SNIA | Cloud Storage Technologies

Intelligent Data Management: Shaping the Future for AI Workloads

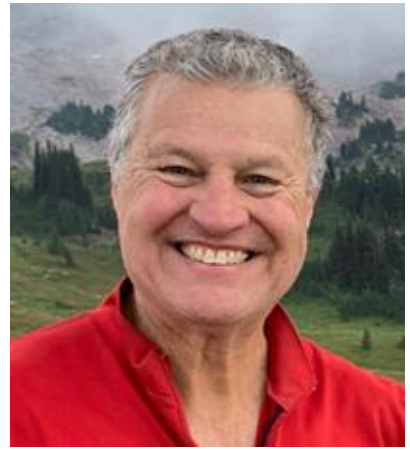
Live Webinar
January 20, 2026
10:00 am PT / 1:00 pm ET

Today's Presenters

David Slik
Scientist
Huawei Technologies
Canada Co., Ltd

Gary Mazzaferro
Industry Advisor

Carl D'Halluin
CTO
Datadobi



Michael Hoard
Moderator
SNIA
CST Chair

SNIA | CST A SNIA Community

SNIA Cloud Object Storage Plugfests sponsored by CST

SNIA | Cloud Storage Technologies

https://snia.org/groups/cst

Committed to the adoption, growth and standardization of **intelligent data storage usage** in cloud infrastructures

- New simplified SNIA Membership Model
- Pay one fee for SNIA Membership – Participate in ANY group!
- Join SNIA groups at https://www.snia.org/member_com

The SNIA Community

200
industry leading organizations

2,000
active contributing members

50,000
IT end users & storage pros worldwide

SNIA | CST A SNIA Community

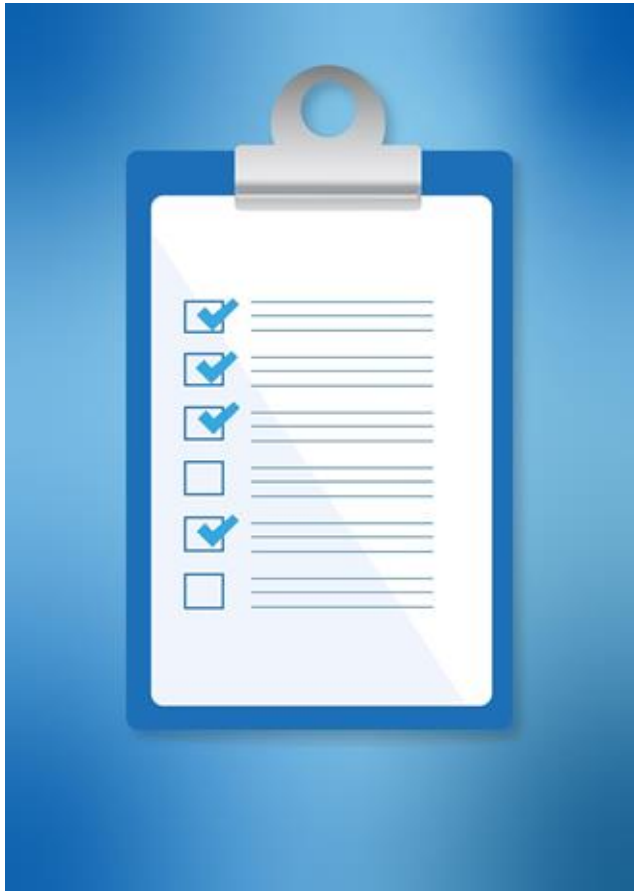
SNIA Legal Notice

- The material contained in this presentation is copyrighted by SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of SNIA.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

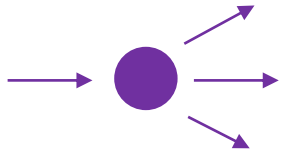
  NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Agenda

- Intelligent Data Management Challenges
- What is SNIA CDMI™ (Cloud Data Management Interface)?
- Challenge 1 – Access Method Discovery
- Challenge 2 – Metadata Portability
- Challenge 3 – AI Agent-driven Operation
- Getting Involved



Importance of Data and Metadata Portability

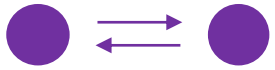


Data Exchange & Interoperability

Producing ≠ consuming (meta)data. AI breaks silos!

