

# Developing with CDMI

**David Slik**  
**NetApp, Inc.**

- ❑ A Brief overview of CDMI
- ❑ Introduction to CDMI Client Patterns
- ❑ CDMI Client Patterns
  - ❑ Store Client Pattern
  - ❑ Browse Client Pattern
  - ❑ Retrieve Client Pattern
  - ❑ Queue Client Pattern
  - ❑ Query Client Pattern
- ❑ Client Demonstrations
  - ❑ Ruby Client Demonstration
  - ❑ JavaScript/AJAX Client Demonstration
  - ❑ iOS Client Demonstration

## Cloud Data Management Interface



Cloud Storage TWG

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

This session assumes a basic understanding of CDMI concepts and terminology

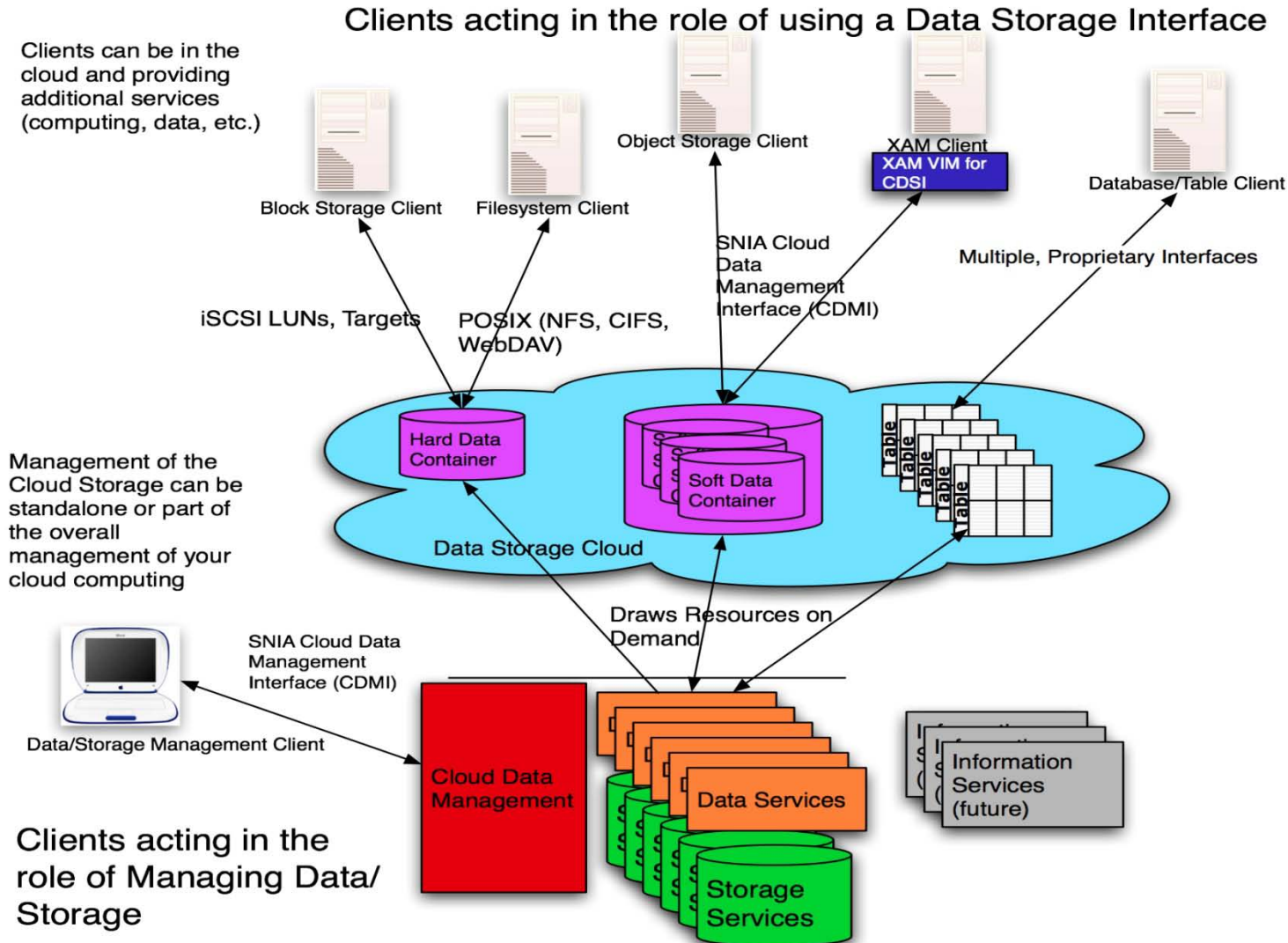
# A Brief Overview of CDMI

- 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 Federation use cases

For more details on use cases, see:

[http://www.snia.org/tech\\_activities/publicreview/CloudStorageUseCasesv0.5.pdf](http://www.snia.org/tech_activities/publicreview/CloudStorageUseCasesv0.5.pdf)

# A Brief Overview of CDMI



# 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/>

- ❑ In order to demonstrate the ways that a CDMI client will interact with a CDMI cloud, several common patterns have been identified.
- ❑ These patterns are:
  - ❑ Store Client Pattern
  - ❑ Browse Client Pattern
  - ❑ Retrieve Client Pattern
  - ❑ Queue Client Pattern
  - ❑ Query Client Pattern

# Store Client Pattern

## ❑ Objective

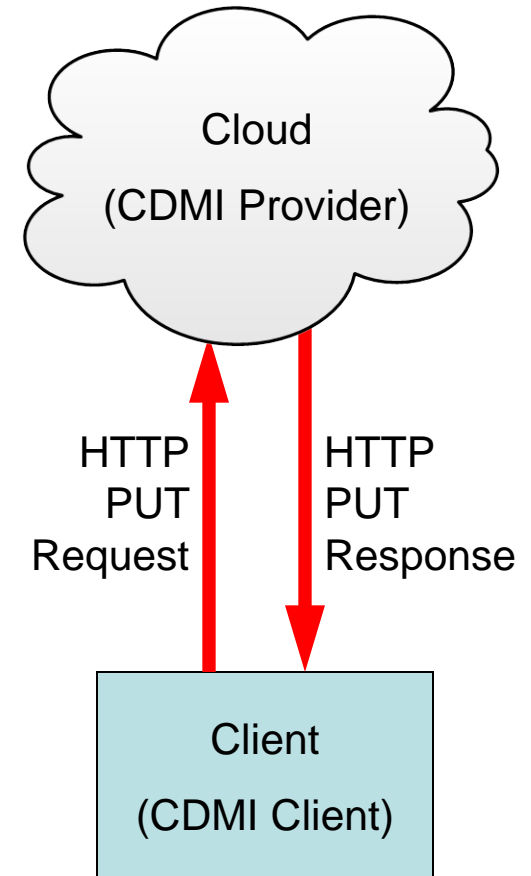
- ❑ I as a client, wish to store data on a cloud using CDMI

## ❑ Actors

- ❑ CDMI Storage Client
- ❑ CDMI Storage Provider (Cloud)

## ❑ Concepts

- ❑ Data to be stored
- ❑ Metadata to be associated with the data
- ❑ Organization of where the data should be placed in the cloud

