SIRF: Self-contained Information Retention Format

Presenter: Sam Fineberg, HP

Co-Authors: Simona Rabinovici-Cohen, IBM Research – Haifa
Roger Cummings, Antesignanus
Phillip Viana, IBM

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the 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
  - The 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 the SNIA Education Committee.
- 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.

Abstract

SIRF: Self-contained Information Retention Format

Generating and collecting very large data sets is becoming a necessity in many domains that also need to keep that data for long periods. Examples include astronomy, genomics, medical records, photographic archives, video archives, and large-scale e-commerce. While this presents significant opportunities, a key challenge is providing economically scalable storage systems to efficiently store and preserve the data, as well as to enable search, access, and analytics on that data in the far future.

Both cloud and tape technologies are viable alternatives for storage of big data and SNIA supports their standardization. The SNIA Cloud Data Management Interface (CDMI) provides a standardized interface to create, retrieve, update, and delete objects in a cloud. The SNIA Linear Tape File System (LTFS) takes advantage of a new generation of tape hardware to provide efficient access to tape using standard, familiar system tools and interfaces. In addition, the SNIA Self-contained Information Retention Format (SIRF) defines a storage container for long term retention that will enable future applications to interpret stored data regardless of the application that originally produced it.

This presentation describes the challenges in long term retention data, as well as initial work on how to combine SIRF with LTFS and SIRF with CDMI to address some of those challenges. We will also describe an emerging SIRF specification as well as an implementation of SIRF for the cloud.

Outline

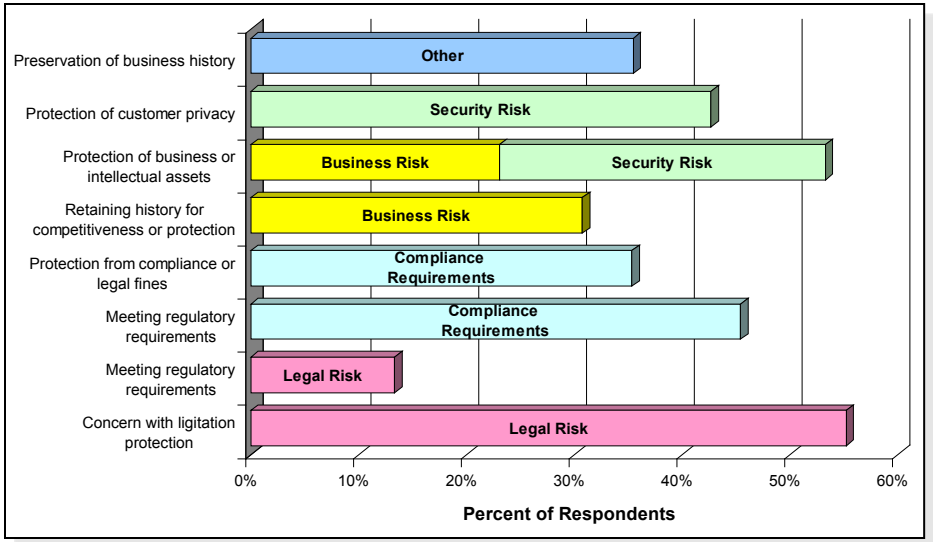
- **Introduction**
- SNIA Long Term Retention technology
  - Self-contained Information Retention Format (SIRF)
  - SIRF metadata catalog schema
- Mapping SIRF to SNIA storage technologies
  - SIRF Serialization for CDMI
  - SIRF Serialization for LTFS
- OpenSIRF: SIRF Implementation for OpenStack Swift
- Summary

The SNIA LTR TWG

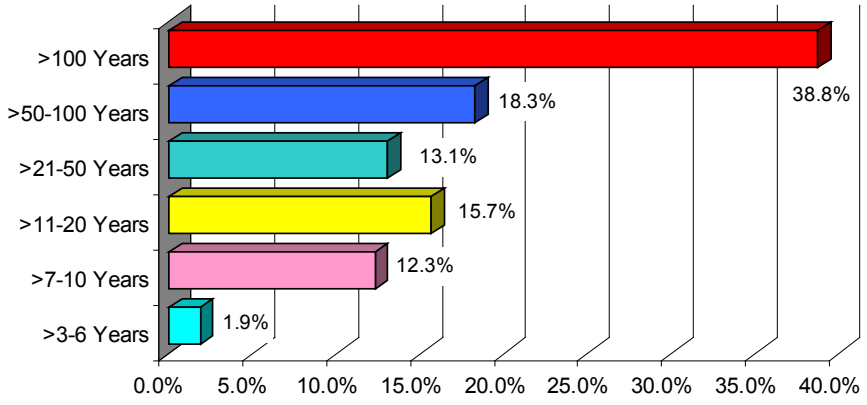
- The Storage Networking Industry Association (SNIA)
  - SNIA is a nonprofit trade association for producers and consumers of networked storage products. SNIA works towards its goals by forming and sponsoring technical work groups, producing conferences, building and maintaining a vendor neutral Technology Center, and by promoting activities that expand the breadth and quality of the storage market.
- The Long Term Retention (LTR) TWG
  - Created to lead storage industry collaboration with groups concerned with, and develop technologies, models, educational materials and practices related to, data & information retention & preservation.