Data Migration

Storage refresh or consolidation, cloud adoption, data relocation, …

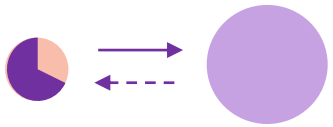


Data Replication

Cross-vendor data replication, hybrid cloud copy, secure vault copy, …

Data Copying

Feeding to AI for training & inference, copy to collaborators, …



Data Archival

Long-term data preservation, e-Discovery, data lifecycle

SNIA | CST A SNIA Community

Areas of Data and Metadata Portability

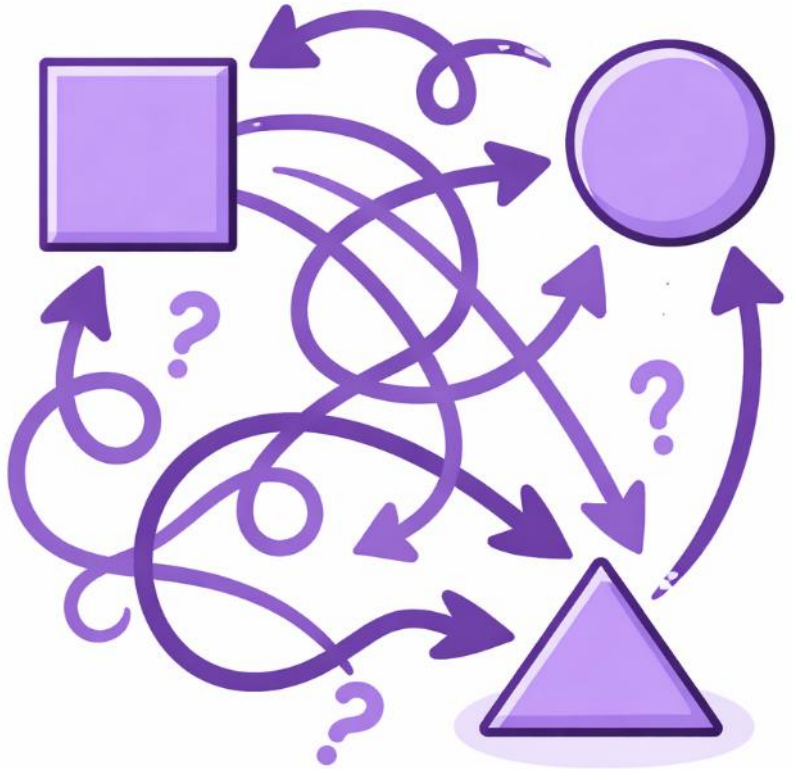
- **Access**
  - File Protocols
    - SMB, NFS, multi-protocol access, local FS, FUSE & custom drivers, …
  - Object Protocols
    - S3, Azure Blob, Google Cloud Storage, CDMI, Swift, CAS, FTP, …
- **Data**
  - Path & name
  - Content
  - Content metadata header
- **Metadata**
  - System metadata
  - User metadata
  - Administrative metadata (e.g. tags)
  - Derived / enriched metadata

Lessons From the Field: Portability Pitfalls (1/2)

- **Access**
  - Discovery
  - Protocol, authentication, transfer encoding, chunked and parallel access, …
- **Data**
  - File format
  - Content Encoding `Müller - M?ller - MÃ¼ller - M�ller`
  - Content Size limits
  - Alternate Data Streams (NTFS)
- **File/object Name & Path**
  - Encoding & case sensitivity `invoice.pdf vs INVOICE.PDF`
  - Illegal characters `myObject/`
  - Name/path/key limits
  - Number of files per dir

Lessons From the Field: Portability Pitfalls (2/2)

- **Metadata**
  - Semantics
  - Format, encoding, case sensitivity, illegal chars
  - Size & count limits
  - Timestamp semantics
  - Timestamp granularity & timezone
  - Version number/id
  - Ownership & Permissions model
  - Encryption model
  - Data retention model
  - Metadata security and access model

