

# **CDMI™ and Cloud Federation Year 3**

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

# Session Agenda

- ❑ A Brief Overview of CDMI
- ❑ Federation Overview
- ❑ Bidirectional Federation between two writable clouds, in depth

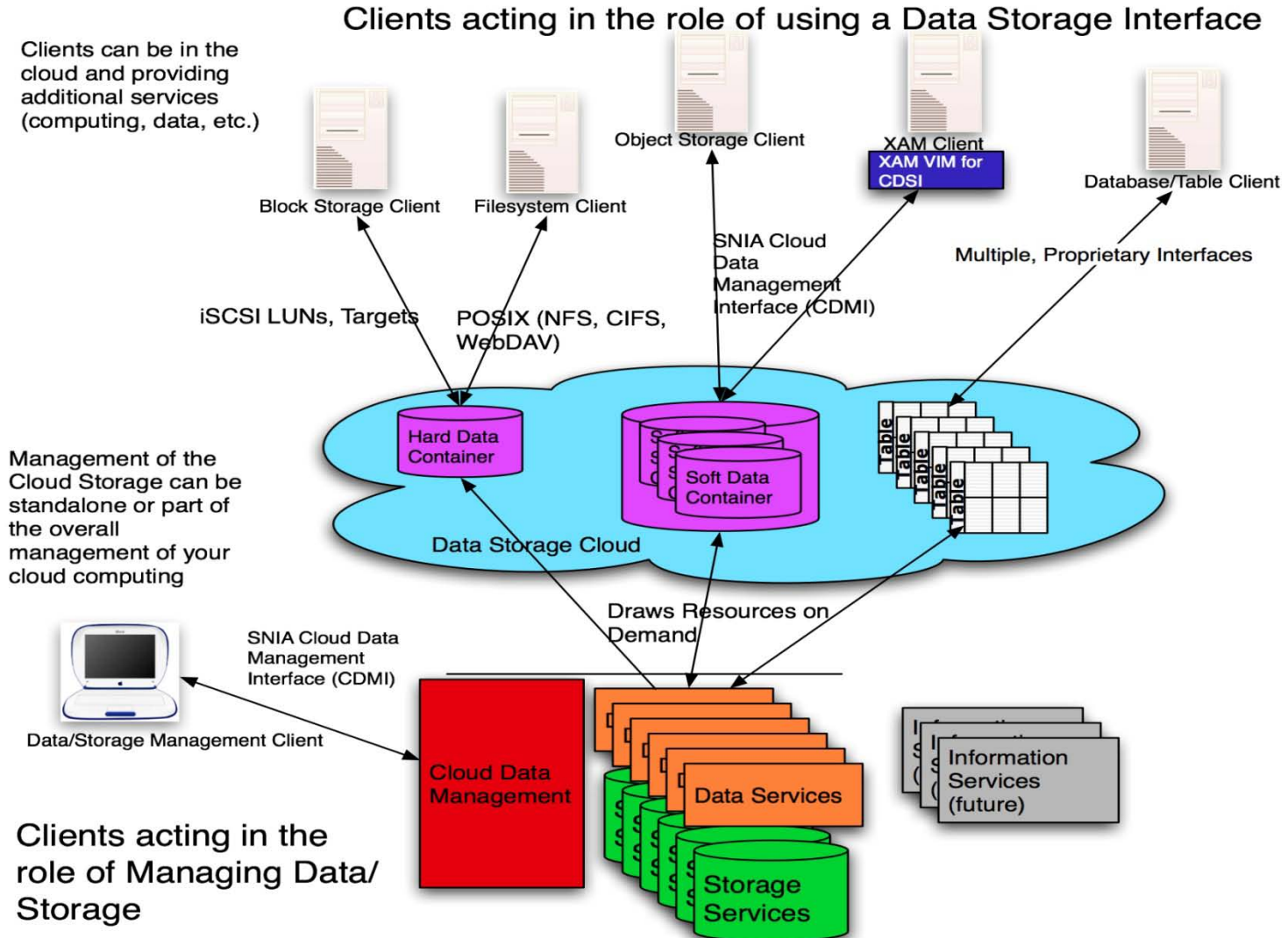
# A Brief Overview of CDMI

- 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 (Federations)**

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

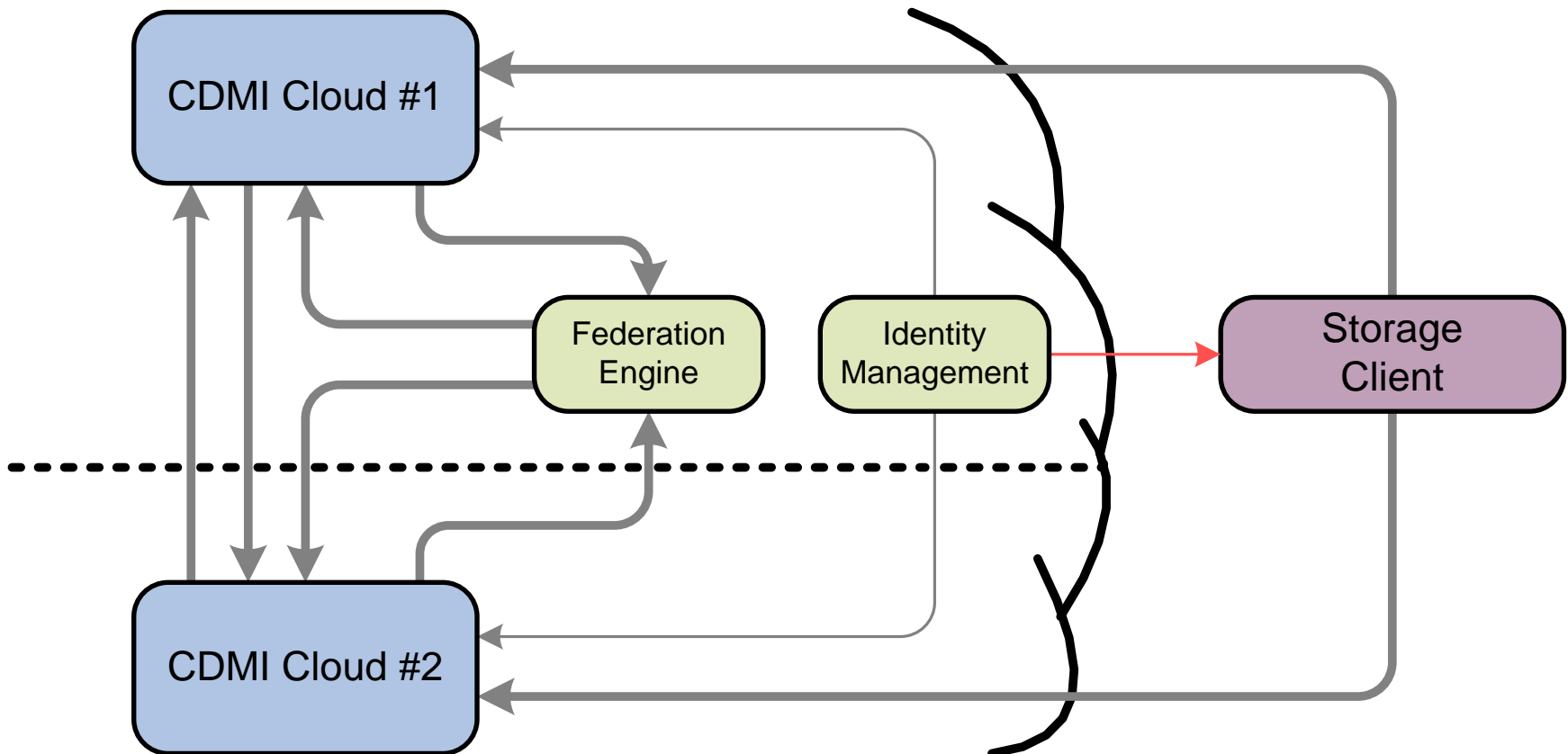
For more details on the CDMI standard, see:  
<http://www.snia.org/cloud/> and <http://cdmi.sniacloud.com/>

