complete the Reference Implementation!
  - Access by ObjectID
  - Container Move/Copy
  - Domains
  - Queues
  - .....
Questions?

sjbaker@mindspring.com