SNIA Survey from 2007

Top External Factors Driving Long-Term Retention Requirements:
Legal Risk, Compliance Regulations,
Business Risk, Security Risk



Source: SNIA-100 Year Archive Requirements Survey, January 2007.



What does Long-Term Mean?
Retention of 20 years or more is required by 70% of responses.

The Need for Digital Preservation

- Regulatory compliance and legal issues
  - Sarbanes-Oxley, HIPAA, FRCP, intellectual property litigation
- Emerging web services and applications
  - Email, photo sharing, web site archives, social networks, blogs
- Many other fixed-content repositories
  - Scientific data, intelligence, libraries, movies, music

Scientific and Cultural

Satellite data is kept for ever

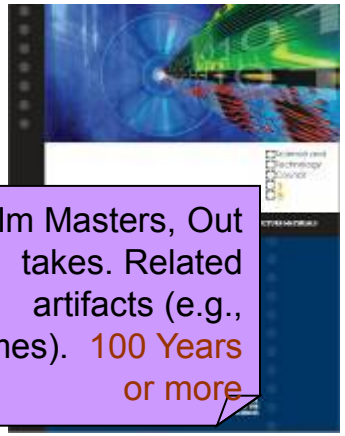
We would like to keep digital art for ever

Healthcare

X-rays are often stored for periods of 75 years

Records of minors are needed until 20 to 43 years of age

M&E



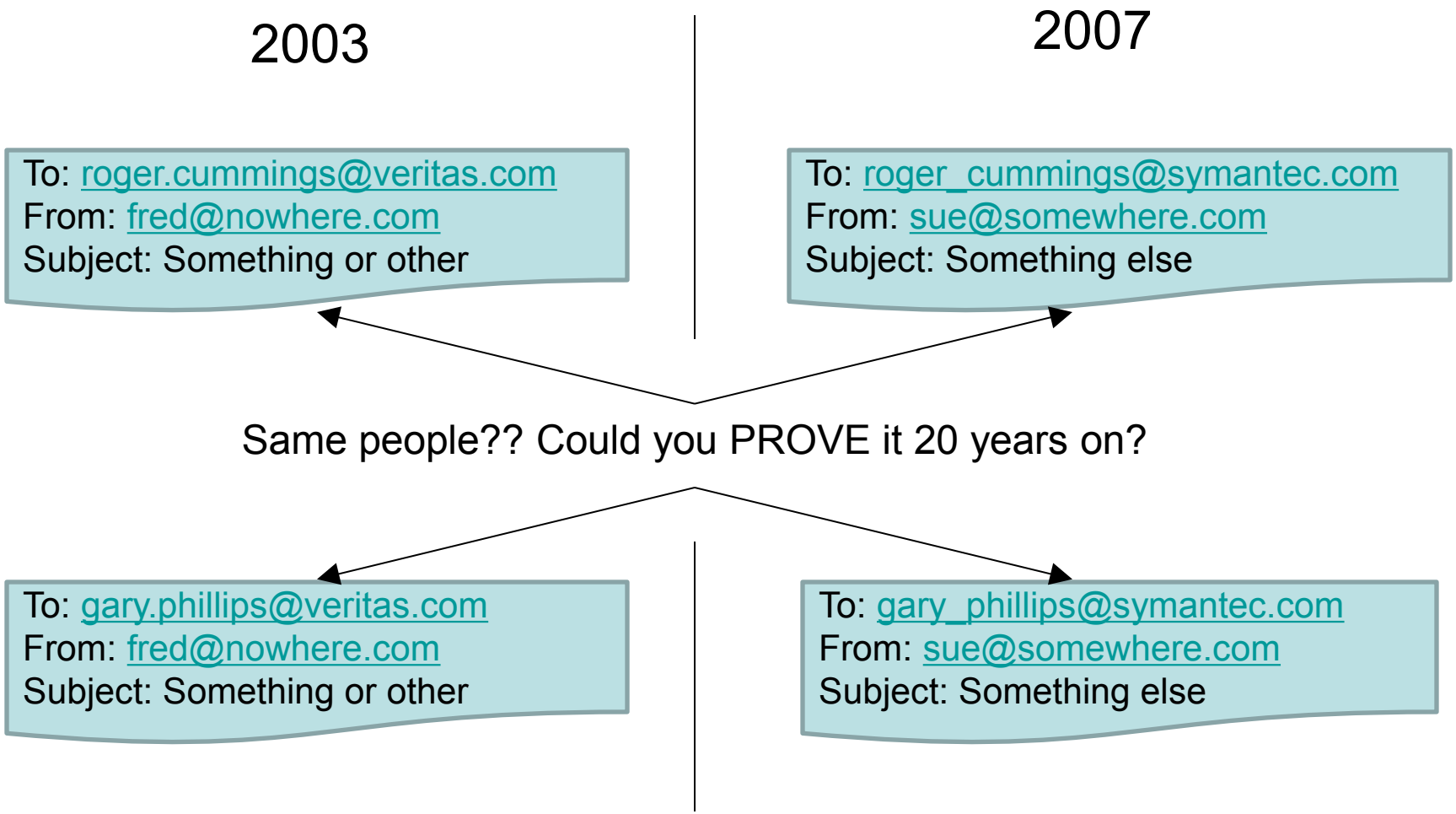
Film Masters, Out takes. Related artifacts (e.g., games). 100 Years or more

