

# **CDMI™ and Cloud Federation Year 2**

**David Slik  
NetApp, Inc.**

# Session Agenda

- ❑ A Brief Overview of CDMI
- ❑ Federations Roadmap
- ❑ Federation Use Cases
- ❑ Requirements for Federation Targets
- ❑ Requirements for Federation Initiators
- ❑ Next Steps

## 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

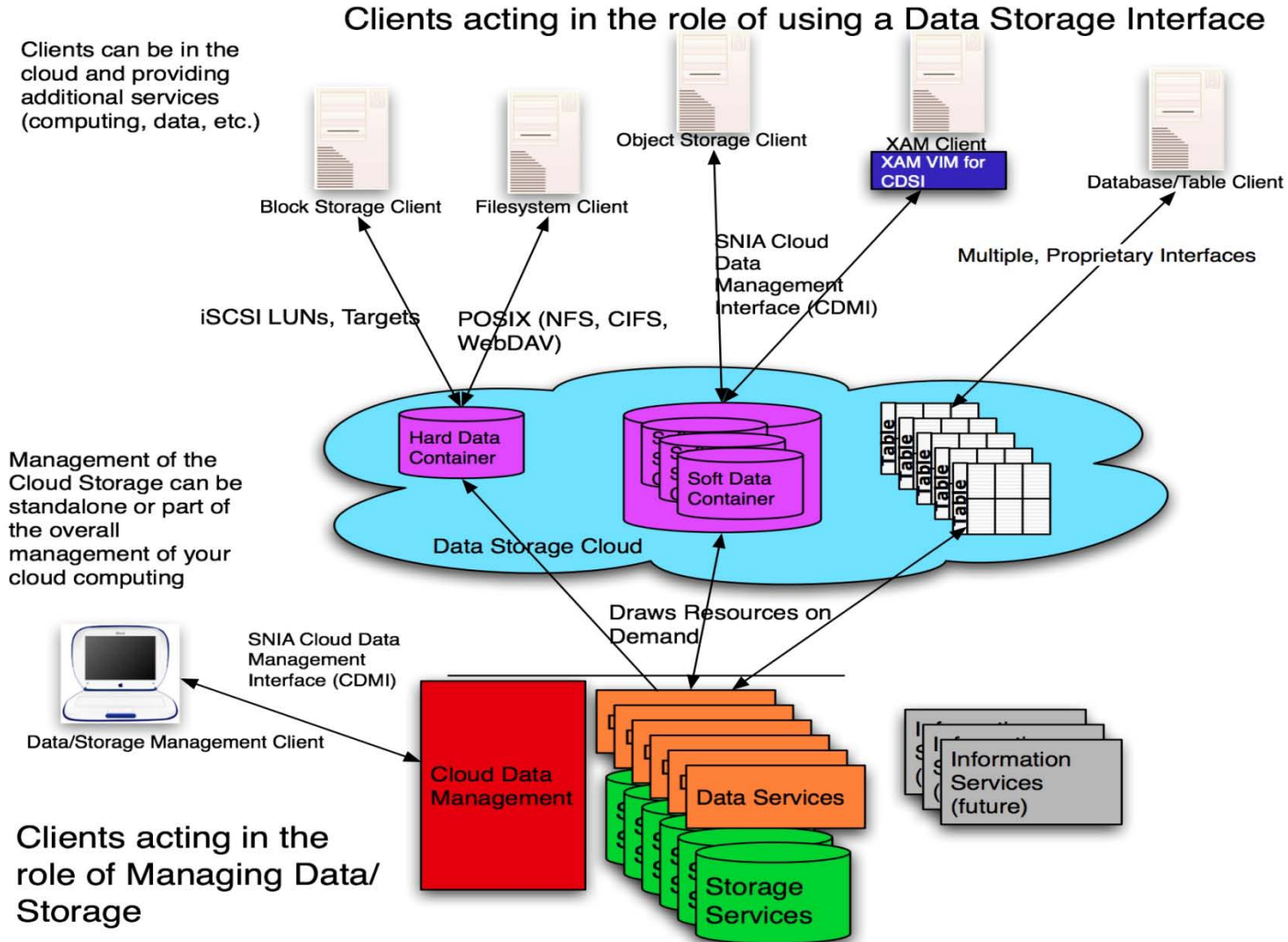
# A Brief Overview of CDMI

- 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/>