- ❑ We shall present federation functionality that can be accomplished using CDMI 1.0.2
- ❑ Our focus will be on federation use case #4, which is the most advanced scenario
- ❑ Use cases #1 through #3 can be accomplished by removing steps from use case #4

- ❑ Bidirectional Cloud Federation can be demonstrated by performing the following operations:
  - ❑ Step 1: Client obtains credentials
  - ❑ Step 2: Store an object into cloud 1
  - ❑ Steps 3 - 5: Federation Engine syncs to cloud 2
  - ❑ Step 6: Retrieve object from cloud 2
  - ❑ Step 7: Update the object on cloud 2
  - ❑ Steps 8 - 10: Federation Engine syncs to cloud 1
  - ❑ Step 11: Retrieve updated object from cloud 1

# Bidirectional Federation

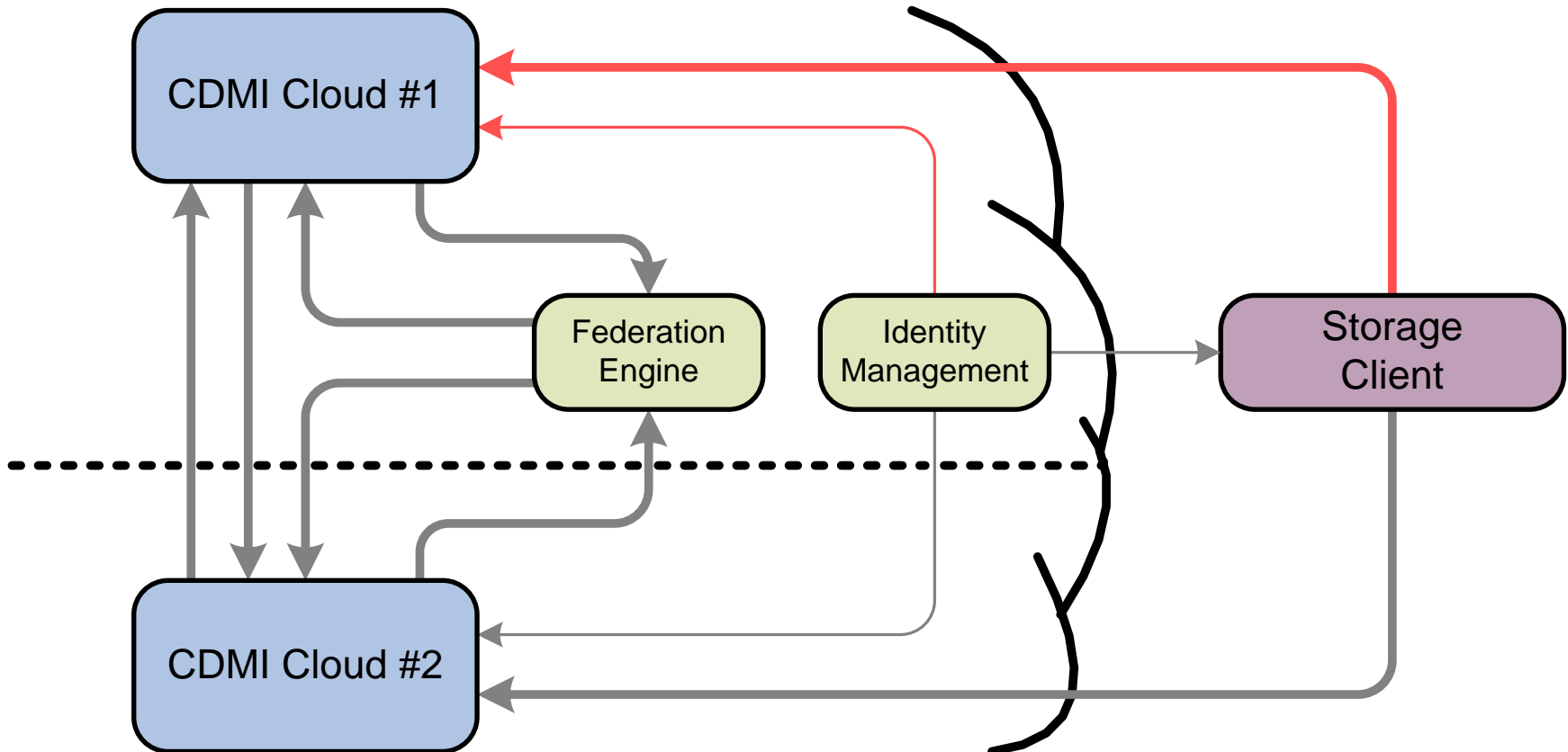
1. Client receives cloud credentials  
(Not part of the CDMI standard)