SIRF: Self-contained Information Retention Format

Goals of Digital Preservation

- Digital assets stored now should remain
  - Accessible
  - Undamaged
  - Usable
- For as long as desired – beyond the lifetime of
  - Any particular storage system
  - Any particular storage technology
- And at an *affordable cost*
  - "affordable" depends on the perceived future value of information
  - Good preservation practice can bring down the cost – potentially allowing more information to be preserved

Real Life Example Semantic Problem



Outline

- Introduction
- **SNIA Long Term Retention technology**
  - Self-contained Information Retention Format (SIRF)
  - SIRF metadata catalog schema
- Mapping SIRF to SNIA storage technologies
  - SIRF Serialization for CDMI
  - SIRF Serialization for LTFS
- OpenSIRF: SIRF Implementation for OpenStack Swift
- Summary

SIRF: Self-contained Information Retention Format

Being developed by SNIA Long Term Retention (LTR) TWG

An Analogy

- Standard physical archival box
  - Archivists gather together a group of related items and place them in a physical box container
  - The box is labeled with information about its content e.g., name and reference number, date, contents description, destroy date
- SIRF is the digital equivalent
  - Logical container for a set of (digital) preservation objects and a catalog
  - The SIRF catalog contains metadata related to the entire contents of the container as well as to the individual objects
  - SIRF standardizes information in the catalog, and enables data and metadata to be maintained

Photo courtesy Oregon State Archives

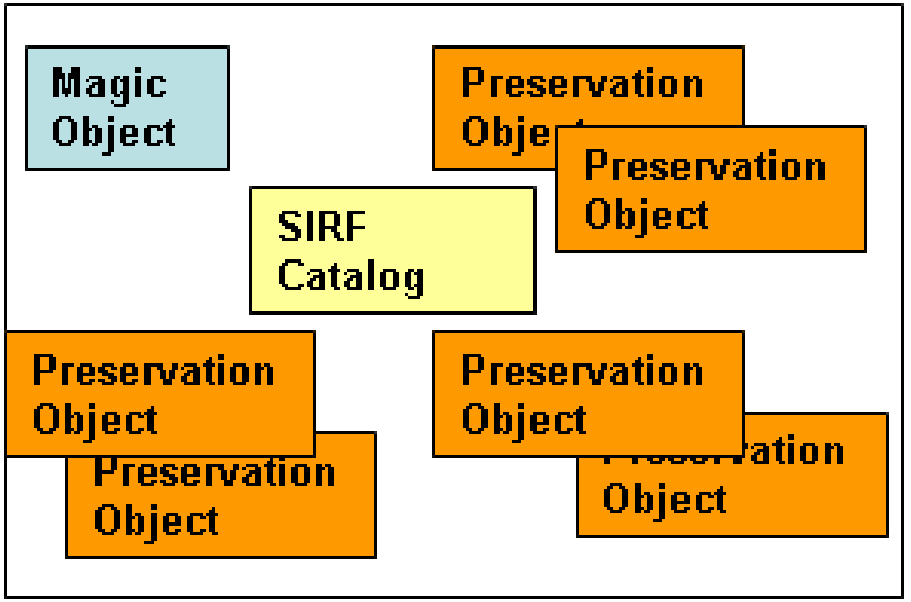
SIRF Properties

- SIRF is a logical data format of a storage container appropriate for long term storage of digital information
  - A storage container may comprise a logical or physical storage area considered as a unit.
  - Examples: a file system, a tape, a block device, a stream device, an object store, a data bucket in a cloud storage
- Required Properties
  - Self-describing – can be interpreted by different systems
  - Self-contained – all data needed for the interpretation is in the container
  - Extensible – so it can meet future needs

