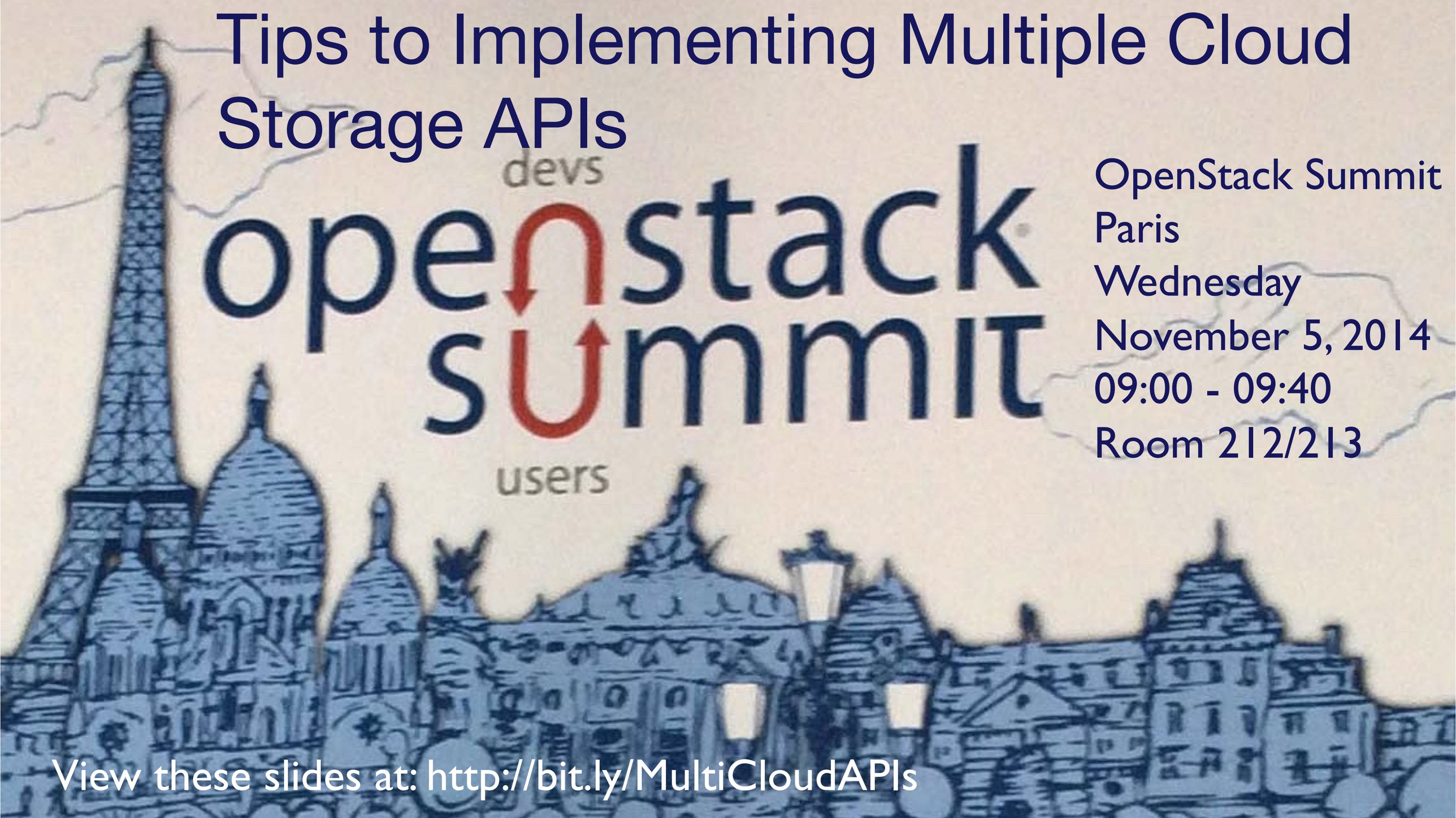
Tips to Implementing Multiple Cloud Storage APIs