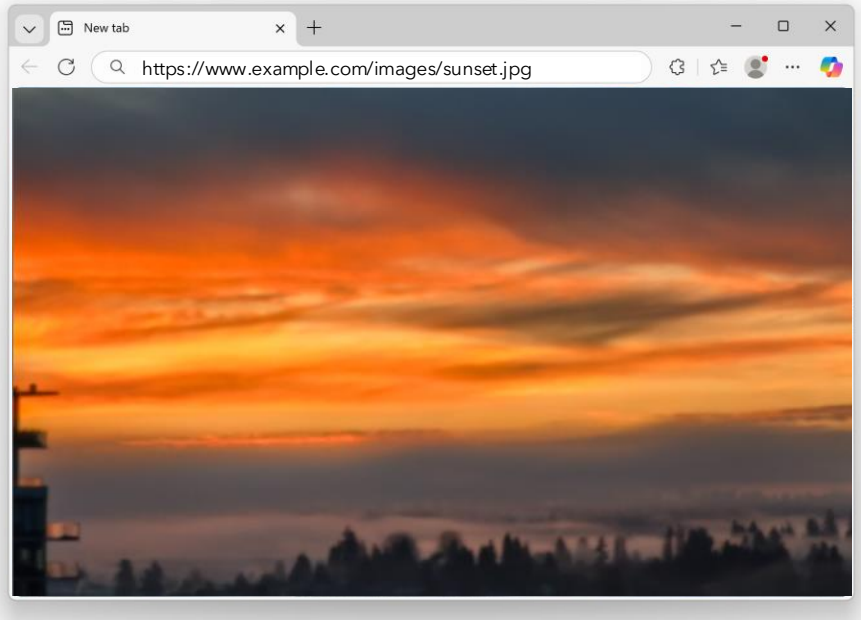


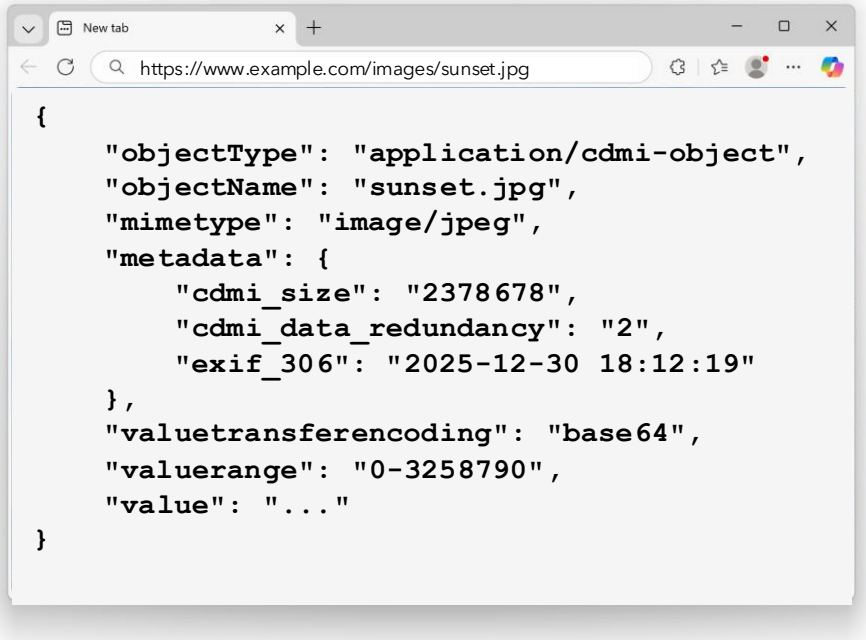
Intelligent Data Management Challenges

- Challenge 1 – Access Method Discovery
  - How do I discover how I can access my data?
  - How do I discover what is the best way to access my data?
- Challenge 2 – Data and Metadata Portability
  - When I access it, how do I learn about my data?
  - When I access metadata, how do I interpret it?
- Challenge 3 – AI Agent-driven Operation
  - When I start using AI agents, how do they access data?
  - When I start using AI agents, how do they access metadata?
  - When I start using AI agents, how do they manage storage?

An ISO Standard for Data Management helps!