SIRF Components

A SIRF container includes:

- **Magic object**: identifies SIRF container and its version
- **Preservation objects** that are immutable
- **Catalog** that is
  - Updatable
  - Contains metadata to make container and preservation objects portable into the future without external functions
  - Includes relational and semantic metadata to preserve the preservation objects' meaning



Catalog Metadata

- Catalog schema is fixed
  - Catalog contains basic information
  - Self describing
- Metadata may vary based on the preservation needs
  - Different domains require different types of metadata
  - Utilizes domain specific formats and ontologies
  - Self contained, requiring minimal external information
- Additional levels of SIRF metadata may exist
  - Enabling deeper description of preservation objects
  - Preserving content meaning and relationships between objects

SIRF Catalog Metadata Schema

1 Container Information
- 1.1 Specification
- 1.2 Container ID
- 1.3 State
- 1.4 Provenance
- 1.5 Audit log

2 Objects Set

3 Object Information
- 3.1 Object IDs
- 3.2 Related objects
- 3.3 Dates
- 3.4 Packaging format
- 3.5 Fixity
- 3.6 Retention
- 3.7 Audit log
- 3.8 Extension

Outline

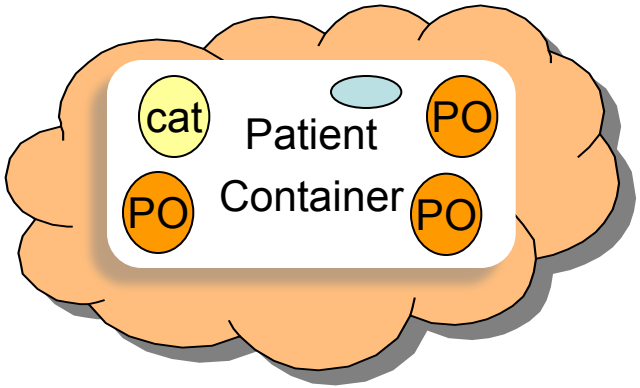
- Introduction
- SNIA Long Term Retention technology
  - Self-contained Information Retention Format (SIRF)
  - SIRF metadata catalog schema
- **Mapping SIRF to SNIA storage technologies**
  - SIRF Serialization for CDMI
  - SIRF Serialization for LTFS
- OpenSIRF: SIRF Implementation for OpenStack Swift
- Summary

Cloud Data Management Interface (CDMI)

- An ISO/IEC 17826:2012 Information technology standard being developed by SNIA CDMI TWG
- The CDMI standard defines an interoperable format for moving data and associated metadata between cloud providers
- CDMI data objects can be accessed by standard browsers and internet tools (subject to owner's access control lists)
- CDMI data objects may "order" data services from the cloud
  - Secure Erasure, Encryption, Replication, Retention, Backup/Restore, Tiering, Hashing, Preservation, etc. (extensible)
  - Done through Data System Metadata (key/value) on the Containers or Objects
- Has several implementations including in OpenStack