OpenStack Summit
Paris
Wednesday
November 5, 2014
09:00 - 09:40
Room 212/213

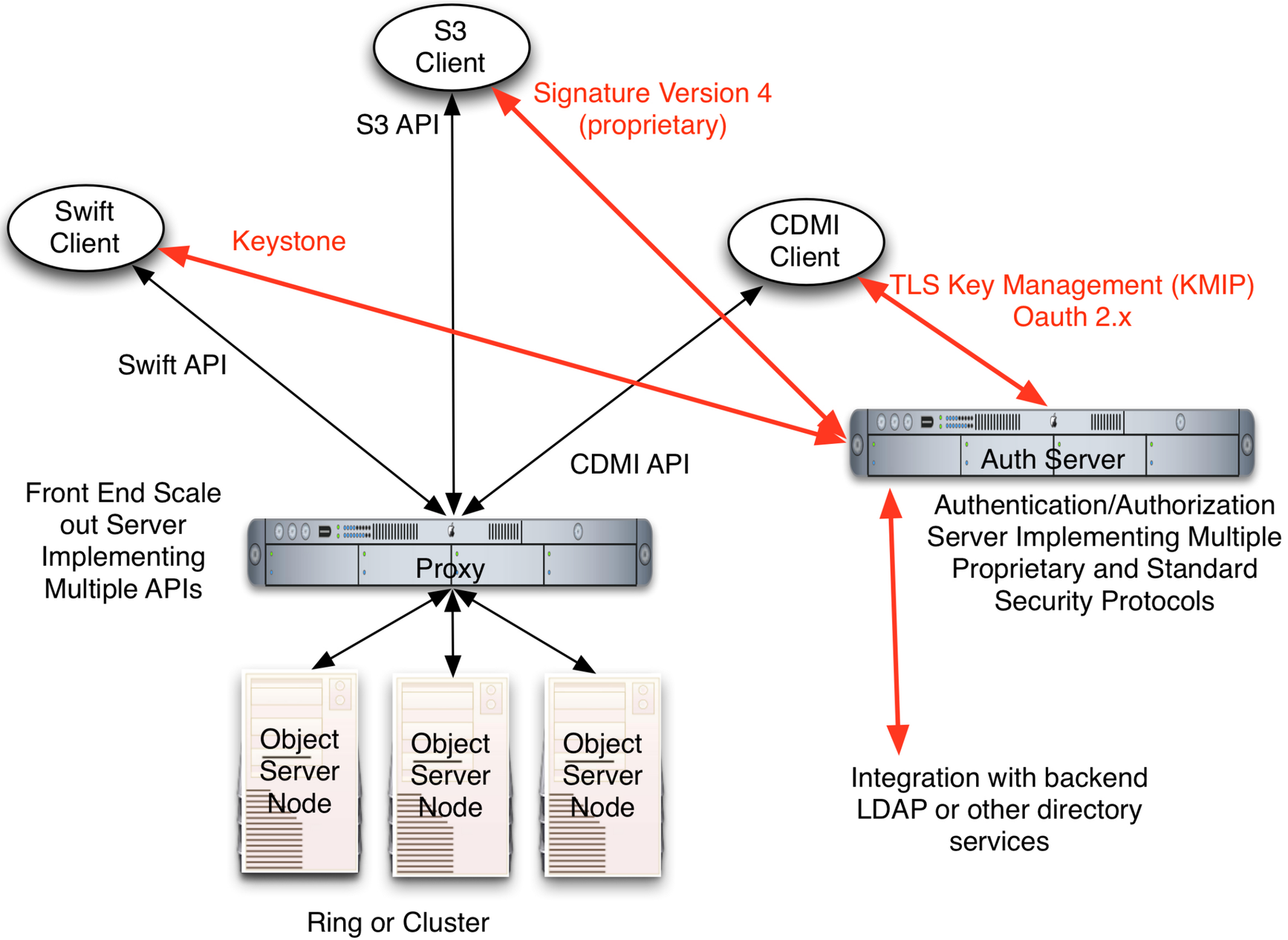
View these slides at: http://bit.ly/MultiCloudAPIs

Abstract

When customers ask for support of a given API, can a vendor survive if they ignore these requests? A strategy many vendors are taking is to support multiple APIs with a single implementation. Besides the Swift API, many support the S3 defacto and CDMI standard APIs in their implementation. What is needed for these APIs to co-exist in an implementation? There are basic operations that are nearly identical between them, but what about semantics that have multiple different expressions such as metadata?

View these slides at: http://bit.ly/MultiCloudAPIs

What does this look like?



Breakdown

Storage Operations

•CRUD – All pretty much determined by HTTP standard (common code)