# Federation Roadmap

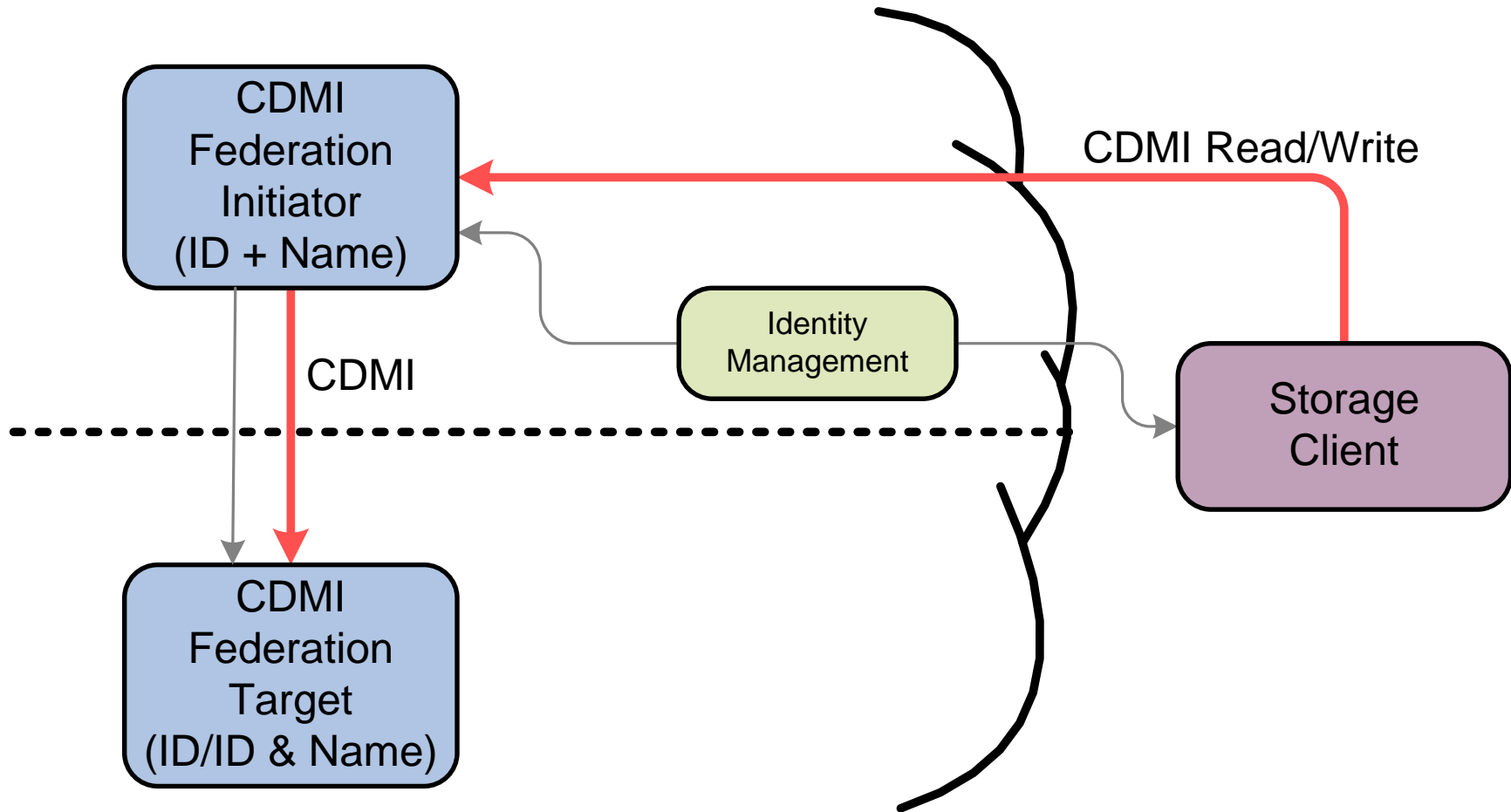
- ❑ The CDMI standard does not mention or define what federations are
- ❑ However, CDMI 1.0.1 does provide the primitive operations required to implement federations (as described later in this presentation)
- ❑ Standardizing federations is one of the items selected to be worked on as part of the CDMI 1.1 standard
  - ❑ Some work has already been done, in the area of terminology and use cases