SIRF Serialization for CDMI: Interface

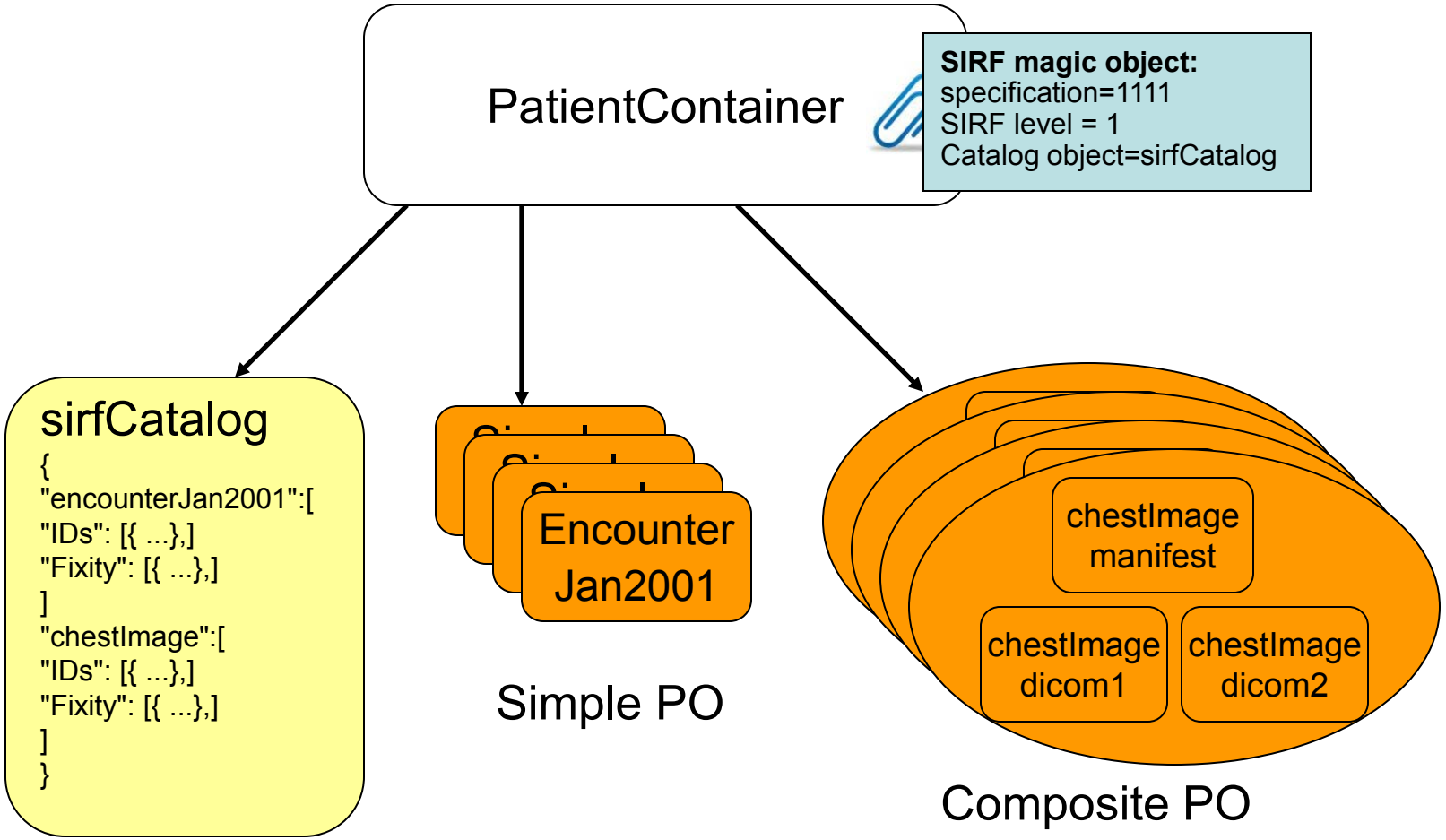
- CDMI API can be used to access the various preservation objects and the catalog object in a SIRF-compliant CDMI container
- Example
  - Assume we have a cloud container named "PatientContainer" that is SIRF-compliant
    - the container has a catalog object
    - each encounter is a preservation object
    - each image is a preservation object



  - We can read the various preservation objects and the catalog object via CDMI REST API as follows:

    GET <root URI>/ PatientContainer>/sirfCatalog
    GET <root URI>/<PatientContainer>/encounterJan2001
    GET <root URI>/<PatientContainer>/chestImage

SIRF Serialization for CDMI



SIRF Serialization for CDMI: General

A CDMI Container can be qualified also as a SIRF Container when:

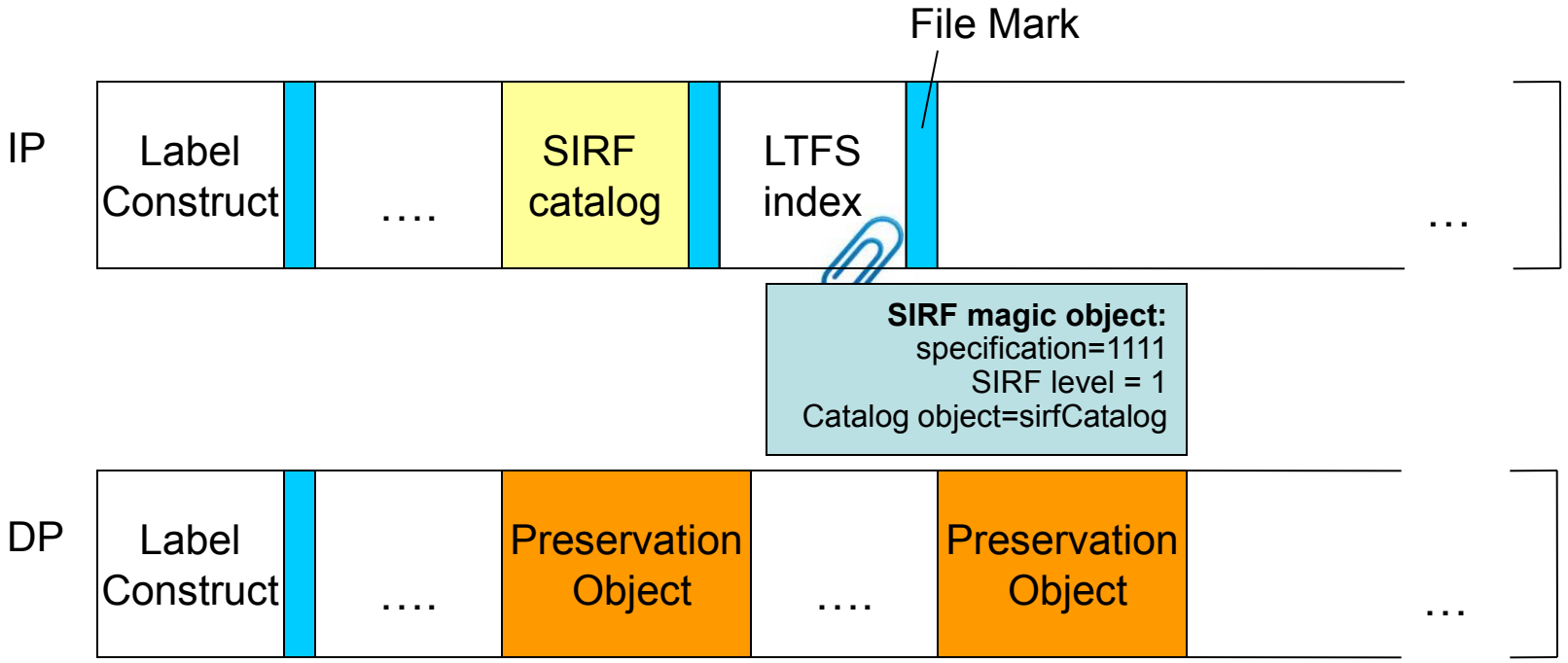
- The SIRF magic object is mapped to the CDMI container metadata and includes, for example, specification ID and version, SIRF level, SIRF catalog object ID.
- The SIRF catalog is an object in the CDMI container formatted in JSON
- A SIRF preservation object (PO) that is a simple object (contains one element) is mapped to a CDMI data object
  - The simple object can be a tar/zip
- A SIRF PO that is a composite object (contains several elements) is mapped to:
  - a set of data objects (one for each element) and a manifest data object that its content includes the IDs and fixities of the element data objects

Linear Tape File System (LTFS)

- A file system implemented on dual-partition linear tape:
  - Index Partition and Data Partition
    - Index Partition is "small" (2 wraps, 37.5 GB out of 1.5 TB on LTO5)
    - Data Partition is remainder of the tape
  - File System module that implements a set of standard file system interfaces
    - Implemented using FUSE
    - On Linux and Mac OS X
    - Windows implementation uses FUSE-like framework
  - Includes an on-tape structure used to track tape contents
    - XML Index Schema
- Format becoming the standard for linear tape
  - Formal standardization through SNIA LTFS TWG

SIRF Serialization for LTFS

- The index partition of the tape is 2 wraps which is 37.5 GB in LTO-5 and probably larger in LTO-6.
  - The tape index partition is large enough to hold the LTFS index, the SIRF catalog, and even additional information e.g. thumbnails of images



SIRF Serialization for LTFS: General

- A LTFS Tape can also be a SIRF Container when:
  - The SIRF magic object is mapped to extended attributes of the "LTFS index" root directory
    - The magic object includes, for example, specification ID and version, SIRF level, reference to SIRF catalog
  - The SIRF catalog resides in the index partition and formatted in XML
  - A SIRF preservation object (PO) that is a simple object (contains one element) is mapped to a LTFS file
  - A SIRF PO that is a composite object (contains several elements) is mapped to:
    - a set of LTFS files (one for each element) and a manifest file that its content includes the IDs and fixities of the element data objects

