SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2011

CDMI™ for Cloud IPC

David Slik
NetApp, Inc.

Session Agenda

- A Brief Overview of CDMI
- What is IPC
- CDMI Queues
- Common IPC Design Patterns
- Using IPC with CDMI
- Examples

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

A Brief Overview of CDMI

SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2011

Cloud Data Management Interface

Cloud Storage TWG

The CDMI standard has been developed over the last two years by leading storage vendors, users and researchers of cloud technology

This session assumes a basic understanding of CDMI concepts and terminology

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

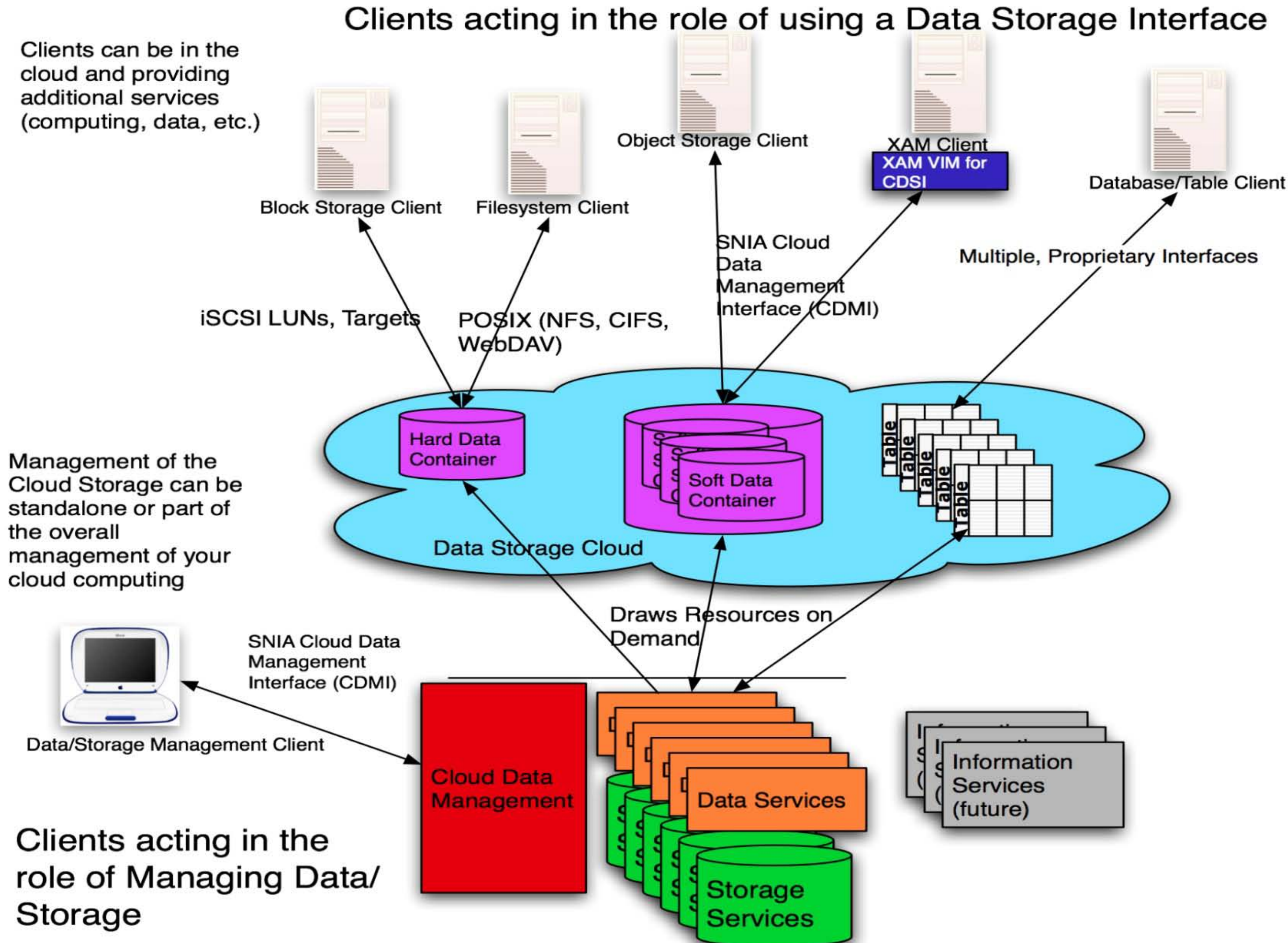
A Brief Overview of CDMI

- CDMI has the following goals:
  - To provide a standard interface for clients to communicate with storage clouds
  - To provide a standard approach for adding vendor-specific functionality without breaking client compatibility
  - **To enable standardized Cloud-to-Cloud use cases (IPC)**

For more details on use cases, see:
http://www.snia.org/tech_activities/publicreview/CloudStorageUseCasesv0.5.pdf

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

A Brief Overview of CDMI



2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

A Brief Overview of CDMI

- CDMI provides:
  - A standardized API for client interactions built on top of JSON and RESTful HTTP
  - A standardized object and metadata model for data storage and management
  - A standardized query and notifications model
  - A standardized foundation for multi-tenancy, ownership and federation

For more details on the CDMI standard, see:
http://www.snia.org/cloud/ and http://cdmi.sniacloud.com/

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

What is IPC?

- IPC, or Inter-Process Communication, is the method by which different processes or programs exchange messages in order to work together.
- This design pattern is commonly found in distributed software, and in cloud-based software
- Examples of IPC mechanisms include RPC and TCP/IP