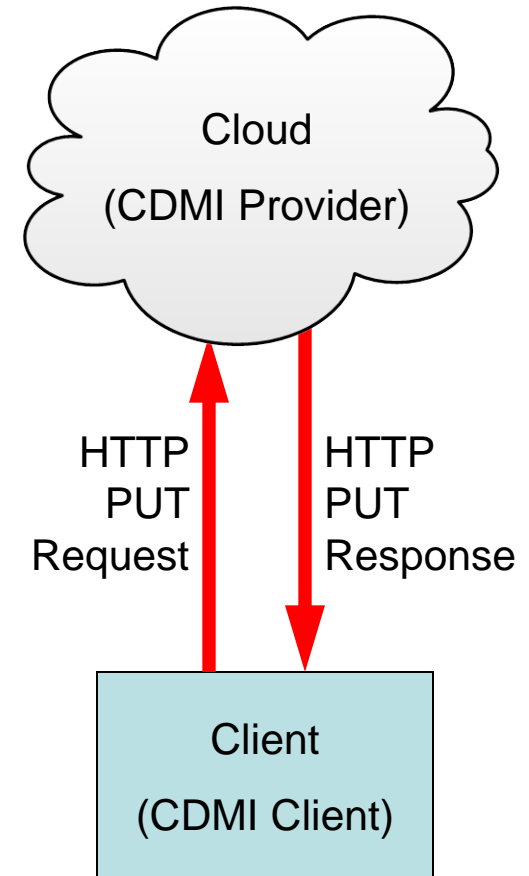




# Store Client Pattern

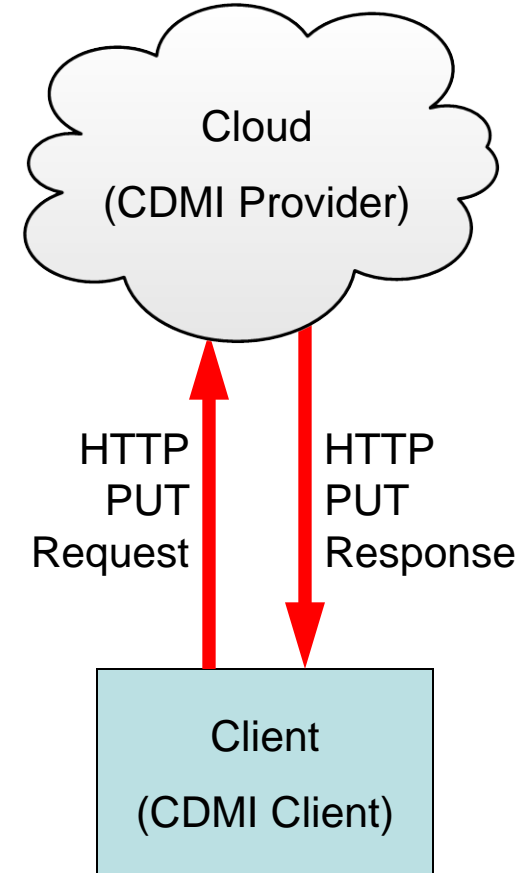
## ❑ Prerequisites

- ❑ Cloud must be capable of storing content
- ❑ Client must have a root CDMI URI
- ❑ Client must have at least one IP address for the cloud
- ❑ Client must have authentication credentials required to communicate with the cloud
- ❑ Client must have permission to store content to the cloud



## ❑ Client Actions

1. Client establishes TCP/IP session to the cloud
2. Client performs CDMI PUT to store object
  - ❑ Client optionally includes path to determine placement in a cloud namespace
  - ❑ Client optionally includes user-specified metadata
  - ❑ Client optionally includes data system metadata. This metadata allows the client to request certain storage behaviour from the cloud.
3. Client receives a response to the PUT
  - ❑ Client receives namespace identifier (URI)
  - ❑ Client receives object identifier



# Store Client Pattern

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

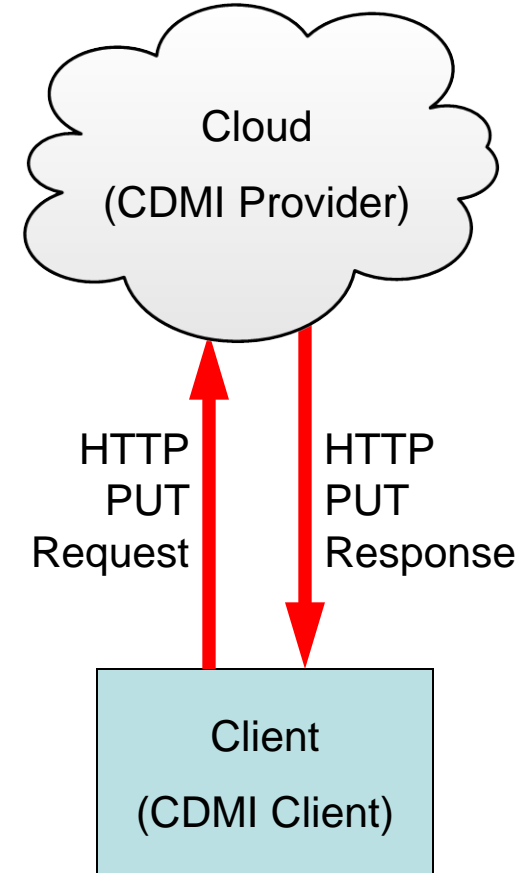
>> PUT /cdmi_root/myfile.txt HTTP/1.1
>> host: cloud.example.com
>> X-CDMI-Specification-Version: 1.0
>> Content-Type: application/vnd.org.snia.cdmi.dataobject+json
>> Content-Length: 99
>>
>> { "mimetype" : "text/plain", "metadata" : { }, "value" : "This is the
Value of this Data Object" }

<< HTTP/1.1 201 Created
<< X-CDMI-Specification-Version: 1.0
<< Content-Type: application/vnd.org.snia.cdmi.dataobject+json
<< Content-Length: 328
<<
<< { "objectURI" : "/cdmi_root/myfile.txt", "objectID" :
"0000706D0010B84FAD185C425D8B537E", "parentURI" : "/cdmi_root/", "domainURI"
: "/cdmi_root/cdmi_domains/test/", "capabilitiesURI" :
"/cdmi_root/cdmi_capabilities/DataObject", "completionStatus" : "Complete",
"mimetype" : "text/plain", "metadata" : { "cdmi_size" : "38" } }