# Federation Terminology

- ❑ Federation – The process by which two systems can establish a trusted relationship to exchange content via CDMI
- ❑ Peering – Where two CDMI systems provide client access to the same objects
- ❑ Proxying – Where a first CDMI system provides client access to objects stored on a second system
- ❑ Federation Initiator – The CDMI system that initiates operations within a federation relationship
- ❑ Federation Target – The CDMI system that processes operations within a federation relationship

# Federation Use Case #1

## Unidirectional Proxy Federation



- ❑ Unidirectional Proxy Federation
  - ❑ A Federation Initiator mirrors or delegates some or all stored objects onto a Federation Target, and clients can access objects from only the Initiator
  - ❑ Applications include:
    - ❑ Cloud Brokerage
    - ❑ Leverage of specialized storage services
  - ❑ Federation Target supports storage by ID
    - ❑ Typical of highly scalable object storage systems that are not based around the concept of a hierarchical path
  - ❑ Federation Target supports storage by ID and Name
    - ❑ Object IDs and names are preserved

# Federation Use Case #1

- ❑ Initiator Requirements (Non-ID Preserving)
  - ❑ Establish CDMI session with Target
  - ❑ Authenticate with Target
  - ❑ For each local object to be stored on the Target:
    - ❑ Create a new object on the target
    - ❑ Store the object ID for the new object on the target locally
  - ❑ For each update to a local object:
    - ❑ Update the corresponding object on the target
  - ❑ For each delete of a local object:
    - ❑ Delete the corresponding object on the target

# Federation Use Case #1

- ❑ Target Requirements (Non-ID Preserving)
  - ❑ Support CDMI Domains
  - ❑ Allow the creation of a local domain member for the Initiator
  - ❑ Provide basic Create/Retrieve/Update/Delete functions by ID