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

What is IPC?

- By decoupling programs into smaller programs that communicate via messages, many advantages are realized, including:
  - Increased performance due to parallelization
  - Increased resiliency due to loose coupling
  - Reduced complexity due to building from simpler components
  - Ability to create geographically distributed systems

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

CDMI Queues

- CDMI Queues are first-in, first-out persistent data storage structures
- A CDMI client can:
  - Store one or more "values" into the queue
  - Read one or more of the oldest values
  - Delete one or more of the oldest values
  - Transfer the oldest value to a data object
  - Transfer one or more of the oldest values to another queue

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

CDMI Queues

- CDMI Queues also have all of the same properties that other CDMI objects have:
  - Metadata
  - CDMI Domains, ACLs, etc…
  - Move/Copy/Rename/Serialize, etc…
  - Notifications, Query, etc…
  - Access by Name or by ID
  - Store Mime type and transfer encoding along with enqueued values

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

Enqueuing into a CDMI Queue


```
HTTP connection established to http://cloud.example.com/ port 80

>> POST /MyContainer/MyQueue HTTP/1.1
>> Host: cloud.example.com
>> Content-Type: application/cdmi-queue
>> X-CDMI-Specification-Version: 1.0.1
>>
>> { "value" : [ "Value to Enqueue" ] }

<< HTTP/1.1 204 No Content

HTTP Connection closed
```

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

Dequeuing from a CDMI Queue


```
HTTP connection established to http://cloud.example.com/ port 80

>> GET /MyContainer/MyQueue?value
>> HTTP/1.1 Host: cloud.example.com
>> Accept: application/cdmi-queue
>> X-CDMI-Specification-Version: 1.0.1
>>

<< HTTP/1.1 200 OK
<< Content-Type: application/cdmi-queue
<< X-CDMI-Specification-Version: 1.0.1
<<
<< { "value" : [ "Value to Enqueue" ] }

HTTP Connection closed
```

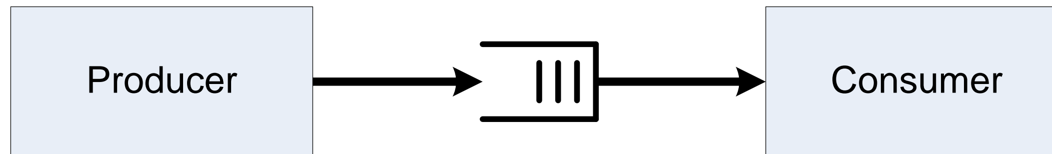
2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