- CDMI is a **Management Representation** that enables data discovery, data portability, and declarative data management.



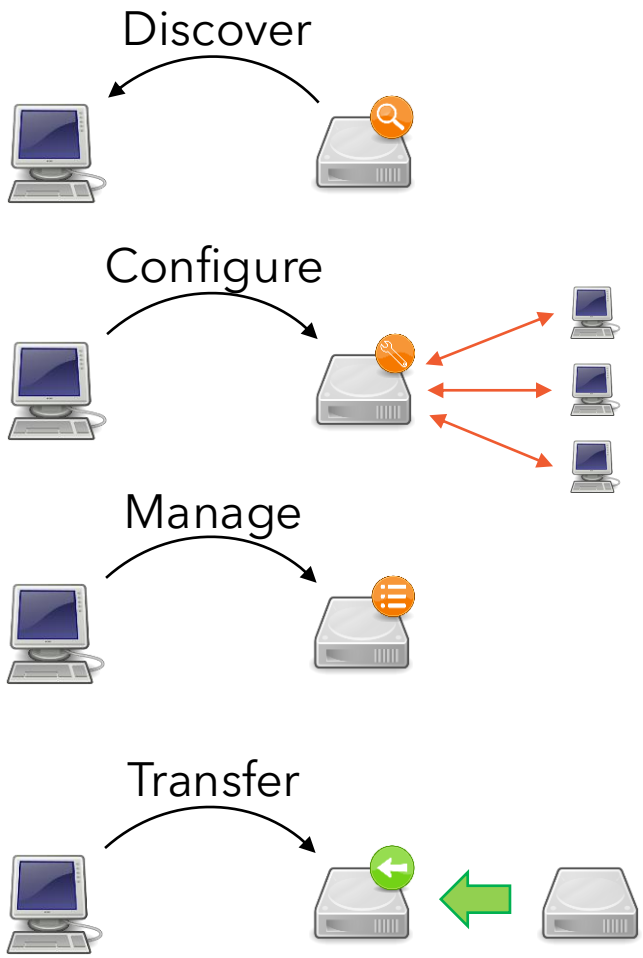


- Every resource (file, object, directory, etc.) has a CDMI representation.

SNIA | CST A SNIA Community

What is CDMI used for?

- The SNIA Cloud Data Management Interface (CDMI), also known as ISO/IEC 17826, provides a vendor-neutral standard interface for multi-protocol **discovery, configuration, management and portable data movement**.
- Implementers should consider using CDMI when applications require **protocol-independent data management**.
- CDMI allows clients to use a lightweight and consistent management interface **independent of data access protocols**.
- By enabling management operations outside each specific data access protocol, **CDMI works across multiple data access protocols and multiple deployment models**, including on-premises, hybrid, and cloud deployment.



Why is CDMI valuable?

- It's an international, formal, unencumbered, and open standard:
  - CDMI 2.0 is an international standard (ISO/IEC 17826)
  - CDMI has a formal specification (https://www.snia.org/sites/default/files/technical-work/cdmi/release/CDMI-v2.0.0.pdf)
  - Contributions from all major storage venders under SNIA's IP policy
  - SNIA does not require licenses, and CDMI is open to anyone to implement
  - SNIA welcomes all interested parties to contribute and participate in the future development of CDMI
- It's easy for developers to implement:
  - **Implement only what you need** – the only required part of the standard is capabilities (to discover what is implemented)
  - Extend as you need – CDMI provides many mechanisms for vendor-specific extensions
  - Vendor extensions welcomed – SNIA welcomes vendor contributions to extend CDMI
  - Works over all existing authentication systems, with no additional development required
  - Works with vendor's existing URI-based data addressing schemes
- It's simple for users:
  - All clients need is curl (for scripts), or the basic HTTP libraries included as part of every programming language

Who Uses CDMI?