HTTP Connection closed
```

# Store Client Pattern

- ❑ Variations on the pattern
  - ❑ Store as non-CDMI HTTP
  - ❑ Store condition on non-existence
  - ❑ Store by ID
  - ❑ Store to existing object (update)
  - ❑ Store specific fields to an existing object
  - ❑ Store to a range of an existing object
  - ❑ Store in multiple parts
  - ❑ Store with source from another object
  - ❑ Store with source from another cloud
  - ❑ Store with a different owning domain



# Browse Client Pattern

## ❑ Objective

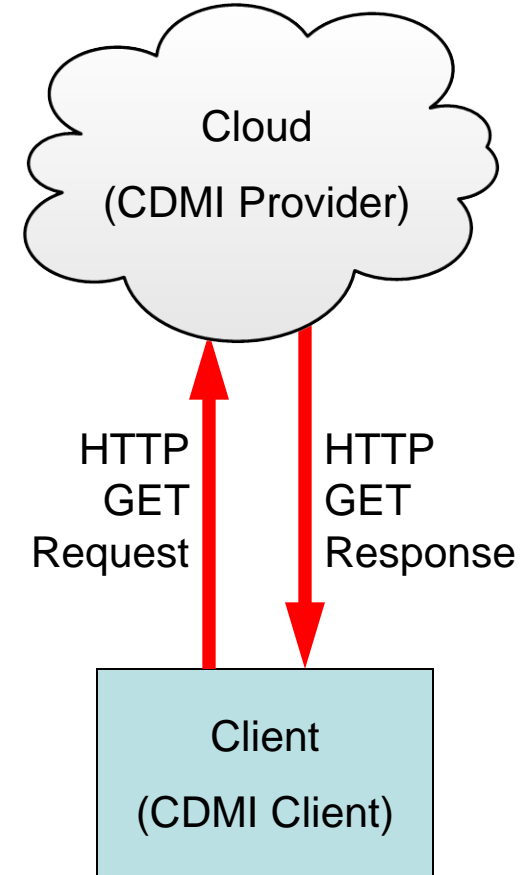
- ❑ I as a client, wish to list content in a specific location within a cloud namespace using CDMI

## ❑ Actors

- ❑ CDMI Storage Client
- ❑ CDMI Storage Provider (Cloud)

## ❑ Concepts

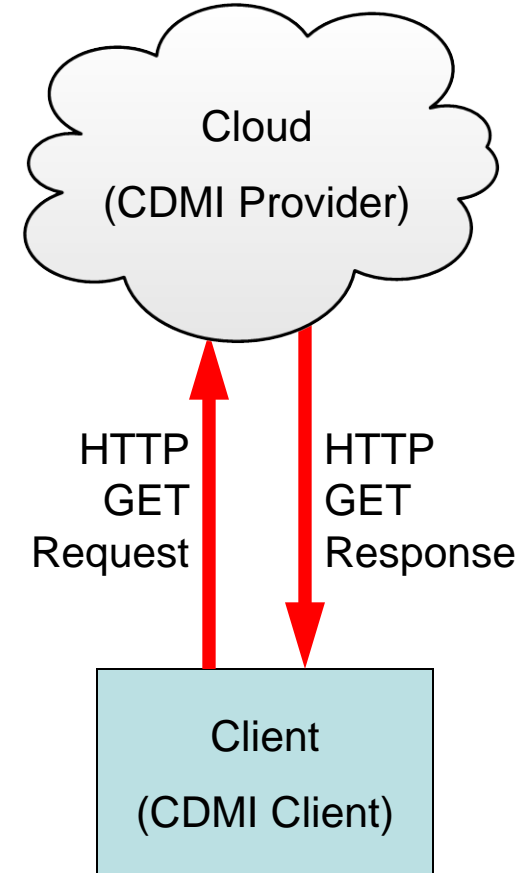
- ❑ Namespace to be listed
- ❑ Types of objects within a namespace



# Browse Client Pattern

## ❑ Prerequisites

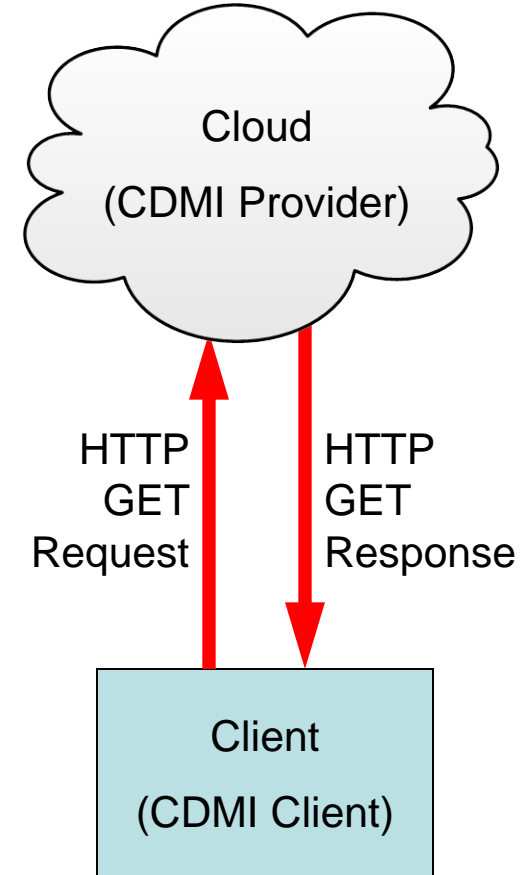
- ❑ **Cloud must be capable of browsing content**
- ❑ Client must have a root CDMI URI
- ❑ **Client must have a URI to a namespace**
- ❑ Client must have at least one IP address for the cloud
- ❑ Client must have authentication credentials required to communicate with the cloud
- ❑ **Client must have permission to browse listings of objects stored in the cloud**



# Browse Client Pattern

## ❑ Client Actions

1. Client establishes TCP/IP session to the cloud
2. Client performs CDMI GET to list container
  - ❑ Client includes path to container to list
3. Client receives a response to the GET
  - ❑ Client receives metadata for the container
  - ❑ Client receives listing of objects contained within the container
  - ❑ Client receives details of exports, snapshots and other container-related attributes



# Browse Client Pattern

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

>> GET /cdmi_root/ HTTP/1.1
>> host: cloud.example.com
>> X-CDMI-Specification-Version: 1.0
>>

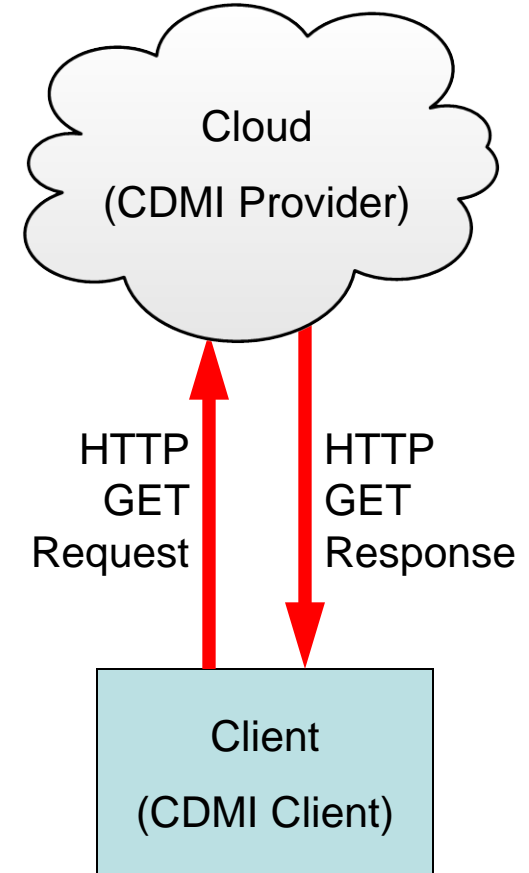
<< HTTP/1.1 200 OK
<< X-CDMI-Specification-Version: 1.0
<< Content-Type: application/vnd.org.snia.cdmi.container+json
<< Content-Length: 324
<<
<< { "objectURI" : "/cdmi_root/", "objectID" :
"0000706D0010B84FAD185C425D8B537E", "parentURI" : "/", "domainURI" :
"/cdmi_root/cdmi_domains/test", "capabilitiesURI" :
"/cdmi_root/cdmi_capabilities/Container", "completionStatus" : "Complete",
"metadata" : { }, "childrenrange" : "0-1", "children" : [ "myfile.txt",
"tests/" ] }

HTTP Connection closed
```



# Browse Client Pattern

- ❑ Variations on the pattern
  - ❑ Browse a container by ID
  - ❑ Browse specific fields in a container
  - ❑ Browse a range of children in a container