Common IPC Patterns

- One to One
  - Sequential Ordering
  - Sequential Chaining
  - Priority Ordering
- One to Many – De-multiplexed & Multiplexed
- Many to One – De-multiplexed & Multiplexed
- Many to Many

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

One to One – Sequential Ordering

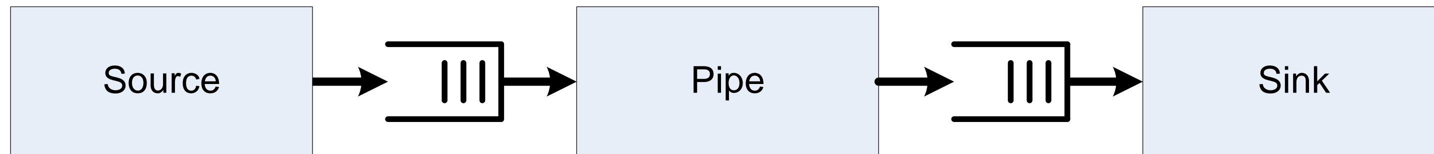


Two programs that are loosely coupled:

- For distributed operation where the producer may be on-site, consumer in the cloud, producer in the cloud and consumer on-site, or both in the same or different clouds
- For decoupled resiliency where the producer can come or go, consumer can come or go, producer's output is buffered

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

One to One – Sequential Chaining

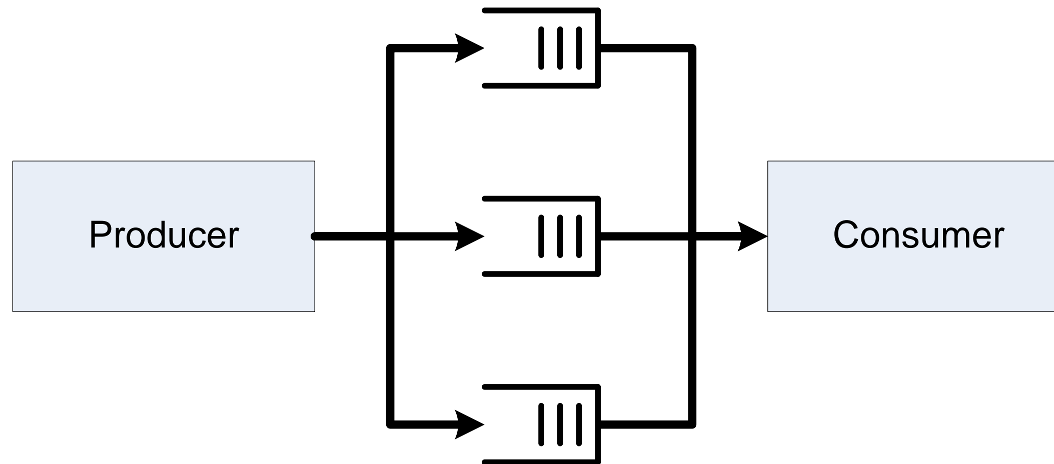


Multiple processes can be chained together

- Location of each program can be easily changed – Allows easy process migration
- Programs can be written as generic components that process the input of a queue and generate outputs to a queue