- Scientific Computing
  - dCache, OneData, Indigo, Cern LHC FTS, CTA, eXtreme DataCloud, Digital Twin Engine, EGI CMS, EGI Federated Cloud, Venus-C, GFAL, SToRM, JClouds, and many more
- Governments
  - Canadian Data Governance Standardization Roadmap, UN Records and Archives, NATO Interoperability Standard, India Enterprise Architecture Framework, India Data Storage Infrastructure, Hong Kong Interoperability Framework for e-Government, Nigeria Cloud Computing Policy, Oman Cloud and Hosting Services, Kenya Cloud, and many more
- Cultural and Data Preservation
  - EUDAT, B2Stage, DRASTIC, SIRF, Eureka3D, IBSER, and more
- Commercial Systems (Internal uses)
  - Electronic health records, banking document retention, enterprise document management, video archiving, virtual machine storage, and many more

Addressing Data Management Challenges

- **Challenge 1: AI requires new ways to access data. How can clients discover access methods and obtain configuration information to permit access?**
  - Solution: Enhanced resource discovery to enable selection from multiple data access methods (e.g. NFS, NFS-over-RDMA, S3, S3-over-RDMA, etc.)
- **Challenge 2: AI enables metadata extraction at scale and creates structured metadata for knowledge graphs. How can CDMI ensure metadata portability?**
  - Solution: Self-describing data graphs to enhance metadata portability and enable attachment of knowledge graphs
- **Challenge 3: AI agent-driven data discovery, access and management**
  - Solution: MCP transport to allow use of CDMI by AI agents

Challenge 1 – Access Method Discovery

- If a client has a path to a file or object, that path doesn't indicate the best way to access it, or provide enough information to access it. E.g.:
  - s3://my-training-bucket/dataset_2/file.jpg
  - /home/user/files/model_2_5.safetensors
- CDMI 2.0 includes an "exports" field to allow clients to specify and discover how a given object is exported
  - But requires you to traverse the entire directory hierarchy to determine this
  - This is not ideal!
- CDMI 3.0 extends this to add a pre-traversed "exports_provided" field
  - Indicates all available methods to access the resource
  - Indicates additional information required to successfully access the resource

Challenge 1 – Access Method Discovery

- Example "exports_provided" for "file.jpg":

```
{
    "exports_provided": [ {
            "client_protocol": "NFSv4.1",
            "client_uri": "nfs://home/user/files/dataset_2/file.jpg",
            "client_protocol_transport": "TCP/IP",
            "client_flags": [ "hard", "port=2049", "sec=krb5p" ],
            "export_definition_uri": "/home/user/"
        }, {
            "client_protocol": "S3",
            "client_uri": "https://dataset2.example.com/file.jpg",
            "client_protocol_transport": "TCP/IP",
            "client_capabilities": [ "rdma-put", "rdma-get" ],
            "export_definition_uri": "/home/user/dataset_2/"
        }
    ]
}
```


Challenge 1 – Access Method Discovery

- The "exports_provided" indicates:
  1. An ordered list of access methods as ranked by the server. The client can re-order and select from this list based on client compatibility and preferences.
  2. If the data is replicated, local replicas can be listed first. Client proximity can be based on the network location of the client as determined by the server.
  3. The access path, which can vary based on the access method
  4. What client configuration flags and settings are required to successfully connect
  5. What optional capabilities are supported, such as RDMA support.
  6. Where the share definition is specified, when defined using CDMI.
- This allows any client to find the optimal compatible data access method with a single HTTP GET request
- Functionality can easily be integrated into catalogs and directories

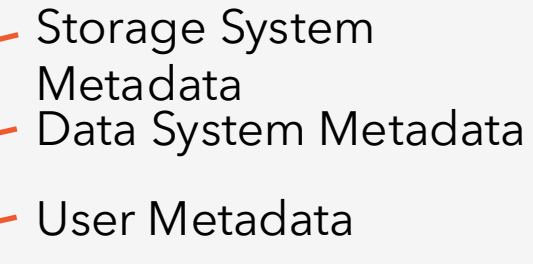
A quick aside…

- If you're interested in standardizing how object protocols such as S3 can be enhanced to use RDMA to accelerate data I/O, SNIA has a newly launched technical working group that is working on this problem
- Find out more and join the new *Accelerated Object I/O TWG* here:
  - https://www.snia.org/technical-work-groups