# Retrieve Client Pattern

## ❑ Objective

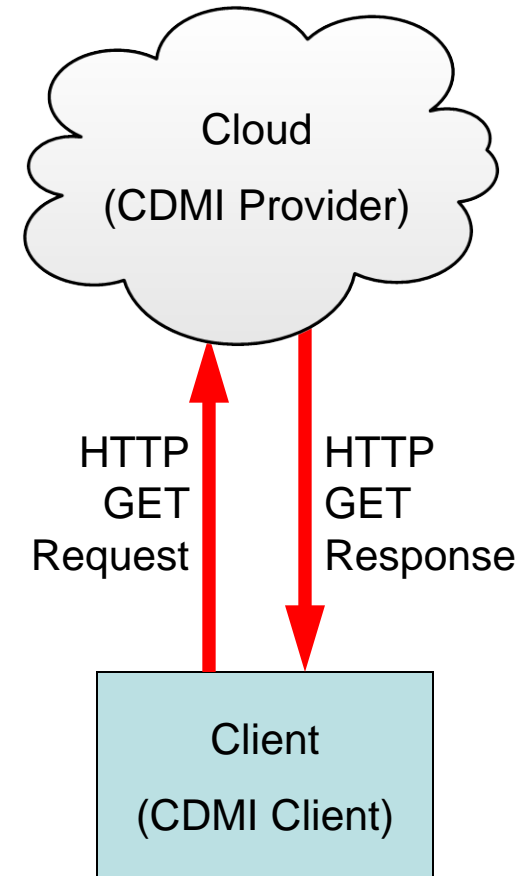
- ❑ I as a client, wish to retrieve data stored in a cloud using CDMI

## ❑ Actors

- ❑ CDMI Storage Client
- ❑ CDMI Storage Provider (Cloud)

## ❑ Concepts

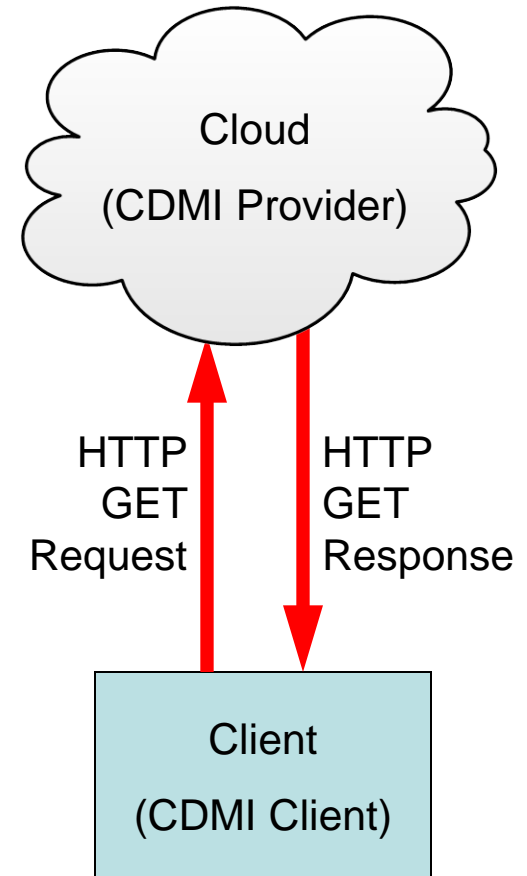
- ❑ ID or Path of object to retrieve data from
- ❑ Metadata to associated with the data
- ❑ Organization of where the data is placed in the cloud



# Retrieve Client Pattern

## ❑ Prerequisites

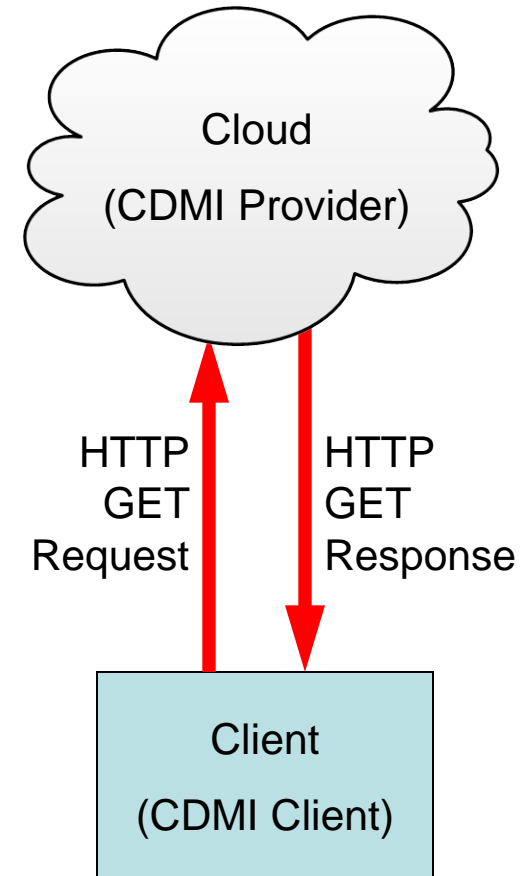
- ❑ **Cloud must be capable of retrieving content**
- ❑ Client must have a root CDMI URI
- ❑ **Client must have an Object ID or a namespace path (URI) to an Object**
- ❑ Client must have at least one IP address for the cloud
- ❑ Client must have authentication credentials required to communicate with the cloud
- ❑ **Client must have permission to retrieve content from the cloud**



# Retrieve Client Pattern

## □ Client Actions

1. Client establishes TCP/IP session to the cloud
2. Client performs CDMI GET to retrieve the contents of a stored data object
  - Client uses object ID or path to object
3. Client receives a response to the GET
  - Client receives metadata for the object
  - Client receives data stored in the object
  - Client receives details of versions and other data object-related attributes



# Retrieve Client Pattern

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

>> GET /cdmi_root/myfile.txt HTTP/1.1
>> host: cloud.example.com
>> X-CDMI-Specification-Version: 1.0
>>

<< HTTP/1.1 200 OK
<< X-CDMI-Specification-Version: 1.0
<< Content-Type: application/vnd.org.snia.cdmi.dataobject+json
<< Content-Length: 377
<<
<< { "objectURI" : "/cdmi_root/myfile.txt", "objectID" :
"0000706D0010B84FAD185C425D8B537E", "parentURI" : "/cdmi_root/", "domainURI"
: "/cdmi_root/cdmi_domains/test/", "capabilitiesURI" :
"/cdmi_root/cdmi_capabilities/DataObject", "completionStatus" : "Complete",
"mimetype" : "text/plain", "metadata" : { "cdmi_size" : "38" }, "value" :
"This is the Value of this Data Object" }

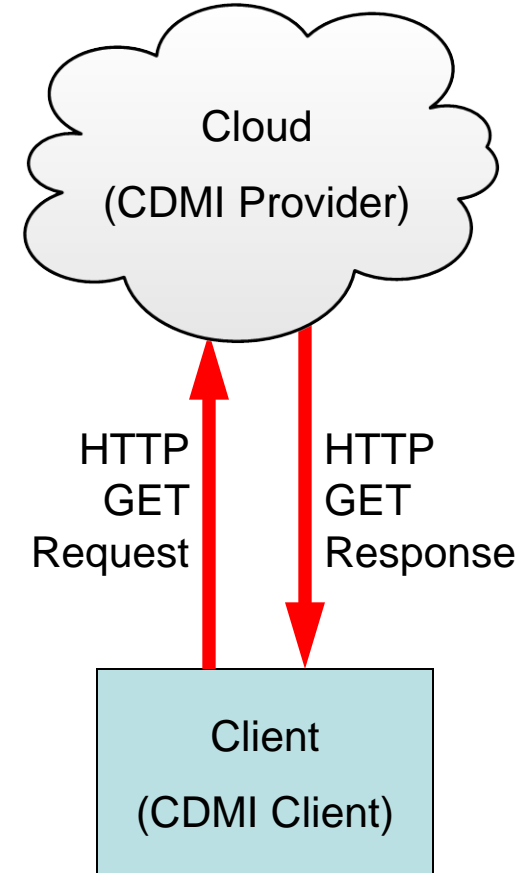
HTTP Connection closed
```

# Retrieve Client Pattern