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

One to One – Priority Ordering

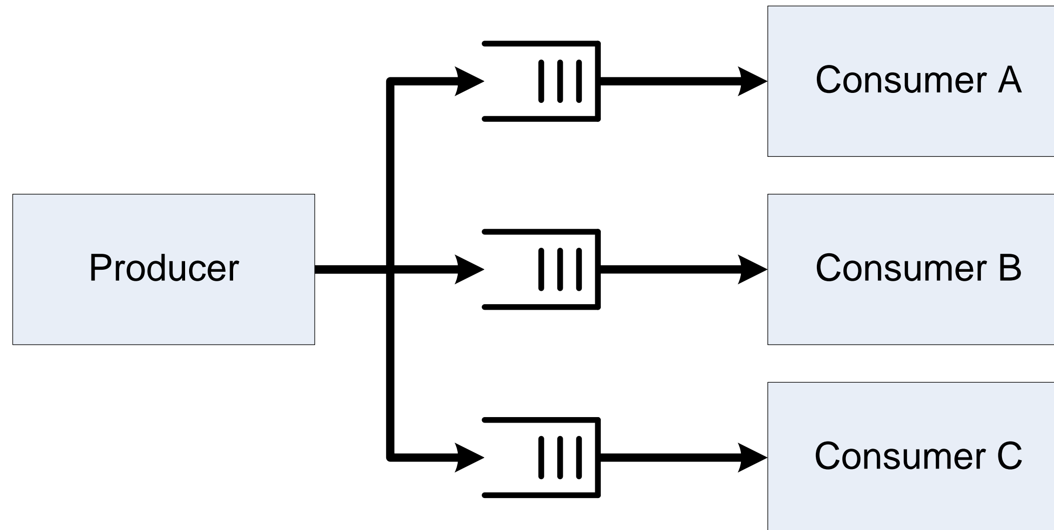


Prioritized Ordering

- Where a producer prioritizes or otherwise separates data send to the consumer, so that the consumer can act on the sent data in a different order than generated by the producer

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

One to Many – De-multiplexed



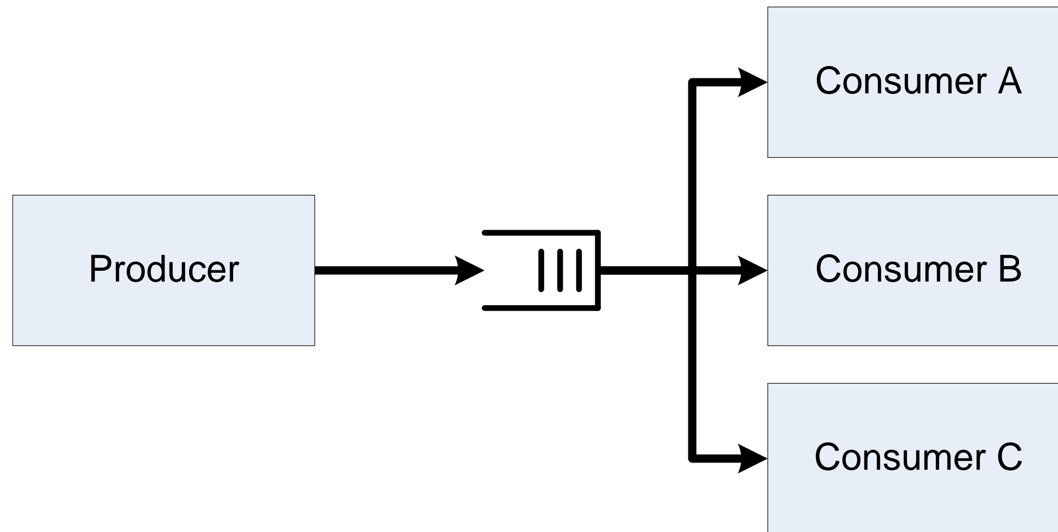
Distribution and De-multiplexing

- Producer sorts objects by type or destination to fan out messages to specialized consumers, or when messages are sent to multiple identical consumers to distribute the workload.
- Single queue per consumer
- Producer controls consumer distribution/load balancing