Challenge 2 – Data & Metadata Portability

- CDMI allows metadata to be attached to each file/object/directory

```
{
    "objectType": "application/cdmi-object",
    "objectName": "sunset.jpg",
    "mimetype": "image/jpeg",
    "metadata": {
        "cdmi_size": "2378678",
        "cdmi_data_redundancy": "2",
        "exif_306": "2025-12-30 18:12:19",
        "exif_33432": "Example Inc."
    },
    "valuetransferencoding": "base64",
    "valuerange": "0-3258790",
    "value": "..."
}
```



Challenge 2 – Data & Metadata Portability

- Standardized metadata helps
  - File systems – File size, file name, file creation date, etc.
  - JPEG EXIF – Manufacturer, model, orientation, resolution, exposure, etc.
- But… Even standardized metadata still has ambiguities!
  - Character sets, time zones, units, differences between versions, bugs, etc.
- Outside of standardized metadata, every application stores metadata differently:
  - Custom schemas, namespace collisions, dependencies between metadata

- **How can we make metadata self-describing?**

Challenges with the CDMI Metadata Model

- No mechanism to indicate the meaning of metadata
  - E.g. "cdmi_size" is defined by SNIA as part of the CDMI specification
  - E.g. "exif_306" is defined as part of the EXIF specification
- No mechanism to attach additional information to metadata
  - E.g. "cdmi_size" is measured in bytes
  - E.g. "exif_306" needs to indicate the time zone
- No mechanism to link metadata to other metadata / objects / web resources
  - E.g. How do we link a copyright to a web site?
  - E.g. How do we indicate what software populated a metadata item?

- This needs to be accomplished without altering the existing metadata, which is owned by the application

CDMI 3.0 "rel" Metadata Relationships

- CDMI addresses these challenges by adding a new top-level "rel" field:

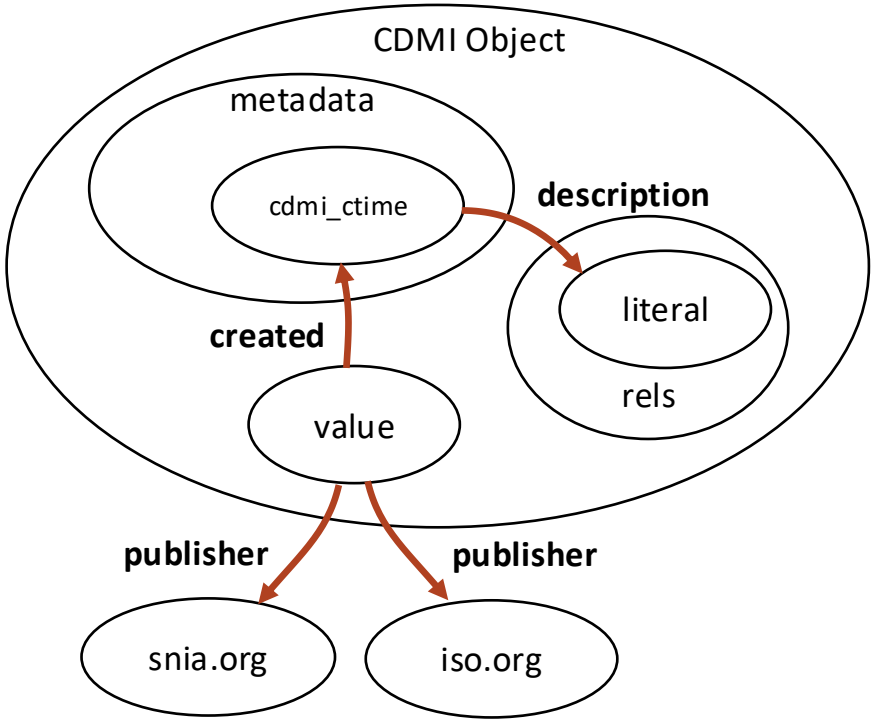
```
{
    "objectType": "application/cdmi-object",
    "objectName": "sunset.jpg",
    "mimetype": "image/jpeg",
    "metadata": {
        "cdmi_size": "2378678",
        "cdmi_data_redundancy": "2",
        "exif_306": "2025-12-30 18:12:19",
        "exif_33432": "Example Inc."
    },
    "rel": {
        "#/metadata/exif_33432": { "http://purl.org/dc/terms/rights": [ {
            "value": "https://www.example.com/copyright" } ]
    }
}
```