# Bidirectional Federation

## 2. Client stores object to CDMI Cloud #1



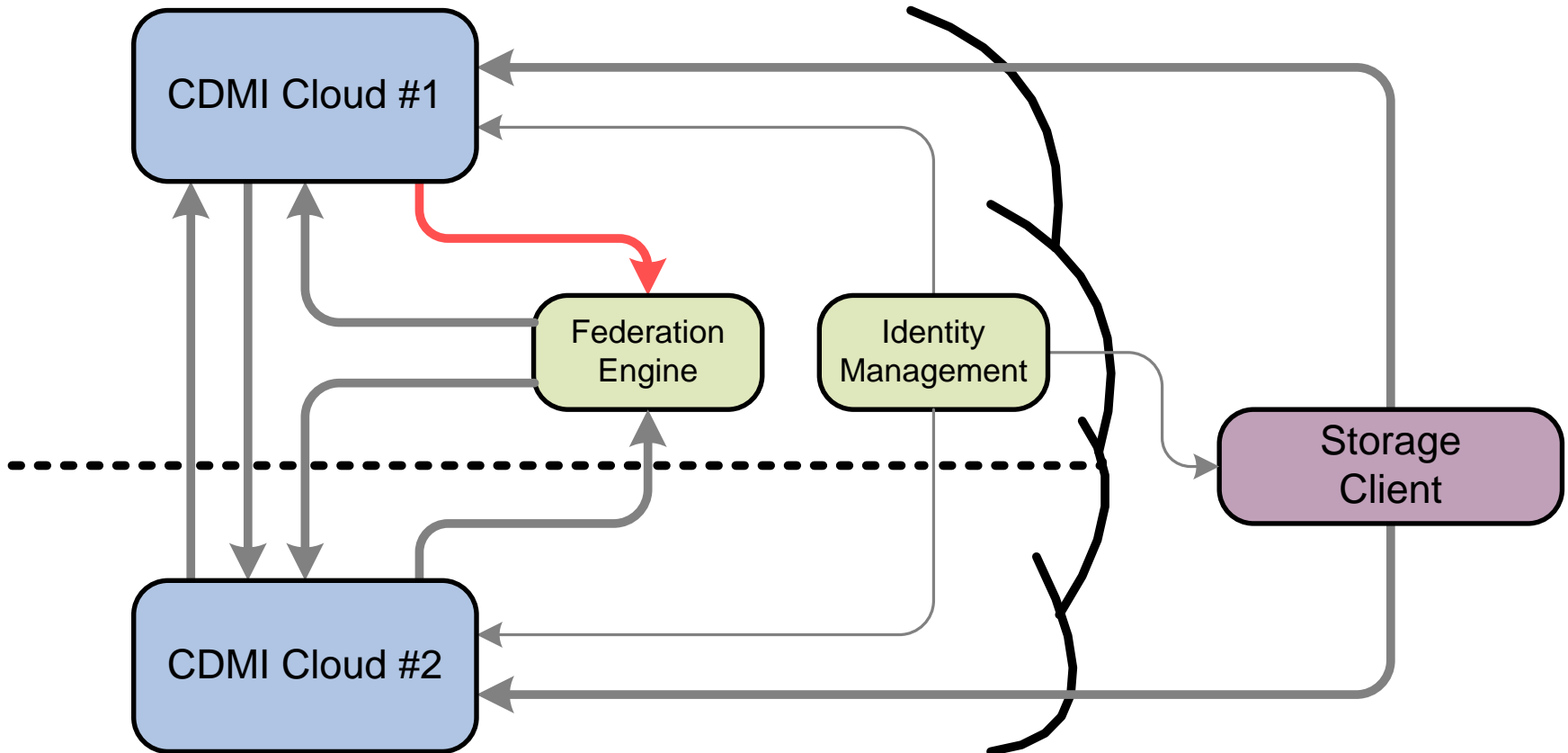
# Bidirectional Federation

## 2. Client stores object to CDMI Cloud #1

```
> PUT http://cloud1.example.com/MyDataObject.txt HTTP/1.1
> Host: cloud1.example.com
> Authorization: Basic cm9vdDpyb290
> Content-Type: text/plain;charset=utf-8
> Content-Length: 10
>
> Federation
< HTTP/1.1 201 Created
```

# Bidirectional Federation

## 3. Federation Engine receives notification



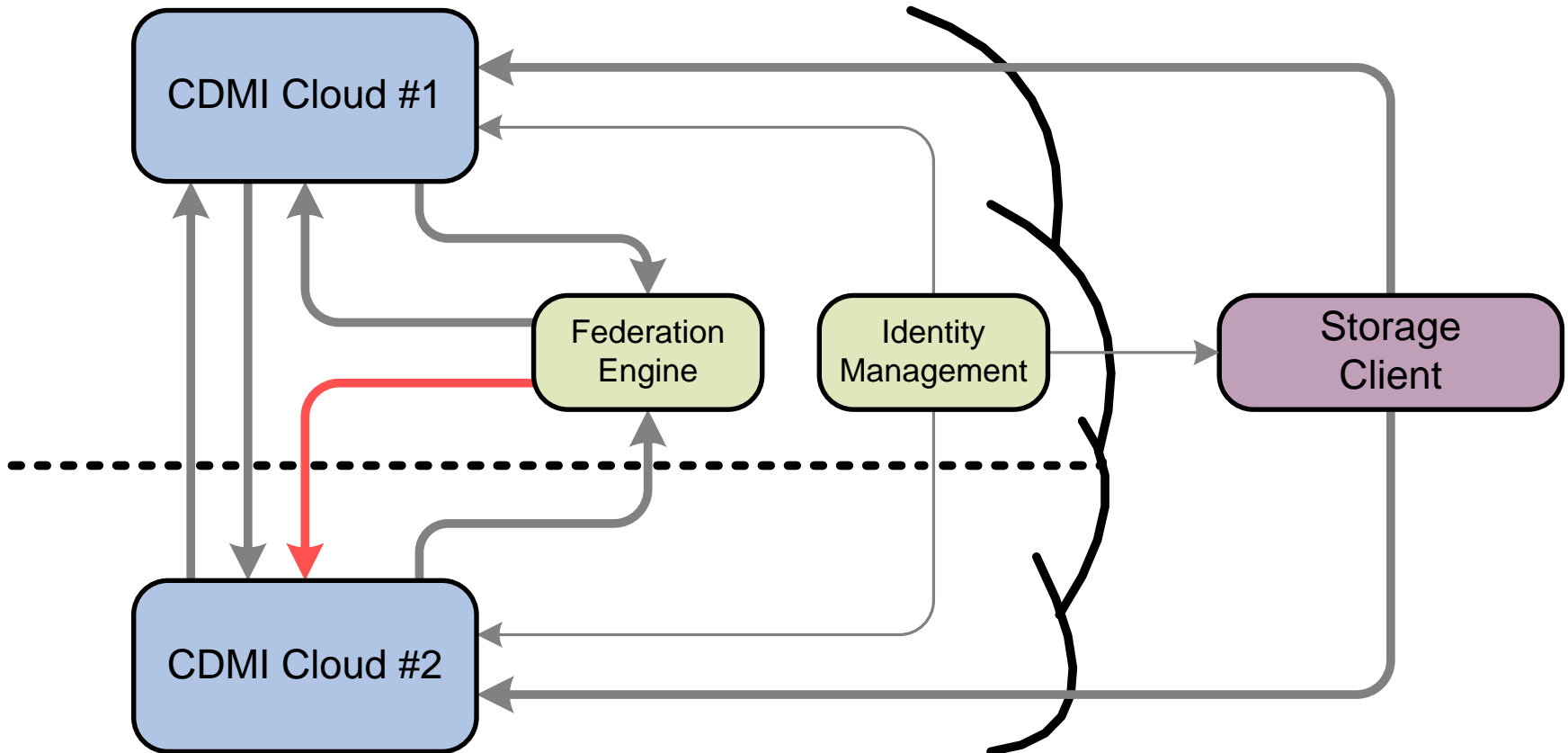


## 3. Federation Engine receives notification

```
{
  "cdmi_event": "cdmi_create_complete",
  "cdmi_event_result": "200 OK",
  "cdmi_event_time": "2012-09-01T12:09:54.572905Z",
  "objectID": "00007ED900105E7868BE445F79B09A95",
  "objectName": "/MyDataObject.txt",
  "metadata": {
    "cdmi_size": "10"
  }
}
```