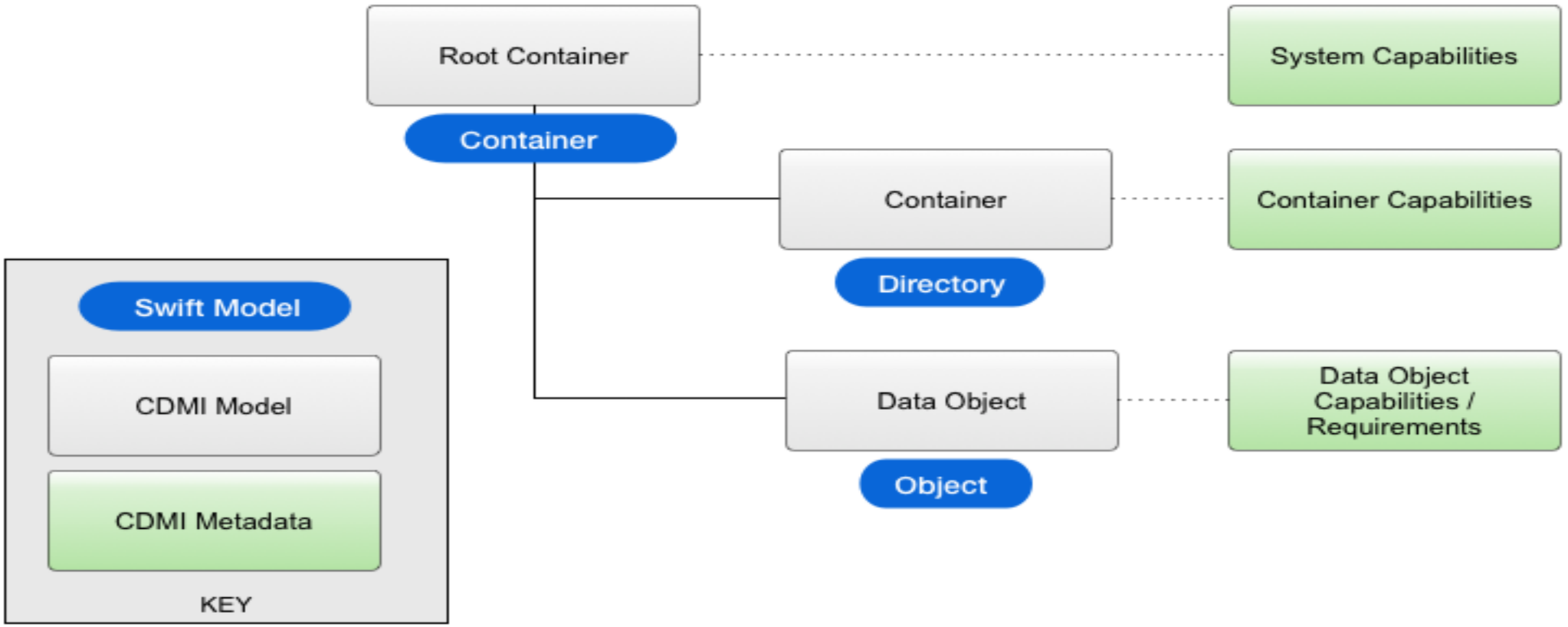
•Headers are API unique however (handle in API specific modules)

Security Operations

•Client communication with Auth Server (API unique)

•Multiple separate services running in Auth Server

Swift and CDMI Models



5

Comparing S3 to CDMI

Function	S3 support	CDMI Support
List the objects in a container	GET Bucket	GET Container
Display simplified access controls	GET Bucket acl	GET Container metadata item named `cdmi_acl`
Display Windows and NFS compatible access controls		GET Container metadata item named `cdmi_acl`
Discover retention autodelete interval for all objects in a mutable container	GET Bucket lifecycle	GET Container metadata item named `s3_lifecycle`
Discover whether a container shall be deleted at the end of its retention period		GET Container `cdmi_retention_autodelete` metadata item
Discover retention data on a container		GET Container `cdmi_retention_period` and `cdmi_retention_start_time` metadata items
Find legal holds that have been placed on a container		GET Container `cdmi_hold_id` metadata item
Discover the policy set on a container	GET Bucket policy	
Discover the geographic location(s) in which a container's data is stored	GET Bucket location	GET Container `cdmi_geographic_placement_provided` metadata item
	(US and EU only)	(Full ISO 3166 support)
Get logging status	GET Bucket logging	GET logging queue metadata
Get status for full featured logging		GET logging queue metadata (CDMI logging features much richer functionality than S3)

Whitepaper "*A CDMI Guide for S3 Programmers*"
Alan Yoder, Huawei

http://snia.org/sites/default/files/S3-like_CDMI_v1.0.pdf

CDMI Object Metadata

2 major classes – System and User

SYSTEM Metadata

- Storage System: Timestamps, Traditional ACLs, Counts etc.
- Data System: requirements of the object – eg. Retention, Backup, Replication, Performance

USER Metadata

- Application and data specific
- E.g. EXIF data on photos
- Location data of objects
- Relationship data
- SEARCHABLE!

7

CDMI Capability Metadata

Describes the capability of a service participating in CDMI cloud environment

- Performance
- Retention capability
- Location information
- Storage features (Compression, Encryption, Hashes etc.)