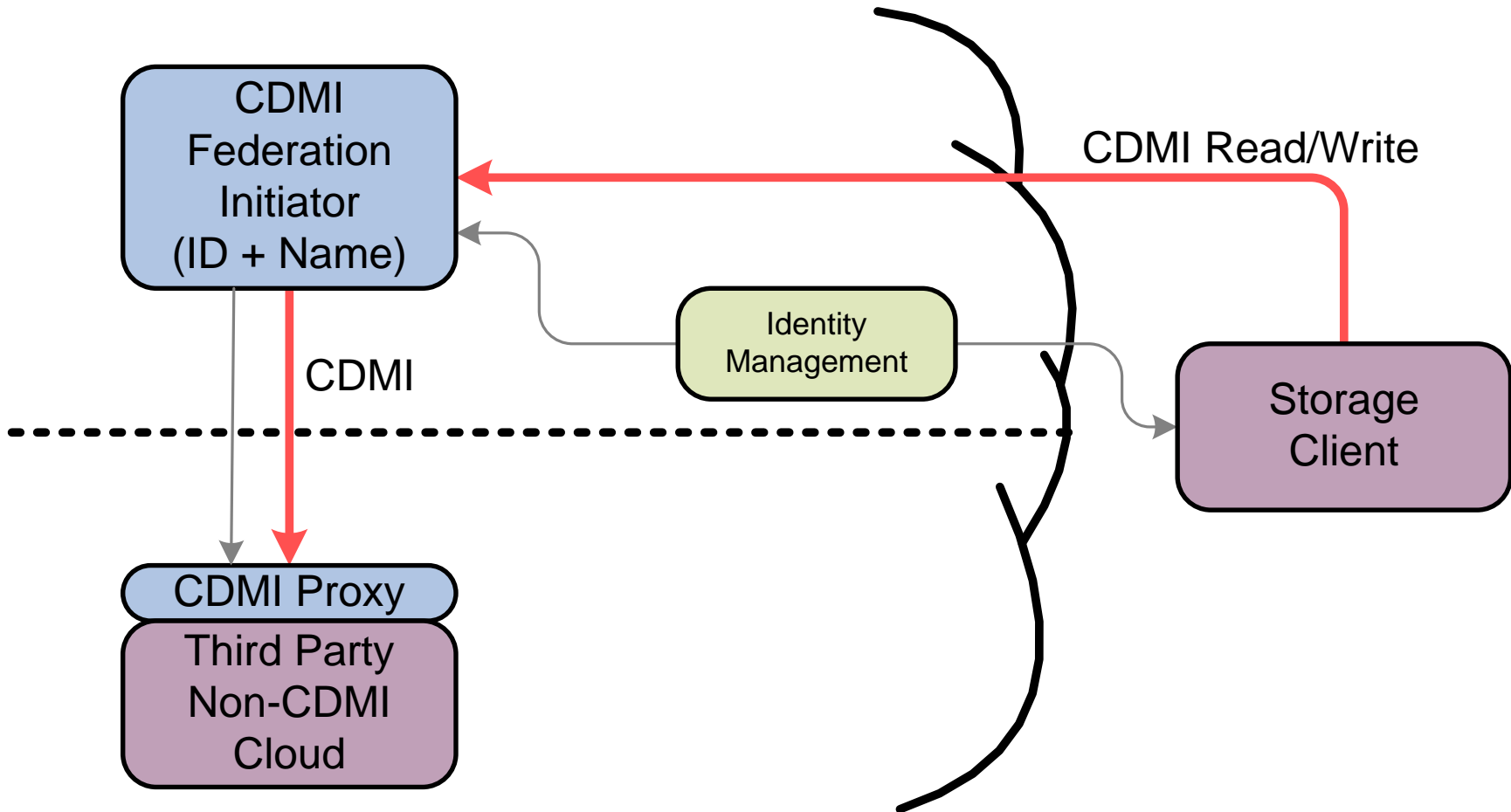
# Federation Use Case #1

- ❑ Initiator Requirements (ID Preserving)
  - ❑ Establish CDMI session with Target
  - ❑ Authenticate with Target
  - ❑ Have “backup\_operator” privilege on Target
  - ❑ For each locally created object:
    - ❑ Serialize the local object, and deserialize it on the target
  - ❑ For each update to a local object:
    - ❑ Serialize the updated object and deserialize it on the target
    - ❑ Serialize the change in object state (Future)
  - ❑ For each delete of a local object:
    - ❑ Delete the corresponding object on the target

- ❑ Target Requirements (ID Preserving)
  - ❑ Support CDMI Domains, and the creation of a domain with a user-provided ID or support the deserialization of domains
  - ❑ Allow the creation of a local domain member for the Initiator, with the “backup\_operator” privilege
  - ❑ Provide basic Create/Retrieve/Update/Delete functions by ID
  - ❑ Support deserialization