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

One to Many – Multiplexed

SDC
STORAGE DEVELOPER CONFERENCE
SNIA ■ SANTA CLARA, 2011

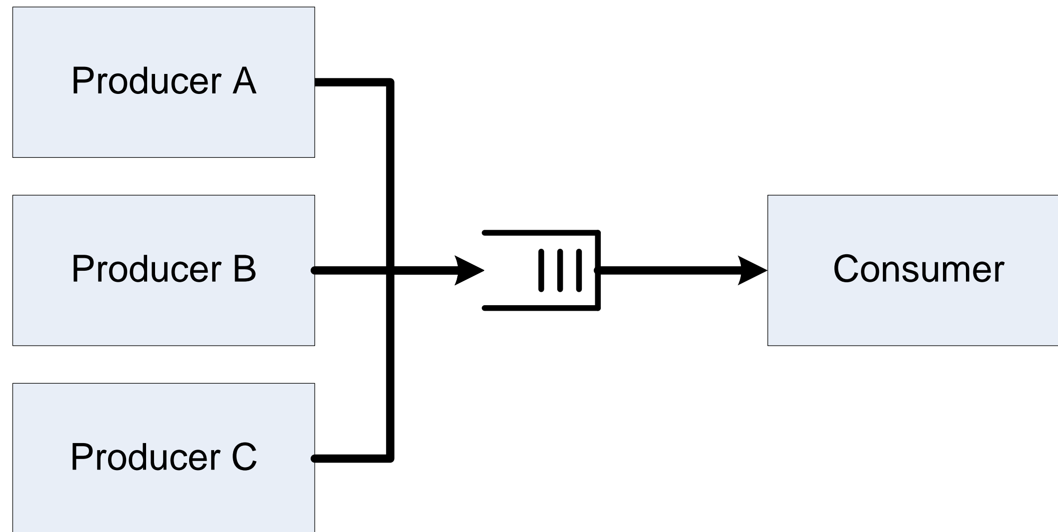


Distribution and De-multiplexing

- Since CDMI does not have locking, the consumers must atomically transfer enqueued items to a data object or local queue
- Single queue per consumer
- Consumer controls load balancing

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

Many to One – Multiplexed

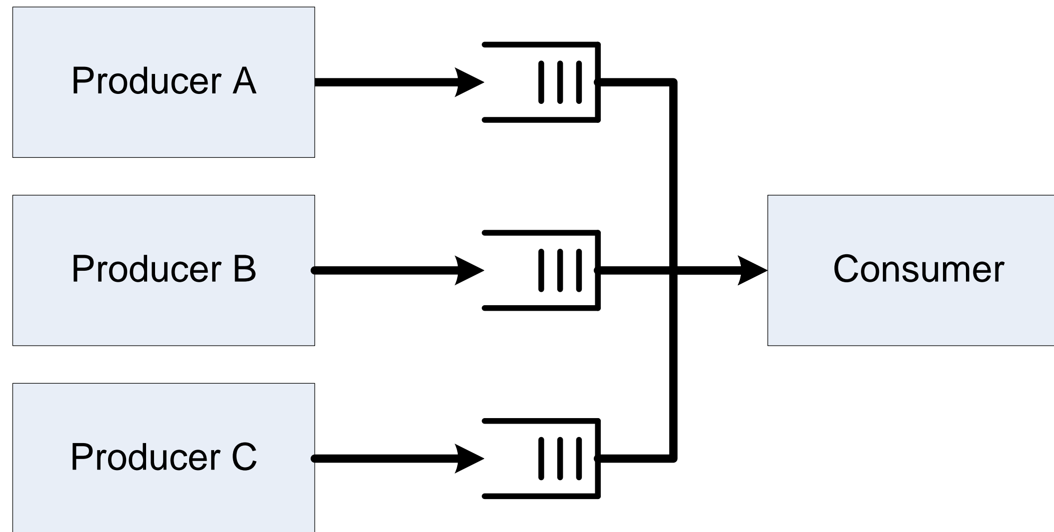


Gathering of output from multiple producers:

- Producer is computationally intensive or geographically distributed, with results gathered by a single consumer
- Single queue per consumer
- Typical pattern where messages are stored on the consumer's cloud