# Bidirectional Federation

4. Federation Engine instructs Cloud #2 to deserialize object from Cloud #1

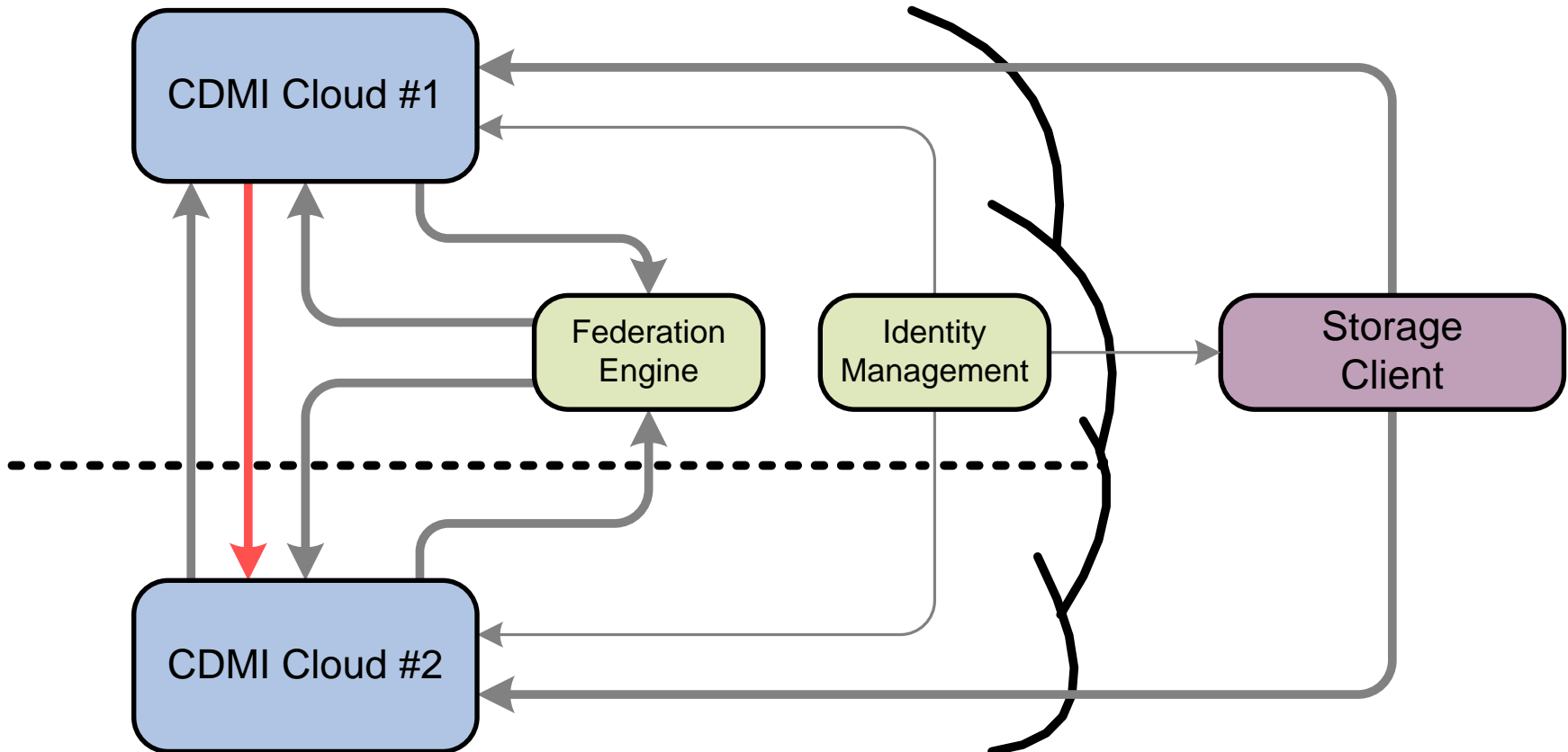


## 4. Federation Engine instructs Cloud #2 to deserialize object from Cloud #1

```
> PUT http://cloud2.example.com/MyDataObject.txt HTTP/1.1
> Host: cloud2.example.com
> Authorization: Basic cm9vdDpyb290
< Content-Type: application/cdmi-object
< Accept: application/cdmi-object
< X-CDMI-Specification-Version: 1.0.2
>
> {
>   "deserialize": "http://cloud1.example.com/MyDataObject.txt"
> }
< HTTP/1.1 201 Created
< Content-Type: application/cdmi-object
< X-CDMI-Specification-Version: 1.0.2
<
< {
<   ...
< }
```