# Federation Use Case #2

## Unidirectional Proxy Federation



# Federation Use Case #2

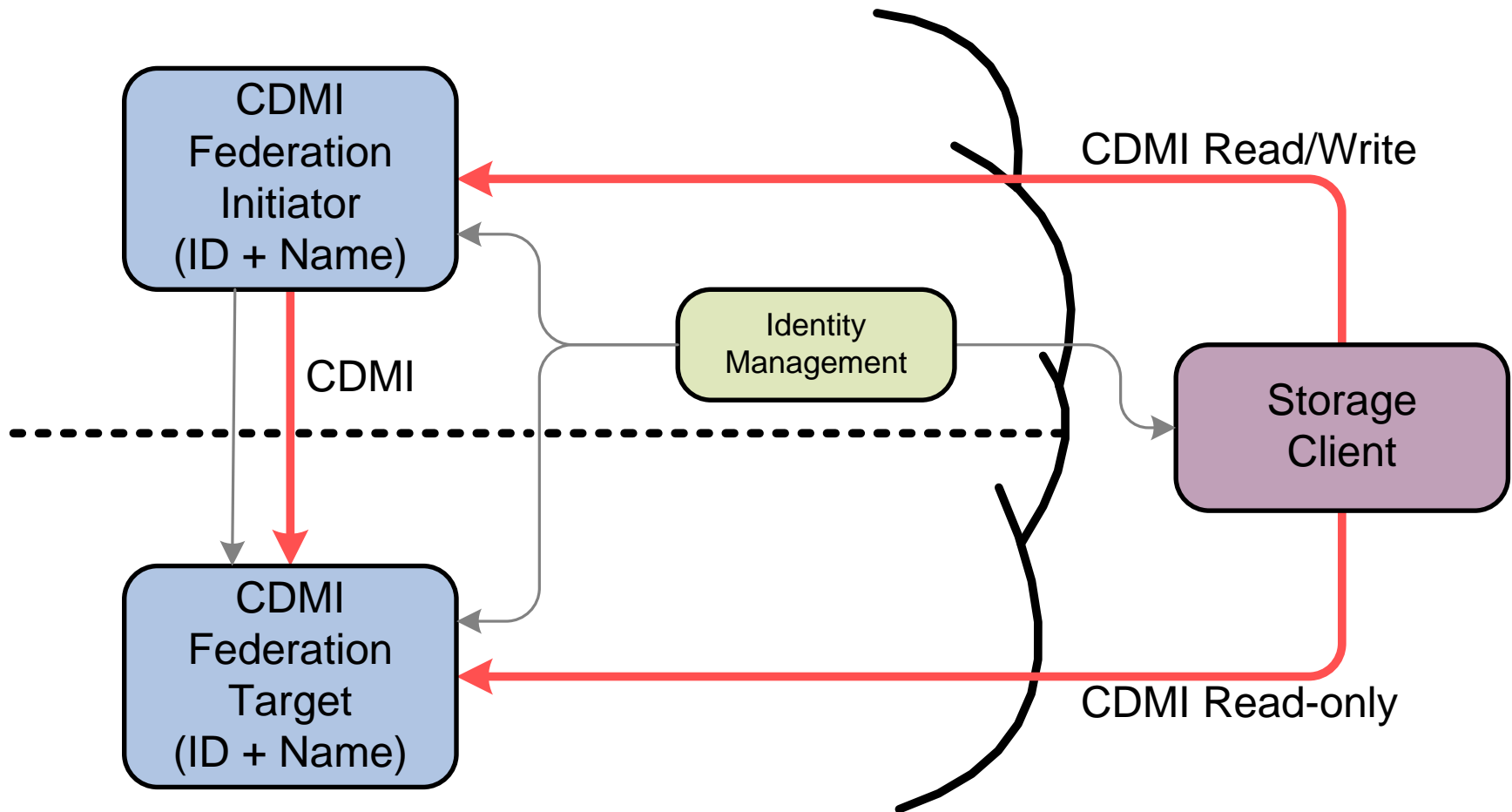
- ❑ Unidirectional Proxy Federation to a Non-CDMI Cloud
  - ❑ A Federation Initiator mirrors or delegates some or all stored objects onto a Non-CDMI Cloud through a CDMI Proxy acting as a Federation Target
  - ❑ Applications include:
    - ❑ Providing multi-system interoperability between non-CDMI and CDMI cloud storage systems
  - ❑ Examples include:
    - ❑ Remote VM instances that translates CDMI operations into non-CDMI cloud operations
    - ❑ Local VM that encrypts objects before storing them to a remote storage cloud via non-CDMI operations

# Federation Use Case #2

- ❑ Initiator Requirements
  - ❑ All requirements for Use Case #1
  
- ❑ Target Requirements
  - ❑ All requirements for Use Case #1
  
- ❑ CDMI Proxy translates CDMI operations into the native cloud commands for the non-CDMI cloud being proxied
- ❑ CDMI Proxy maintains credentials to communicate with the non-CDMI cloud, or translates the provided CDMI credentials