Metadata Annotation
(Copyright relationship)

CDMI 3.0 "rel" Metadata Relationships

- Relationships create a graph:

```
{
    "rel": {
        "@context": {
            "dc": "http://purl.org/dc/terms/"
        },
        "#/value": {
            "dc:publisher": [ {
                "value": "https://snia.org/" }, {
                "value": "https://iso.org/" } ],
            "dc:created": [ {
                "value": "#metadata/cdmi_ctime" } ]
        },
        "#/metadata/cdmi_ctime": {
            "dc:description": [ {
                "value": "The creation time of the standard",
                "type": "literal"
            } ]
        }
    }
}
```



SNIA CST A SNIA Community

Self-describing Solutions to Metadata Challenges

Format & Semantics	Add annotations for schemas, origin, and meanings
Encoding, case sensitivity, illegal chars	Add annotations for encoding and restrictions
Size & count limits	Add annotations for limits and validation
Timestamp semantics	Add annotations for formats and sources
Timestamp granularity & timezone	Add annotations for precision and timezones
Version number/id	Add annotations for applications and processing
Ownership & Permissions model	Add annotations for identities and restrictions
Encryption model	Add annotations for encapsulation and processing
Data retention model	Add annotations for management and ownership
Metadata security and access model	Add annotations for authenticity and classification

Come get involved in standardizing CDMI graph relationships, and help define the future of storage metadata management!

Challenge 3 – AI Agent-driven Operation

- The first generation of storage management APIs are operator driven
  - Management APIs provide basic discovery and imperative commands
  - Beginnings of common abstractions and standardization across vendors
- The second generation of storage management APIs are built around orchestration environments and deployment automation (e.g. k8s)
  - Management APIs provide declarative discovery and specify desired state
  - Common abstractions and standardization is framework-driven
- The third generation of storage management APIs are AI agent driven
  - Management APIs provide context and describe possible changes / results
  - AI agents determine how to use these APIs as tools to accomplish specific results on behalf of operators and orchestration environments

Challenge 3 – AI Agent-driven Operation via MCP

- The most widely adopted interface for AI agents to invoke "tools", for example, to perform storage management operations, is known as the Model Context Protocol (MCP)
  - An open standard managed by the Linux Foundation
  - Based on JSON-RPC and JSON Schema
  - Enables AI agents to discover, call and receive responses from external systems

- CDMI already defines a JSON document format that is part of all of the major foundation model's training set
- MCP allows AI agents to use this knowledge to perform storage management operations

Challenge 3 – AI Agent-driven Operation via MCP

- IETF RFC 6208 specifies a set of CDMI JSON representations that are transported over HTTP, including:
  - application/cdmi-object – Operations on a data item
  - application/cdmi-container – Operations on a data namespace
  - application/cdmi-queue – Operations on a data queue

- CDMI defines the contents and meanings of these JSON representations
  - For example, to take a snapshot of a namespace, a JSON document with a body of **{ "snapshot": "my_snapshot_name" }** would be sent to the resource that a snapshot is to be taken

Challenge 3 – AI Agent-driven Operation via MCP

- CDMI maps directly to MCP, as both CDMI and MCP are JSON-based
- The defined CDMI representations map into the MCP “arguments” field, with the method, path and content-type mapping to function, representation and uri fields, respectively.

HTTP-based CDMI operation

```
PATCH /dir/myfiles/ HTTP/1.1
Host: example.com
Content-Type: application/cdmi-container

{
    "snapshot": "MySnapshot"
}
```

MCP-based CDMI operation

```
{
    "jsonrpc": "2.0",
    "id": 4,
    "method": "tools/call",
    "params": {
        "function": "patch",
        "representation": "cdmi-container",
        "uri": "/dir/myfiles/"
        "arguments": {
            "snapshot": "MySnapshot"
        }
    }
}
```