# Bidirectional Federation

## 5. Cloud #2 retrieves the object from Cloud #1





# Bidirectional Federation

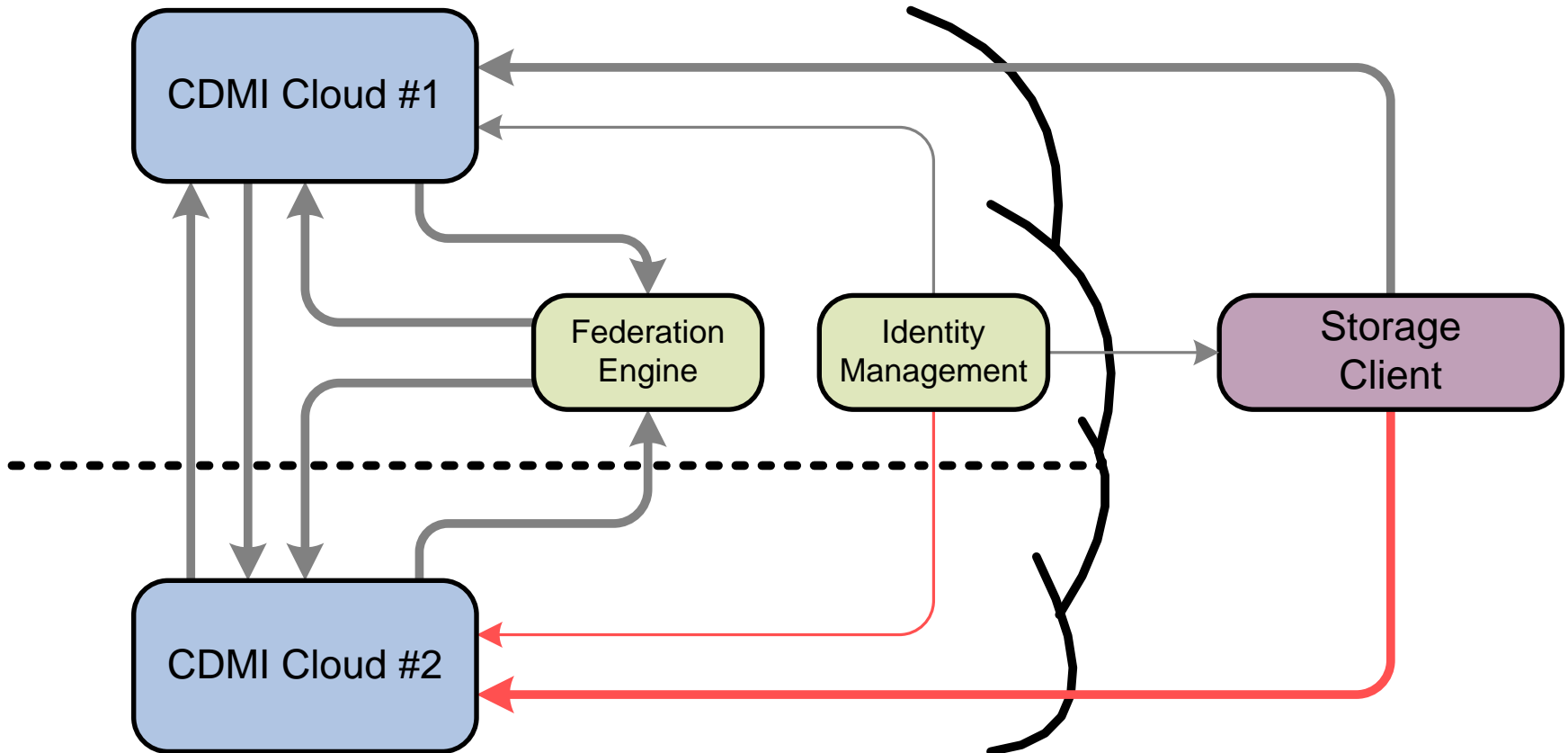
## 5. Cloud #2 retrieves the object from Cloud #1

```
> GET http://cloud1.example.com/MyDataObject.txt HTTP/1.1
> Host: cloud1.example.com
> Authorization: Basic cm9vdDpyb290
< Accept: application/cdmi-object
< X-CDMI-Specification-Version: 1.0.2

< HTTP/1.1 200 OK
< Content-Type: application/cdmi-object
< X-CDMI-Specification-Version: 1.0.2
<
< {
<   ...
< }
```

# Bidirectional Federation

6. Client can now access the object from Cloud #2 using same credentials as were used for Cloud #1



# Bidirectional Federation

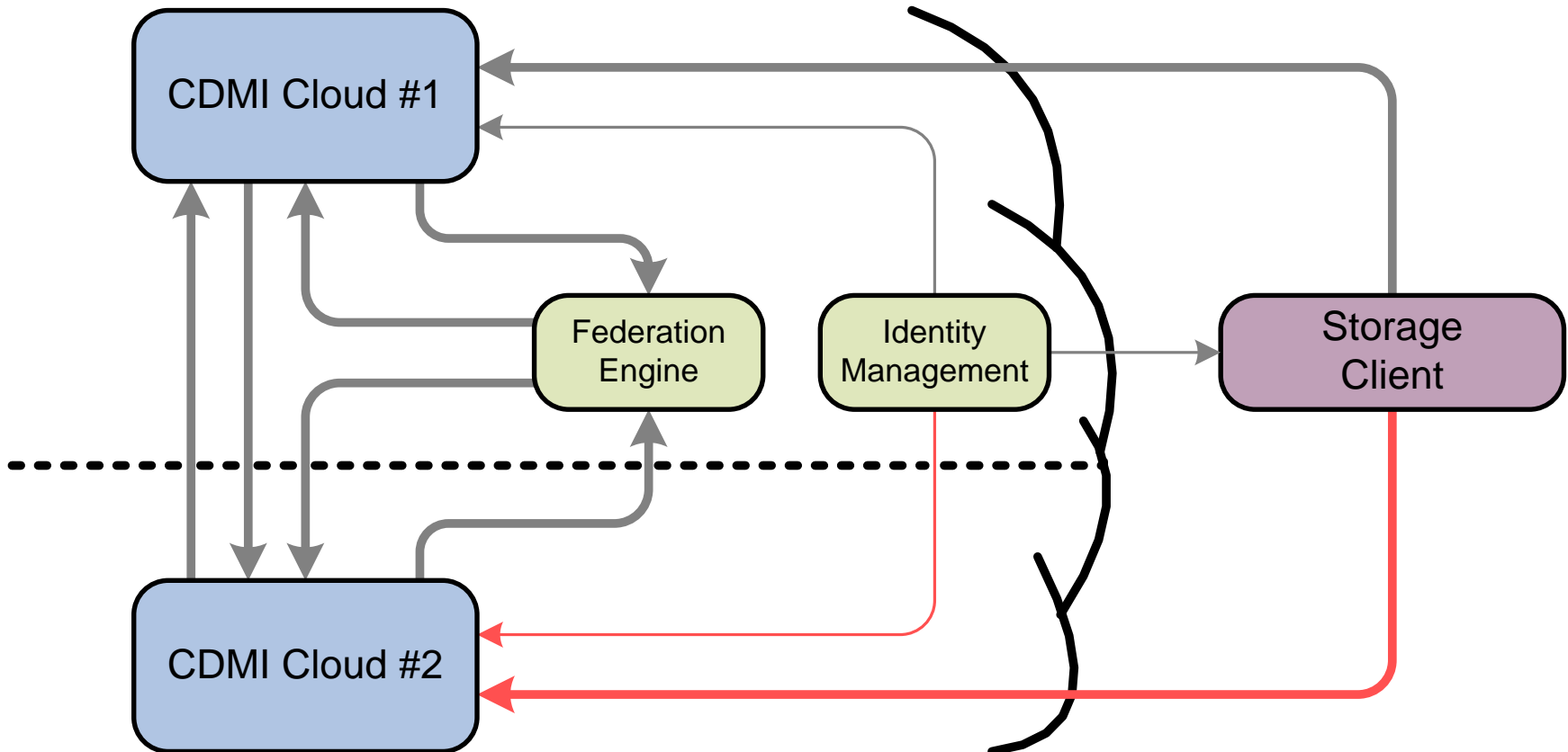
## 6. Client can now access the object from Cloud #2

```
> GET http://cloud2.example.com/MyDataObject.txt HTTP/1.1
> Host: cloud2.example.com
> Authorization: Basic cm9vdDpyb290

< HTTP/1.1 200 OK
< Content-Type: text/plain;charset=utf-8
< Content-Length: 10
<
< Federation
```