```
HTTP connection established to http://cloud.example.com/ port 80
>> GET /cdmi_root/myfile.txt HTTP/1.1
>> host: cloud.example.com
>>
<< HTTP/1.1 200 OK
<< Content-Type: text/plain
<< Content-Length: 38
<<
<< This is the Value of this Data Object
HTTP Connection closed
```

# Retrieve Client Pattern

- ❑ Variations on the pattern
  - ❑ Retrieve as non-CDMI HTTP
  - ❑ Retrieve specific fields from a data object
  - ❑ Retrieve a range of the data stored in an object



# Queue Client Pattern

## ❑ Objective

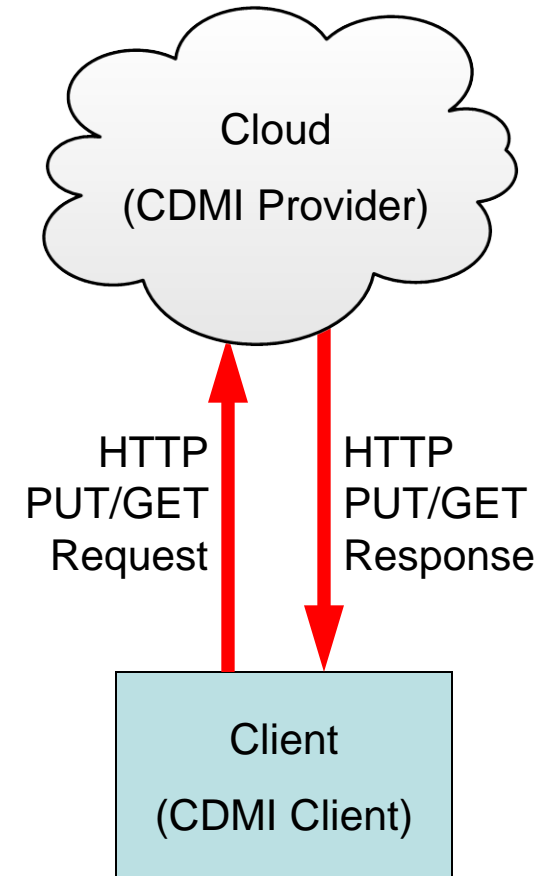
- ❑ I as a client, wish to enqueue and dequeue data in a cloud using CDMI

## ❑ Actors

- ❑ CDMI Storage Client
- ❑ CDMI Storage Provider (Cloud)

## ❑ Concepts

- ❑ ID or Path of queue
- ❑ Metadata to associated with the queue
- ❑ Organization of where the queue is placed in the cloud

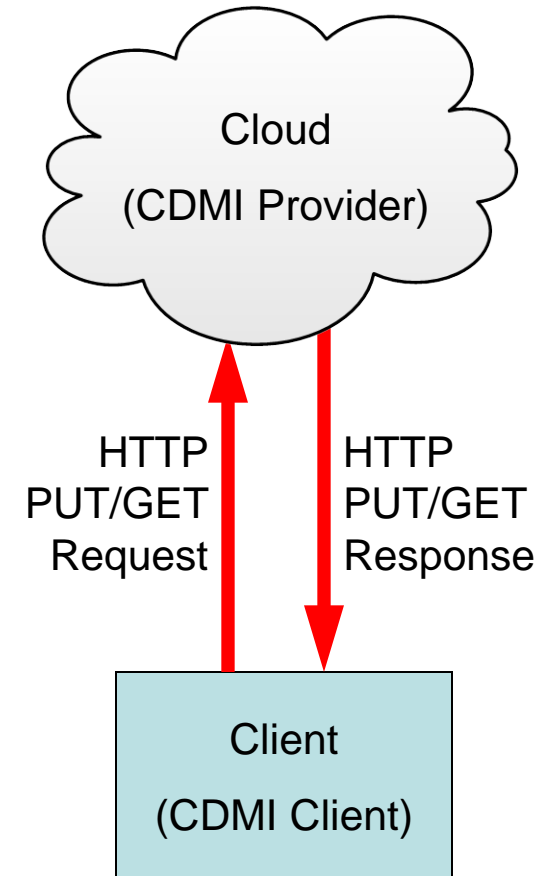




# Queue Client Pattern

## ❑ Prerequisites

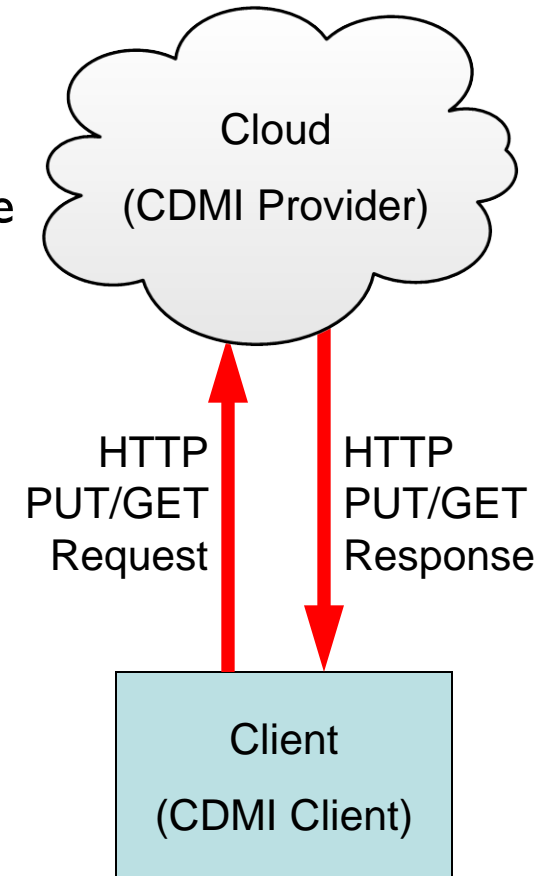
- ❑ **Cloud must be capable of queuing content**
- ❑ Client must have a root CDMI URI
- ❑ Client must have at least one IP address for the cloud
- ❑ Client must have authentication credentials required to communicate with the cloud
- ❑ **Client must have permission to queue content in the cloud**



# Queue Client Pattern

## ❑ Client Actions, Part I

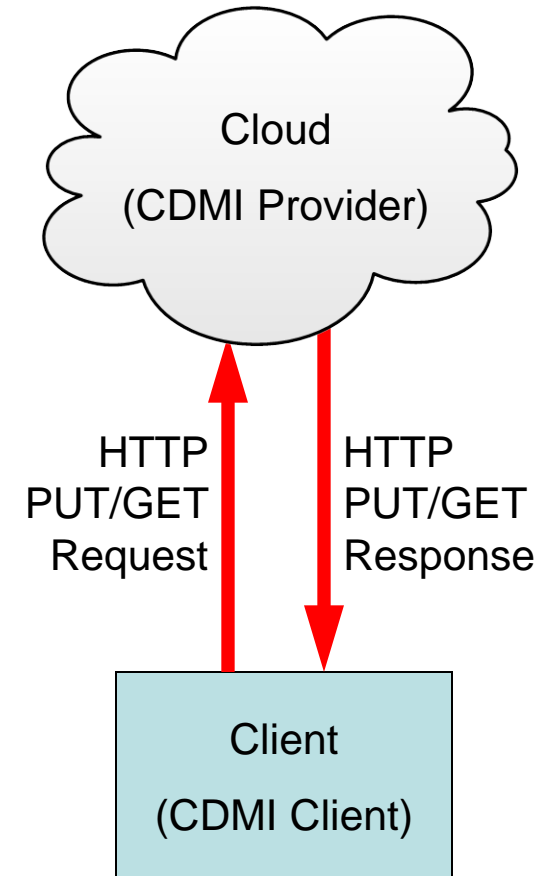
1. Client establishes TCP/IP session to the cloud
2. Client performs CDMI PUT to create new queue
  - ❑ Client optionally includes path to determine placement in a cloud namespace
  - ❑ Client optionally includes user-specified metadata
3. Client receives a response to the PUT
  - ❑ Client receives queue namespace identifier
  - ❑ Client receives queue object identifier



# Queue Client Pattern

## ❑ Client Actions, Part 2

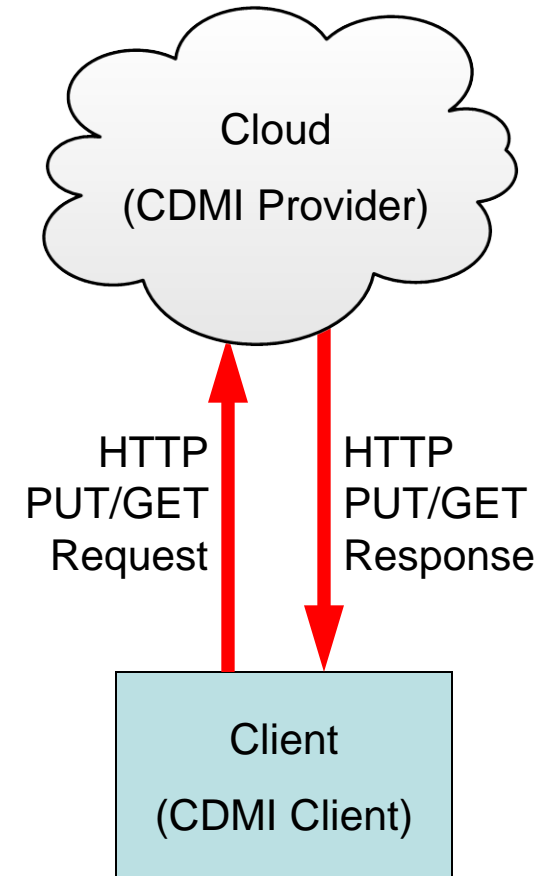
1. Client establishes TCP/IP session to the cloud
2. Client performs CDMI POST to enqueue data into an existing queue (created in Part 1)
3. Client receives a response to the PUT
  - ❑ Client receives queue details



# Queue Client Pattern

## ❑ Client Actions, Part 3

1. Client establishes TCP/IP session to the cloud
2. Client performs CDMI GET to dequeue data from an existing queue (created in Part I)
3. Client receives a response to the GET
  - ❑ Client receives enqueued data
  - ❑ Client receives queue details



# Queue Client Pattern

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

>> PUT /cdmi_root/myqueue HTTP/1.1
>> host: cloud.example.com
>> X-CDMI-Specification-Version: 1.0
>> Content-Type: application/vnd.org.snia.cdmi.queue+json
>> Content-Length: 47
>>
>> { "mimetype" : "text/plain", "metadata" : { } }

<< HTTP/1.1 201 Created
<< X-CDMI-Specification-Version: 1.0
<< Content-Type: application/vnd.org.snia.cdmi.dataobject+json
<< Content-Length: 343
<<
<< { "objectURI" : "/cdmi_root/myqueue", "objectID" :
"0000706D00101ADEBC119D1BFE98672A", "parentURI" : "/cdmi_root/", "domainURI"
: "/cdmi_root/cdmi_domains/test/", "capabilitiesURI" :
"/cdmi_root/cdmi_capabilities/QueueObject", "completionStatus" : "Complete",
"mimetype" : "text/plain", "metadata" : { "cdmi_size" : "0" }, "queueValues"
: "" }

HTTP Connection closed
```