Challenge 3 – AI Agent-driven Operation via MCP

- This approach allows anything that can be done through CDMI can be performed by an MCP client
- For example, an MCP client could request that a new directory be created, or the contents of a directory can be listed:

```
{
    "jsonrpc": "2.0",
    "id": 5,
    "method": "tools/call",
    "params": {
        "function": "put",
        "representation": "cdmi-container",
        "uri": "/dir/myfiles/newdir/",
        "arguments": {
        }
    }
}
```

```
{
    "jsonrpc": "2.0",
    "id": 6,
    "method": "tools/call",
    "params": {
        "function": "get",
        "representation": "cdmi-container",
        "uri": "/dir/myfiles/",
        "arguments": {
        }
    }
}
```


Challenge 3 – AI Agent-driven Operation via MCP

- This extends to CDMI metadata
- For example, the size of a file can be obtained using CDMI + MCP:

```
{
    "jsonrpc": "2.0",
    "id": 7,
    "method": "tools/call",
    "params": {
        "function": "get",
        "representation": "cdmi-object",
        "uri": "/dir/file1?metadata/cdmi_size",
        "arguments": {
        }
    }
}
```

```
{
    "jsonrpc": "2.0",
    "id": 7,
    "result": {
        "function": "get",
        "representation": "cdmi-object",
        "uri": "/dir/file1?metadata/cdmi_size",
        "results": {
            "metadata": {
                "cdmi_size": "23784"
            }
        }
    }
}
```


Challenge 3 – AI Agent-driven Operation via MCP

- This new capability combines with the new relationships functionality to allow AI agents to bulk load in knowledge graphs that span objects:

```
{
    "jsonrpc": "2.0",
    "id": 8,
    "method": "tools/call",
    "params": {
        "function": "get",
        "representation": "cdmi-container",
        "uri": "/dir/?children=![name, rel]",
        "arguments": {
        }
    }
}
```

```
{
    "jsonrpc": "2.0",
    "id": 8,
    "result": {
        "function": "get",
        "representation": "cdmi-object",
        "uri": "/dir/?children=![name, rel]",
        "results": {
            "children": [
                ["file1", { <rel contents> }],
                ["file2", { <rel contents> }],
                ["myfiles/", { <rel contents> }],
                [
                    ["file3", { <rel contents> }],
                ]
            ]
        }
    }
}
```


Challenge 3 – AI Agent-driven Operation via MCP

- Why is this so exciting and powerful? Everything expressible in CDMI can be performed by an agent
- **This means that MCP + CDMI allows for agent-driven discovery, access, management and manipulation of stored cloud data in a standardized way**
- MCP and CDMI are very well aligned:
  - Both CDMI and MCP are JSON-based
  - MCP supports capabilities and tool listing, which correspond to CDMI capabilities
  - MCP supports long-running queries, which correspond to CDMI queues and query queues
  - MCP supports roots, which correspond to CDMI root URIs
  - MCP supports resources, which correspond to CDMI Namespace child listing
- MCP can further enhance CDMI:
  - MCP Prompts allow for a CDMI server to advertise how it can be used
  - MCP tool listing can provide examples of how to use an agent, building on the foundation model's native understanding of CDMI
  - CDMI can enable discovery and management of MCP servers for data in the cloud

Come get involved in standardizing CDMI over MCP,
and help define the future of AI agent-driven storage management!

Getting Involved

- CDMI 3.0 is under development with a target completion at of end of 2026
  - Find more details about CDMI 3.0 plans in the Introduction to CDMI 3.0 Whitepaper
  - Now is a great time to get involved!

- Interested in AI, agents and storage management?
  - Join the Cloud Storage Technical Work Group to help standardize storage management using MCP
- Interested in data access protocol discovery and specification?
  - Join to collaborate on standardizing protocol exports
- Interested in AI cluster accelerated access to data?
  - Join to collaborate on discovery of RDMA and scale-up/scale-out data access methods
- Interested in knowledge graph annotations of data?
  - Join to collaborate on adding graph relationships to CDMI

Q&A

Thanks for Viewing this Webinar

- Please rate this webinar and provide us with feedback
- This webinar and a copy of the slides are available at the SNIA Educational Library
- A Q&A blog from this webinar will be posted at snia.org/blog
- Follow us on LinkedIn and X for future webinar dates

Thank You