# Bidirectional Federation

## 7. Client Updates the object on Cloud #2

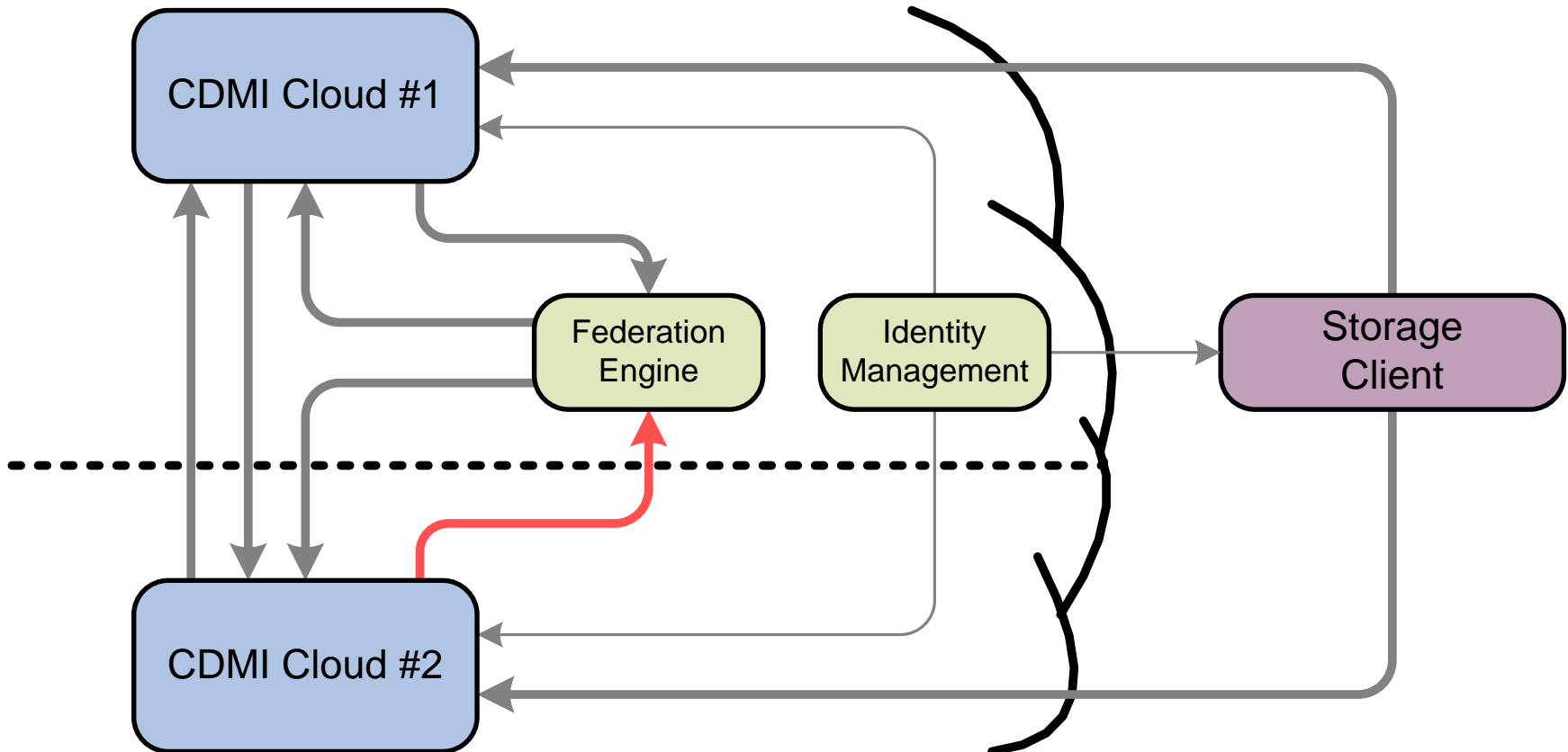


## 7. Client Updates the object on Cloud #2

```
> PUT http://cloud2.example.com/MyDataObject.txt HTTP/1.1
> Host: cloud2.example.com
> Authorization: Basic cm9vdDpyb290
> Content-Length: 9
>
> Federated
< HTTP/1.1 204 No Content
```

# Bidirectional Federation

## 8. Federation Engine receives notification



## 8. Federation Engine receives notification

```
> GET http://cloud2.example.com/FederationQueue?value;queueValues HTTP/1.1
> Host: cloud1.example.com
> Accept: application/cdmi-queue
> X-CDMI-Specification-Version: 1.0.2

< HTTP/1.1 200 OK
< Content-Type: application/cdmi-queue
< X-CDMI-Specification-Version: 1.0.2
<
< {
<   "queueValues" : "0-0",
<   "value" : [
<
"ewsgICAgImNkbWlfZXZlbnQioiAiY2RtaV9tb2RpZnlfY29tcGxldGUiLAsgICAgImNkbWlfZXZlbnRfcmlVzdWx
0IjogIjIwNCBObyBDb250ZW50IiwLICAgICJjZG1pX2V2ZW50X3RpbWUiOiAiMjAxMjAwOS0wMVQxMjoyMToxNi41N
zI5NTIhaIiwLICAgICJvYmplY3RJRCI6ICIwMDAwN0VEOTAwMTA1RTc4NjhCRTQ0NUY3OUiwOUe5NSIsCyAgICAib2J
qZWN0TmFtZSI6ICIvTXlEYXRhT2JqZWN0LnR4dCI6ICAgICAibWV0YWRhdGEiOiB7CyAgICAgICAgImNkbWlfZmZlc2l6Z
SI6IjM5IgsGICAgfQt9DQo="
<   ]
< }
```