# Queue Client Pattern

```
HTTP connection established to http://cloud.example.com/ port 80
>> POST /cdmi_root/myqueue HTTP/1.1
>> host: cloud.example.com
>> X-CDMI-Specification-Version: 1.0
>> Content-Type: application/vnd.org.snia.cdmi.dataobject+json
>> Content-Length: 56
>>
>> { "mimetype" : "text/plain", "value" : "Enqueued Text" }
<< HTTP/1.1 201 Created
HTTP Connection closed
```

# Queue Client Pattern

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

>> GET /cdmi_root/myqueue HTTP/1.1
>> host: cloud.example.com
>> X-CDMI-Specification-Version: 1.0
>>

<< HTTP/1.1 200 OK
<< X-CDMI-Specification-Version: 1.0
<< Content-Type: application/vnd.org.snia.cdmi.queue+json
<< Content-Length: 409
<<
<< { "objectURI" : "/cdmi_root/myqueue", "objectID" :
"0000706D00101ADEBC119D1BFE98672A", "parentURI" : "/cdmi_root/", "domainURI"
: "/cdmi_root/cdmi_domains/test/", "capabilitiesURI" :
"/cdmi_root/cdmi_capabilities/QueueObject", "completionStatus" : "Complete",
"metadata" : { "cdmi_size" : "38" }, "queueValues" : "0-0", "mimetype" : [
"text/plain" ], "valuerange" : [ "0-12" ], "value" : [ "Enqueued Text" ] }

HTTP Connection closed
```

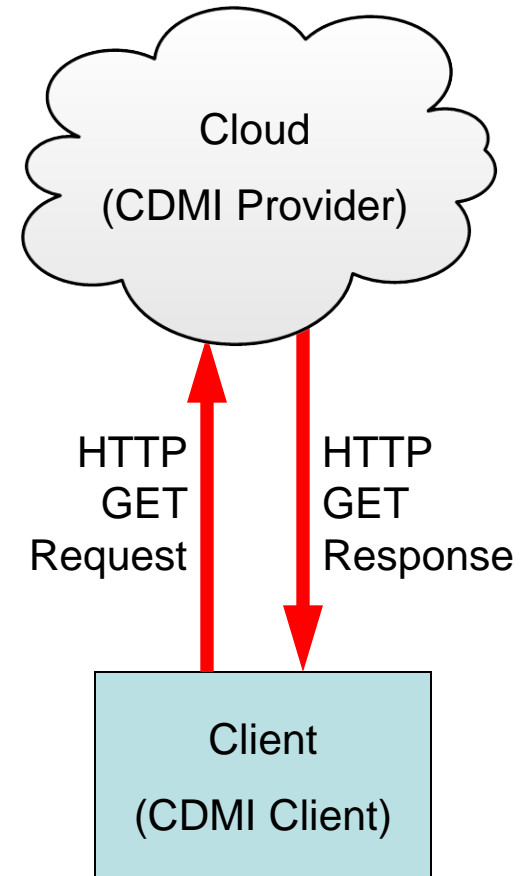
# Queue Client Pattern