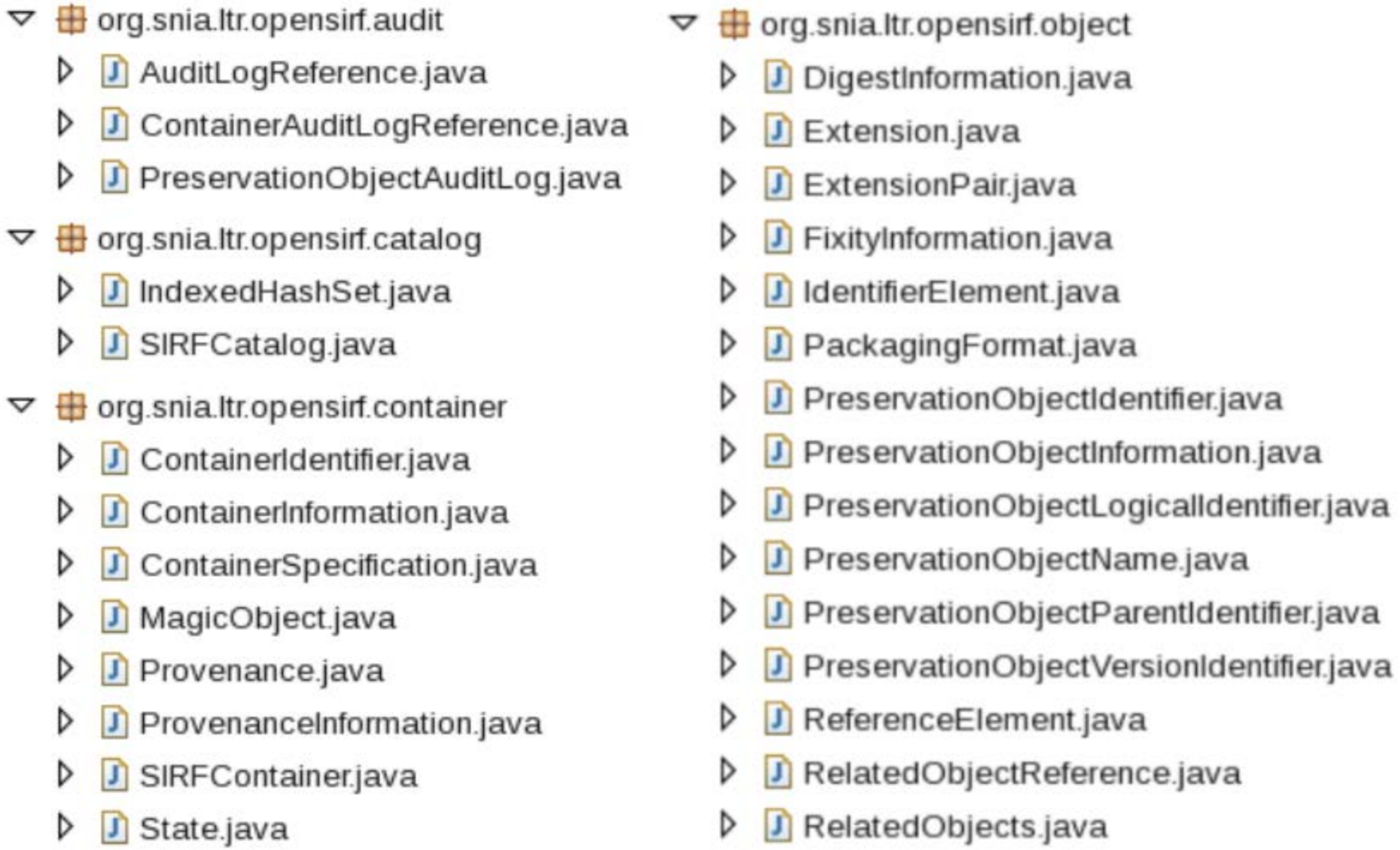
Outline

- Introduction
- SNIA Long Term Retention technology
  - Self-contained Information Retention Format (SIRF)
  - SIRF metadata catalog schema
- Mapping SIRF to SNIA storage technologies
  - SIRF Serialization for CDMI
  - SIRF Serialization for LTFS
- **OpenSIRF: SIRF Implementation for OpenStack Swift**
- Summary

OpenSIRF: SIRF Implementation for OpenStack Swift

- Open source Java implementation of SIRF
- Supports OpenStack Swift as underlying storage
- **OpenSIRF Core (opensirf-core):** contains the model classes for the SIRF container, magic object, catalog, POs and all categories
  - Classes that are part of the catalog contain JAXB annotations for marshalling to/unmarshalling from JSON and XML
- **OpenSIRF REST (opensirf-jaxrs):** RESTful layer using JAX-RS, supporting marshalling and unmarshalling of OpenSIRF Core objects to XML and JSON when the HTTP methods are called