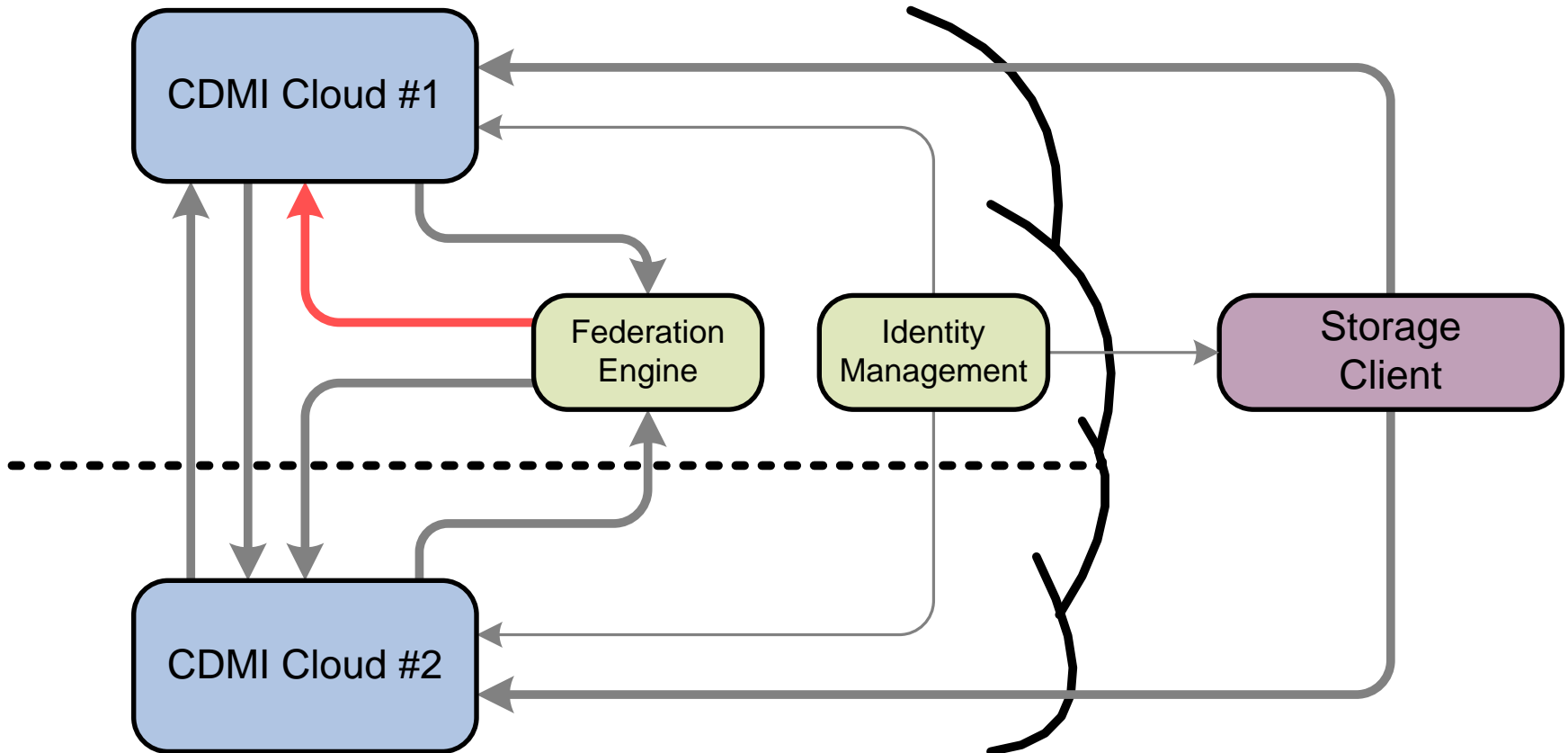
## 8. Federation Engine receives notification

```
{
  "cdmi_event": "cdmi_modify_complete",
  "cdmi_event_result": "204 No Content",
  "cdmi_event_time": "2012-09-01T12:21:16.572958Z",
  "objectID": "00007ED900105E7868BE445F79B09A95",
  "objectName": "/MyDataObject.txt",
  "metadata": {
    "cdmi_size": "9"
  }
}
```