Shrink to fit development and consumption model

- Advertising capability means that developers only need to implement the standard partially and only advertise what is implemented

8

CDMI and Swift

CDMI works alongside Swift and S3 models, not replacing them

Any of the APIs can be used to access the same data

CDMI has been implemented for Swift as a filter which allows leverage of the Swift authentication filter

- If Swift and CDMI disagree, then Swift CDMI "faults"

Swift and CDMI use hierarchical containers but are slight different in the implementation and language used

Swift Folder = CDMI Container

CDMI Metadata can be stored in Swift metadata storage

This means the size limitation is implied for CDMI metadata currently

9

openstack
CLOUD SOFTWARE

Support for common operations in CDMI 1.1

Latest version of CDMI - **http://www.snia.org/sites/default/files/CDMI_Spec_v1.1.pdf**

- Spec text that explicitly forbid (in 1.0) functionality required for S3/ Swift integration has been removed from the spec (“/”s may create intervening CDMI Containers)
- Baseline operations (mostly governed by RFC 2616) now documented in Clause 6 (pgs. 28-35)
- CDMI now uses content type to indicate CDMI-style operations (as opposed to X-CDMI-Specification-Version)
- Specific authentication is no longer mandatory. CDMI implementations can now use S3 or Swift authentication exclusively, if desired.

openstack
CLOUD SOFTWARE

Discovery of Security Protocol Implementations

CDMI 1.1 now includes a standard means of discovering what methods are available:

•cdmi_authentication_methods (Data System Metadata) ***12.1.3***

•If present, this capability contains a list of server-supported authentication methods that are supported by a domain. The following values for authentication method strings are defined:

• "anonymous"-Absence of authentication supported
• "basic"-HTTP basic authentication supported (RFC2617)
• "digest"-HTTP digest authentication supported (RFC2617)
• "krb5"-Kerberos authentication supported, using the Kerberos Domain specified in the CDMI domain (RFC 4559)
• "x509"-certificate-based authentication via TLS (RFC5246)

Extending the standard security types

The following values are examples of other widely used authentication methods that may be supported by a CDMI server:

"s3"-S3 API signed header authentication supported

"openstack"-OpenStack Identity API header authentication supported

Interoperability with these authentication methods are not defined by this international standard.

Servers may include other authentication methods not included in the above list. In these cases, it is up to the CDMI client and CDMI server (implementations themselves) to ensure interoperability.

When present, the cdmi_authentication_methods data system metadata shall be supported for all domains.

CDMI 1.1 – What else is new?

Bandwidth consumption meta-data – Split metrics between public and private interfaces for separate accounting

Addition of CDMI Group meta-data to associate objects with groups as well as users

Commercially tested and implemented extensions and feature–sets including Multi–part MIME, Domain Auth Methods, Expiring ACE and Versioning extensions.

100% backwards compatibility with CDMI v. 1.0.2 to ensure ease of use for investment protection and forward migration

13

openstack
CLOUD SOFTWARE

What resources are available

CDMI for S3 Developers

- **http://snia.org/sites/default/files/S3-like_CDMI_v1.0.pdf**

Comparison of S3/Swift functions

- **https://wiki.openstack.org/wiki/Swift/APIFeatureComparison**
  - Somewhat dated – needs updating

Implementation of CDMI filter driver for Swift

- **https://github.com/osaddon/cdmi**
  - Needs further development

Implementation of S3 filter driver for Swift

- **https://github.com/stackforge/swift3**
- Good community maintenance

What experience have you had?

Are your customers asking for multiprotocol support?

What limitations have you run across?

How do you test?

Are you:

- Modifying Swift?
- Using an other (private) implementation?

Join the Conversation

Extended cloud storage API discussions amongst implementers:

SNIA Cloud Google Group Forum

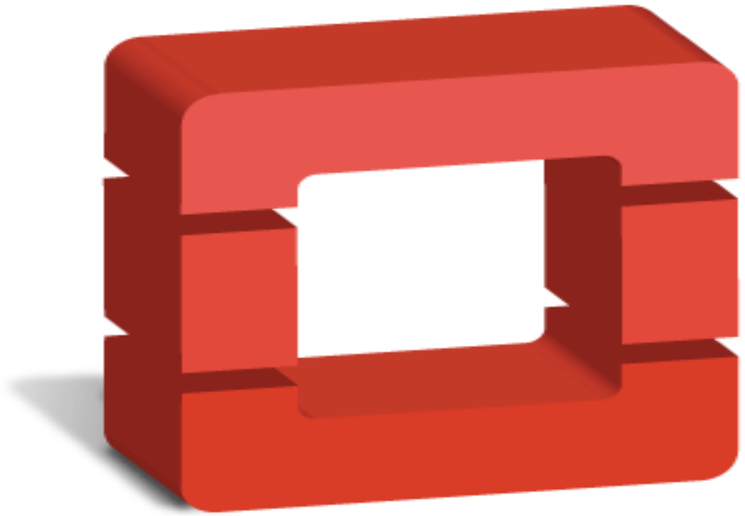
http://groups.google.com/group/snia-cloud

Follow SNIA cloud storage activities on Twitter

http://twitter.com/SNIAcloud

openstack
CLOUD SOFTWARE

Thank You!



openstack®

CLOUD SOFTWARE