```
HTTP connection established to http://cloud.example.com/ port 80
>> DELETE /cdmi_root/myqueue?value HTTP/1.1
>> host: cloud.example.com
>> X-CDMI-Specification-Version: 1.0
>>
<< HTTP/1.1 204 No Content
HTTP Connection closed
```



# Queue Client Pattern

- ❑ Variations on the pattern
  - ❑ Retrieve multiple values
  - ❑ Delete multiple values
  - ❑ Create a queue with an initial value from a data object
  - ❑ Enqueue a value from a data object
  - ❑ Enqueue a value from another queue
  - ❑ Move a value from another queue
  - ❑ Transfer results from one queue to another
  - ❑ Create application to application data flows
  - ❑ Create fan-out/fan-in producer/consumer relationships between distributed actors in a loosely coupled distributed compute system



# Query Client Pattern

## ❑ Objective

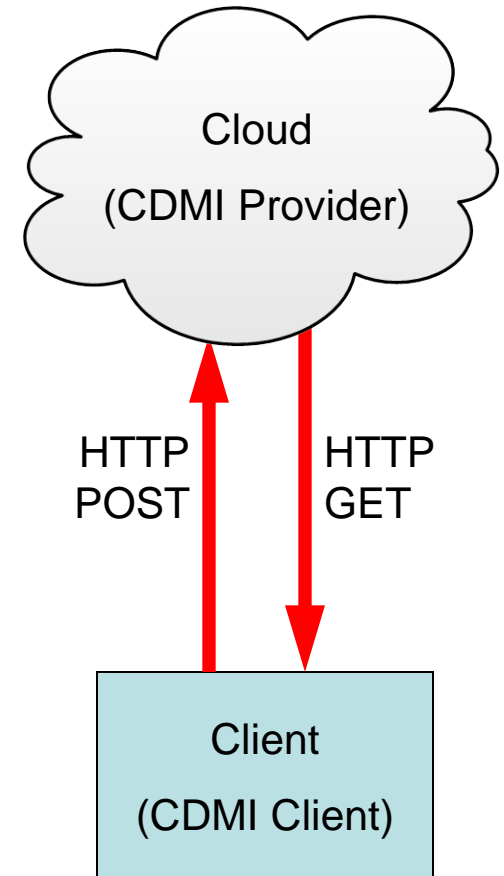
- ❑ I as a client, wish to locate data matching certain metadata in a cloud using CDMI

## ❑ Actors

- ❑ CDMI Storage Client
- ❑ CDMI Storage Provider (Cloud)

## ❑ Concepts

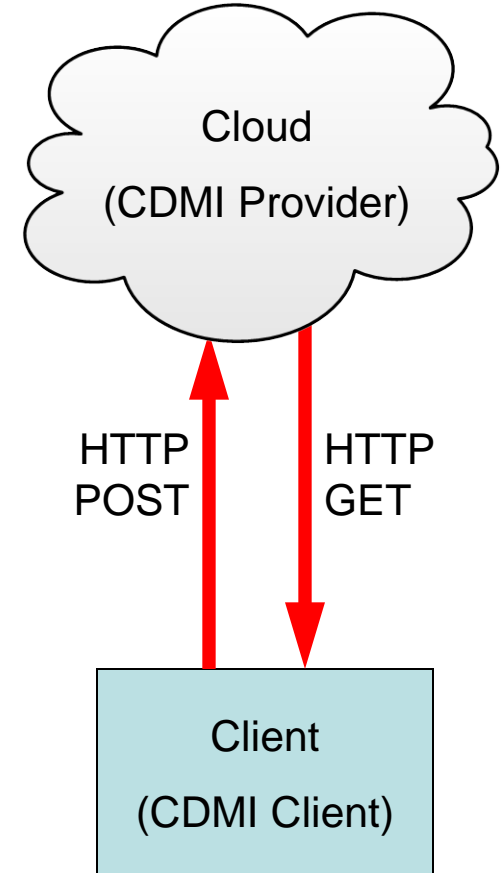
- ❑ Data to be stored
- ❑ Metadata to be associated with the data
- ❑ Query statements



# Query Client Pattern

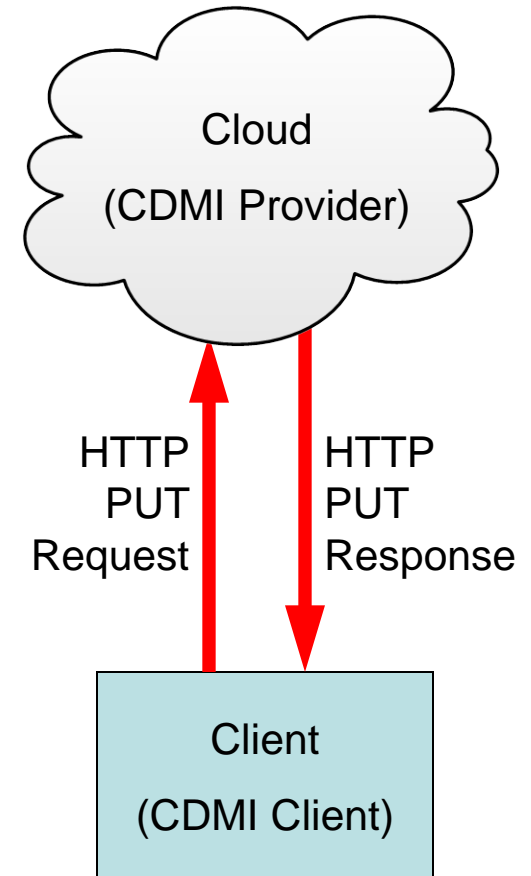
## ❑ Prerequisites

- ❑ **Cloud must be capable of performing queries**
- ❑ Client must have a root CDMI URI
- ❑ Client must have at least one IP address for the cloud
- ❑ Client must have authentication credentials required to communicate with the cloud
- ❑ **Client must have permission to query content to the cloud**



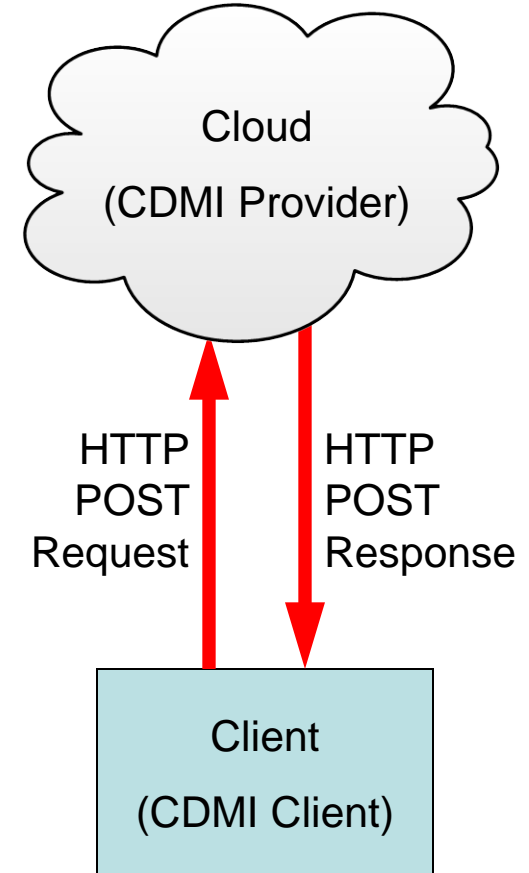
## ❑ Client Actions, Part I

1. Client establishes TCP/IP session to the cloud
2. Client creates a Query Queue
  - ❑ See Query Client Pattern for more details
  - ❑ Client creates a queue with special metadata that indicates to the cloud that the queue is to be used for query
3. Client receives a response to the queue create
  - ❑ Client receives queue namespace identifier
  - ❑ Client receives queue identifier



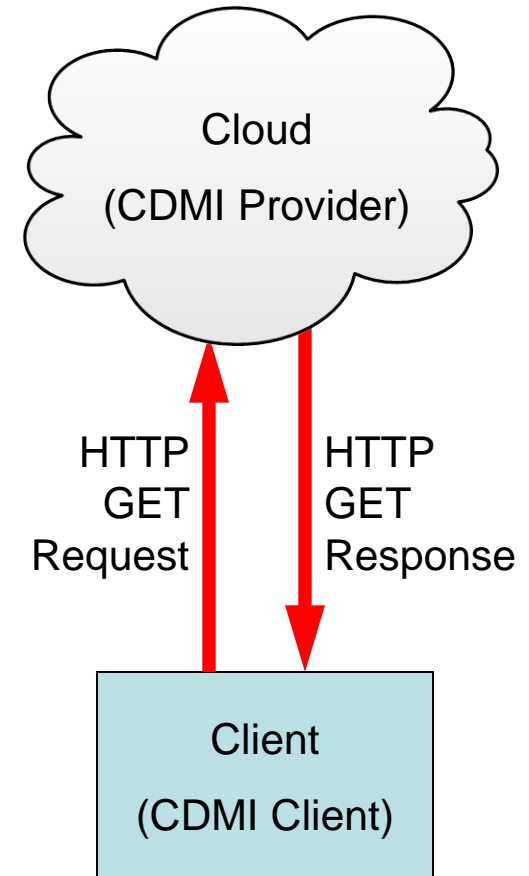
## ❑ Client Actions, Part 2

1. Client establishes TCP/IP session to the cloud
2. Client performs CDMI POST to enqueue a query request into the query queue (created in Part 1)
3. Client receives a response to the PUT
  - ❑ System performs query and places results into the location specified in the query request



## ❑ Client Actions, Part 3

1. Client establishes TCP/IP session to the cloud
2. Client performs CDMI GET to read the query results from the data object (or queue) specified in the query request (submitted in Part 2)
3. Client receives a response to the GET
  - ❑ Client receives query results



# Query Client Pattern

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

>> PUT /cdmi_root/queryqueue HTTP/1.1
>> host: cloud.example.com
>> X-CDMI-Specification-Version: 1.0
>> Content-Type: application/vnd.org.snia.cdmi.queue+json
>> Content-Length: 86
>>
>> { "mimetype" : "text/plain", "metadata" : { "cdmi_queue_type" :
"cdmi_query_queue" } }

<< HTTP/1.1 201 Created
<< X-CDMI-Specification-Version: 1.0
<< Content-Type: application/vnd.org.snia.cdmi.dataobject+json
<< Content-Length: 383
<<
<< { "objectURI" : "/cdmi_root/myqueue", "objectID" :
"0000706D00101ADEBC119D1BFE98672A", "parentURI" : "/cdmi_root/", "domainURI" :
"/cdmi_root/cdmi_domains/test/", "capabilitiesURI" :
"/cdmi_root/cdmi_capabilities/QueueObject", "completionStatus" : "Complete",
"mimetype" : "text/plain", "metadata" : { "cdmi_size" : "0", "cdmi_queue_type" :
"cdmi_query_queue" }, "queueValues" : "" }

HTTP Connection closed
```