# Bidirectional Federation

9. Federation Engine instructs Cloud #1 to deserialize object from Cloud #2

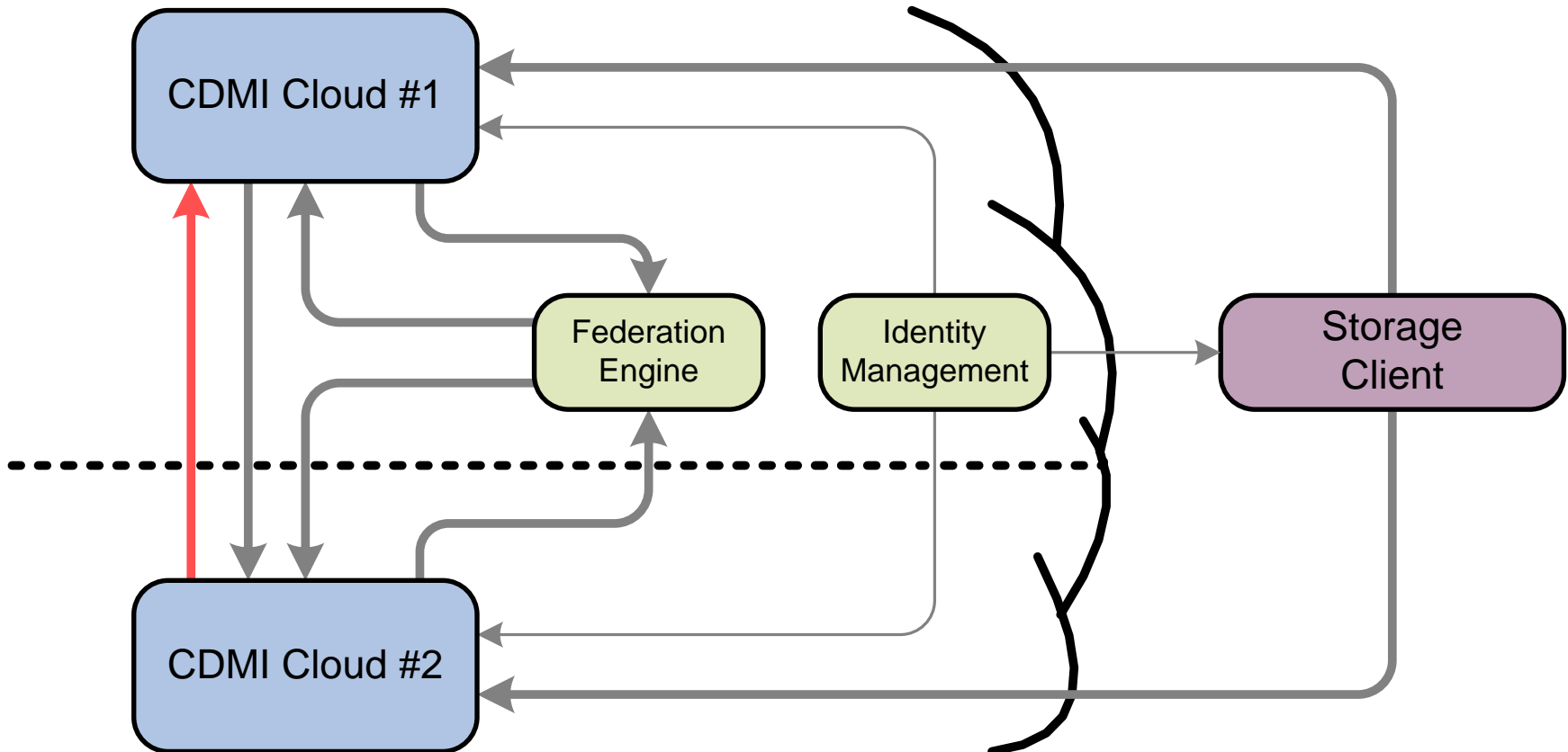


## 9. Federation Engine instructs Cloud #1 to deserialize object from Cloud #2

```
> PUT http://cloud1.example.com/cdmi_objectid/00007ED900105E7868BE445F79B09A95 HTTP/1.1
> Host: cloud1.example.com
> Authorization: Basic cm9vdDpyb290
< Content-Type: application/cdmi-object
< Accept: application/cdmi-object
< X-CDMI-Specification-Version: 1.0.2
>
> {
>   "deserialize":
> "http://cloud2.example.com/cdmi_objectid/00007ED900105E7868BE445F79B09A95"
> }
< HTTP/1.1 204 No Content
```

# Bidirectional Federation

10. Cloud #1 retrieves the object from Cloud #2



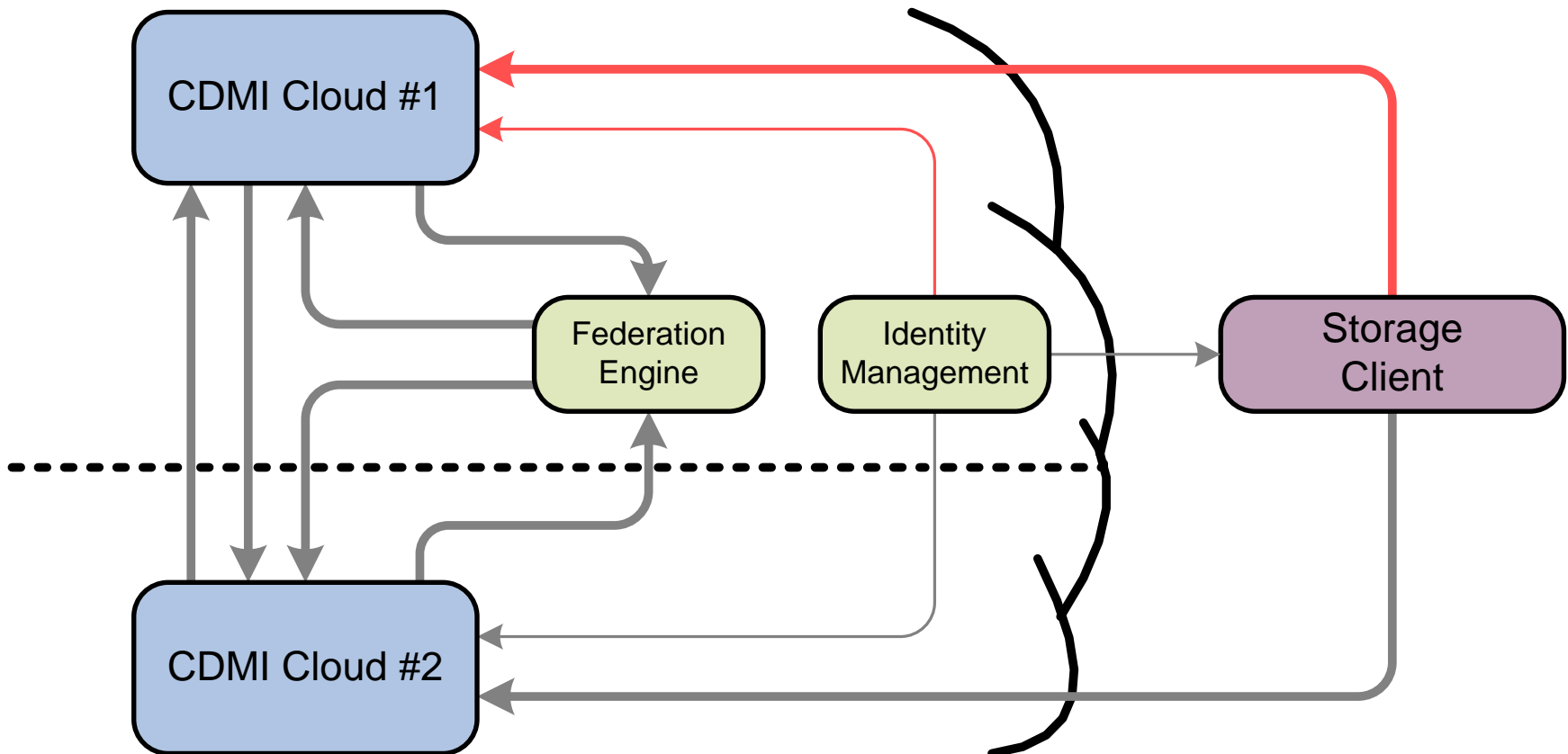
## 10. Cloud #1 retrieves the object from Cloud #2

```
> GET http://cloud2.example.com/cdmid_objectid/00007ED900105E7868BE445F79B09A95 HTTP/1.1
> Host: cloud2.example.com
> Authorization: Basic cm9vdDpyb290
< Accept: application/cdmid-object
< X-CDMI-Specification-Version: 1.0.2

< HTTP/1.1 200 OK
< Content-Type: application/cdmid-object
< X-CDMI-Specification-Version: 1.0.2
<
< {
<   ...
< }
```

# Bidirectional Federation

11. Client can access the updated object from Cloud #1



## 11. Client can now access the object from Cloud #1

```
> GET http://cloud1.example.com/MyDataObject.txt HTTP/1.1
> Host: cloud1.example.com
> Authorization: Basic cm9vdDpyb290

< HTTP/1.1 200 OK
< Content-Type: text/plain;charset=utf-8
< Content-Length: 9
<
< Federated
```

# CDMI Functionality Required

- ❑ Domain Membership Delegation
- ❑ CDMI Object Access by Name
- ❑ CDMI Object Access by ID
- ❑ Notification Queues
- ❑ Deserialization



## Ruby Federation Demonstration



- ❑ The above simple federation approach works for a single level of containers (“buckets”)
  - ❑ Multiple levels can be accommodated with additional logic to handle container rename and moves
  - ❑ Federation of versions requires additional deserialization logic to handle re-integrating historical versions
  
- ❑ Queue Federation requires quorum to ensure ordering

- ❑ Vendors are invited to become involved with the SNIA Cloud Storage Technical Working Group:
  - ❑ Contribute to the standardization of CDMI Federation
  - ❑ Contribute to ensuring multi-vendor interoperability
  - ❑ Contribute to future work areas including efficient update transfers and management

# Thank you!

## Questions and Answers

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