# Federation Use Case #3

## Unidirectional Peering Federation



# Federation Use Case #3

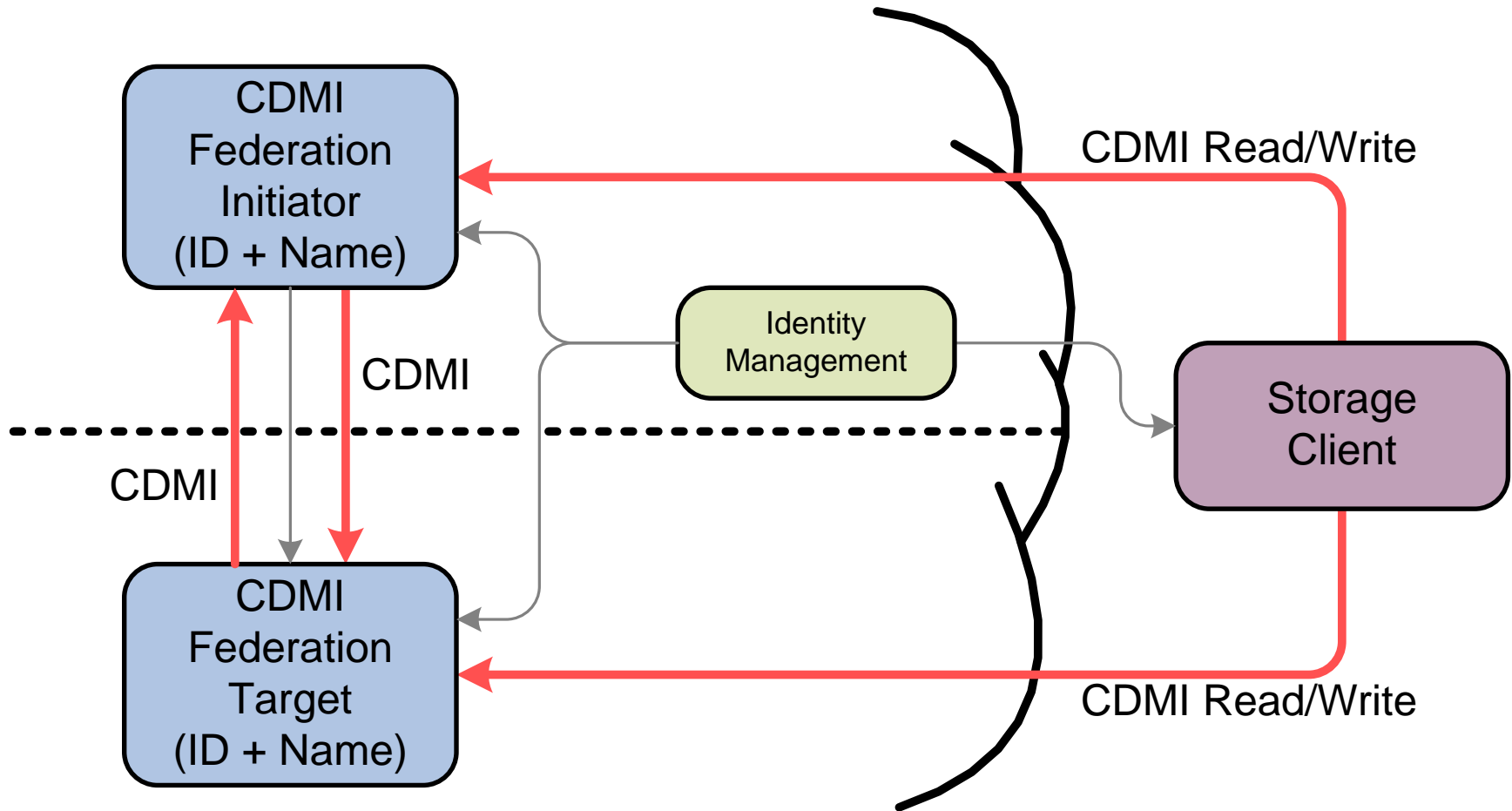
- ❑ Unidirectional Peering Federation
  - ❑ A Federation Initiator mirrors or delegates some or all stored objects onto a Federation Target, and clients can access objects from the Initiator and the Target
  - ❑ Applications include:
    - ❑ Migration
    - ❑ Remote Backup
    - ❑ Edge Caching
    - ❑ Read Performance Scaling
    - ❑ Content Delivery Networks

# Federation Use Case #3

- ❑ Initiator Requirements
  - ❑ All requirements for Use Case #1 (ID Preserving)
  - ❑ Support delegated client authentication
  
- ❑ Target Requirements
  - ❑ All requirements for Use Case #1 (ID Preserving)
  - ❑ Support delegated client authentication
  - ❑ Provide basic Create/Retrieve/Update/Delete functions by **ID and Name**

# Federation Use Case #4

## Bidirectional Peering Federation



# Federation Use Case #4

- ❑ Bidirectional Peering Federation
  - ❑ A Federation Initiator mirrors or delegates some or all stored objects onto a Federation Target, and clients can access objects from the Initiator and the Target
  - ❑ Applications include:
    - ❑ Namespace Scaling
    - ❑ Read/Write Performance Scaling
    - ❑ Branch Office Access

# Federation Use Case #4

- ❑ Initiator Requirements
  - ❑ All requirements for Use Case #3
  - ❑ Support retrieving notifications from the target notification queue
  - ❑ For each remotely created object:
    - ❑ Serialize the remote object, and deserialize it locally
  - ❑ For each update to a remote object:
    - ❑ Serialize the updated remote object and deserialize it locally
  - ❑ For each delete of a remote object:
    - ❑ Delete the corresponding local object

# Federation Use Case #4

- ❑ Target Requirements
  - ❑ All requirements for Use Case #3
  - ❑ Support Notification Queues



## Ruby Federation Demonstration

- ❑ Vendors are invited to become involved with the SNIA Cloud Storage Technical Working Group to:
  - ❑ 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)