# Query Client Pattern

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

>> POST /cdmi_root/queryqueue HTTP/1.1
>> host: cloud.example.com
>> X-CDMI-Specification-Version: 1.0
>> Content-Type: application/vnd.org.snia.cdmidataobject+json
>> Content-Length: 350
>>
>> { "mimetype" : "text/plain", "value" : "{ \"cdmi_query_name\" : \"test\",
\"cdmi_query_uri\" : \"/cdmi_root/queryresults\", \"cdmi_query_specification\" : [
{ \"domainURI\" : \"== /cdmi_domains/test\", \"metadata\" : { \"cdmi_size\" : \">
10\" } } ], \"cdmi_returned_results\" : { \"objectID\" : \"\", \"metadata\" : {
\"cdmi_size\" : \"\" } } }" }

<< HTTP/1.1 201 Created

HTTP Connection closed
```



# Query Client Pattern

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

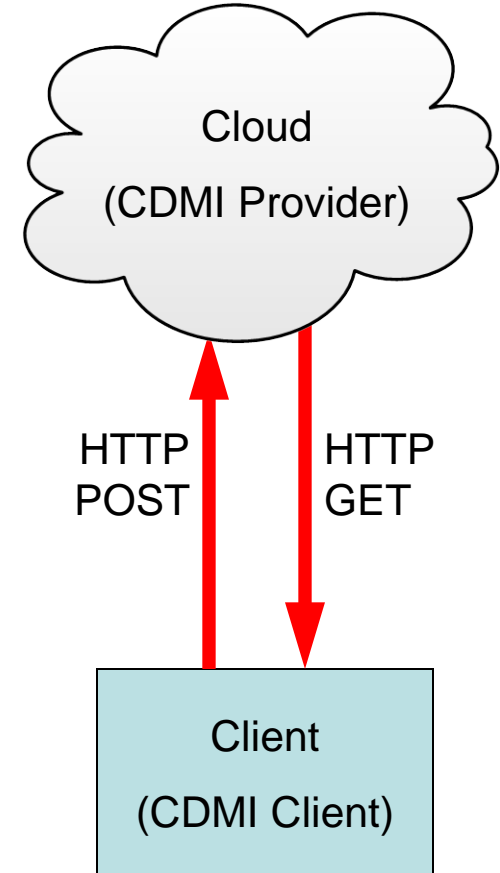
>> GET /cdmi_root/queryresults HTTP/1.1
>> host: cloud.example.com
>> X-CDMI-Specification-Version: 1.0
>>

<< HTTP/1.1 200 OK
<< X-CDMI-Specification-Version: 1.0
<< Content-Type: application/vnd.org.snia.cdmi.dataobject+json
<< Content-Length: 644
<<
<< { "objectURI" : "/cdmi_root/myfile.txt", "objectID" :
"0000706D0010B84FAD185C425D8B537E", "parentURI" : "/cdmi_root/", "domainURI" :
"/cdmi_root/cdmi_domains/test/", "capabilitiesURI" :
"/cdmi_root/cdmi_capabilities/DataObject", "completionStatus" : "Complete",
"mimetype" : "text/json", "metadata" : { "cdmi_size" : "304" }, "value" : "{
\"cdmi_query_name\" : \"test\", \"cdmi_query_count\" : \"2\",
\"cdmi_query_results\" : [ { \"objectID\" : \"0000706D0010D538DEEE8E38399E2815\",
\"metadata\" : { \"cdmi_size\" : \"2398733\" } }, { \"objectID\" :
\"0000706D0010734CE0BAEB29DD542B51\", \"metadata\" : { \"cdmi_size\" : \"2398733\"
} } ] }" }

HTTP Connection closed
```

# Query Client Pattern

- ❑ Variations on the pattern
  - ❑ Store results to a queue
  - ❑ Store results to a object by ID
  - ❑ Merge query results into a common queue
  - ❑ Create a continuously running query by creating a Notification Queue



- ❑ Demonstrations of Store, Browse and Retrieve patterns
  - ❑ Store Pattern – Ruby command line script
  - ❑ Browse Pattern – AJAX Javascript-based web page
  - ❑ Retrieve Pattern – iOS application running on iPad
  
- ❑ Illustrates how patterns can work together across multiple applications and platforms
  - ❑ As an example, store from UNIX, manage on the web, and consume on mobile device



## Ruby Client Demonstration



## JavaScript/AJAX Client Demonstration



## **iPad/iOS Client Demonstration**

# Thank you!

## Questions and Answers

Contact Info:  
[dslik@netapp.com](mailto:dslik@netapp.com)