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

Many to One – De-multiplexed

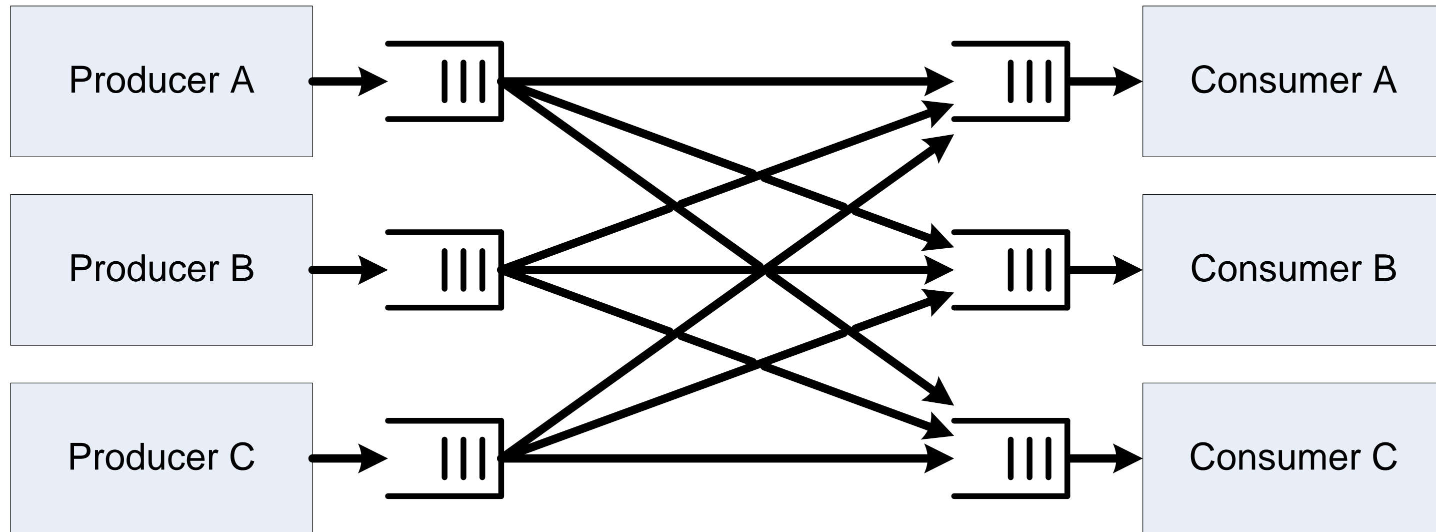


Gathering of output from multiple producers:

- Consumer can prioritize from producers
- Single queue per producer
- Typical pattern for storage of messages on the clouds where each producer resides

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

Many to Many



Universal IPC Model

- Each Producer has a queue
- Each Consumer has a queue
- Producers can xfer from their queue to any consumer queue
- Consumers can xfer from any producer queue to their queue

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

Initiating IPC via CDMI

- Using ID for Bootstrapping IPC
  - When a consumer or producer needs to locate queues, the queue can be specified by Object ID.
  - This is best used when programs are already communicating, (eg, via TCP/IP), and the Object ID can be passed electronically.

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

Initiating IPC via CDMI

- Using Paths for Bootstrapping IPC
  - When a consumer or producer needs to locate queues, the queue can be specified by path.
  - If the programs use paths, this provides an easy way to create and locate queues.
  - Paths are also more "human-friendly" than IDs.

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

Initiating IPC via CDMI

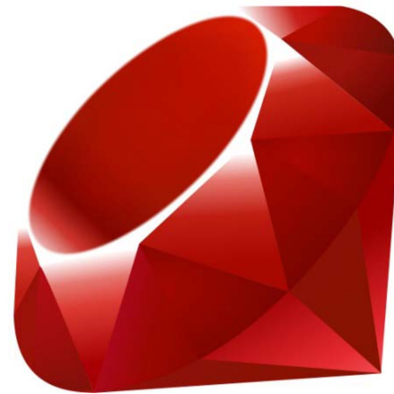
- Using Containers for Bootstrapping IPC
  - When a consumer or producer needs to locate multiple queues, the queues can be grouped by storing them in a container.

- Using Metadata for Bootstrapping IPC
  - When a consumer or producer needs to locate queues, it can query for the queues.

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

Client Demonstrations

SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2011



Ruby IPC Demonstration

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

Client Demonstrations

JavaScript/AJAX IPC Demonstration

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.

Thank you!

Questions and Answers

Contact Info:
dslik@netapp.com

2011 Storage Developer Conference. © NetApp, Inc. All Rights Reserved.