OpenSIRF Core Classes

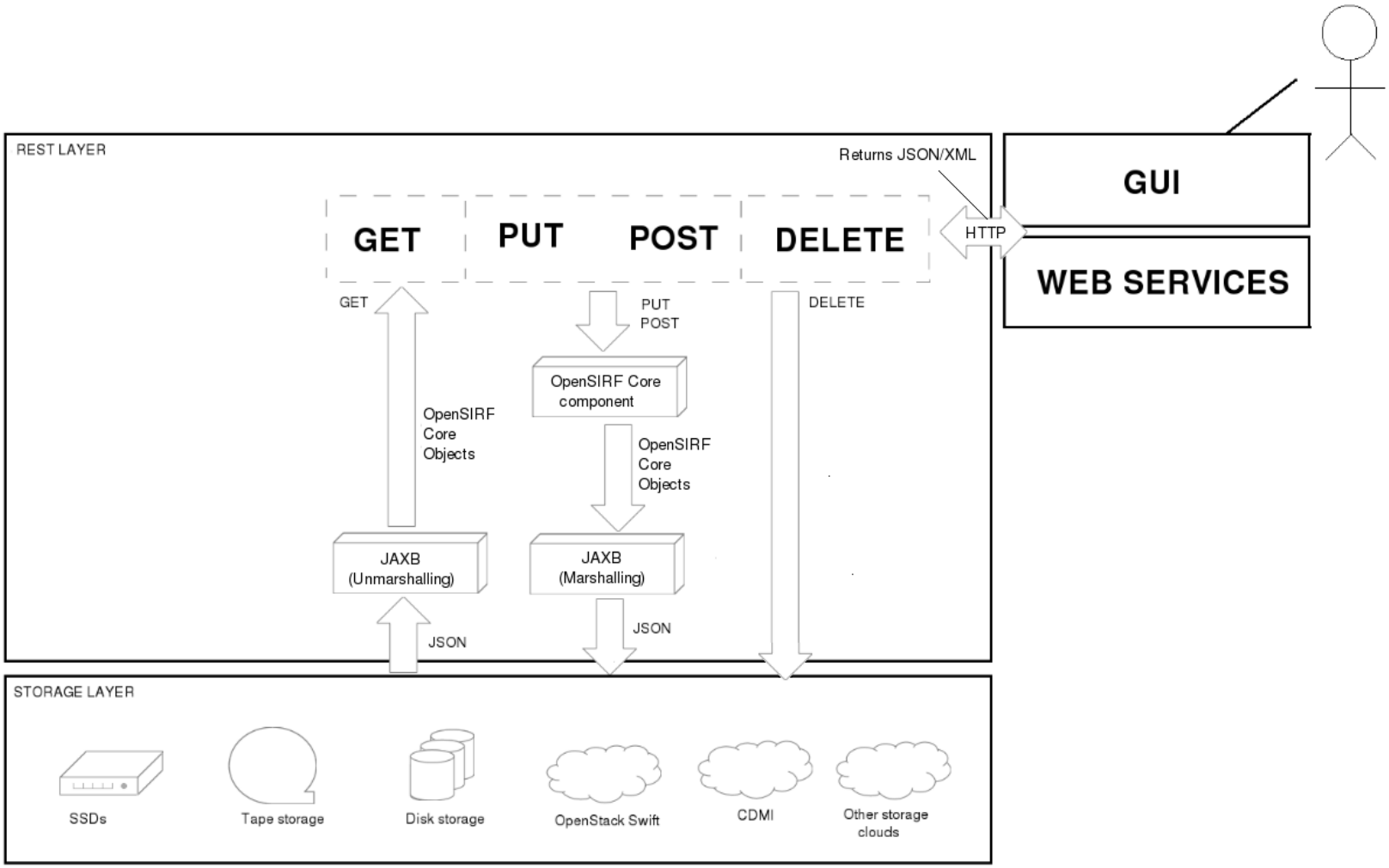


SIRF: Self-contained Information Retention Format

OpenSIRF REST API

URI ↓ / OPERATION →	GET	POST	PUT	DELETE
/sirf/container	List all SIRF containers	–	–	–
/sirf/container/{name}	Get container's magic object	–	Create container (+ MO + catalog)	Delete all container objects
/sirf/container/{name}/catalog	Get container's catalog	–	Update container's catalog	–
/sirf/container/{name}/{po}	Get preservation object metadata	Create or update preservation object	Update preservation object information	Delete preservation object metadata
/sirf/container/{name}/{po}/data	Get PO contents (without metadata)	–	–	Delete preservation object contents and metadata

OpenSIRF Architecture



SIRF: Self-contained Information Retention Format

Outline

- Introduction
- SNIA Long Term Retention technology
  - Self-contained Information Retention Format (SIRF)
  - SIRF metadata catalog schema
- Mapping SIRF to SNIA storage technologies
  - SIRF Serialization for CDMI
  - SIRF Serialization for LTFS
- OpenSIRF: SIRF Implementation for OpenStack Swift
- **Summary**

Summary

- Need to retain not only information of interest but ALL other information to make it fully usable in future
  - Put it all in the SIRF “digital box”, preserve that as a unit
  - SIRF includes metadata about the storage container, to help “understand” the contents of the container in the future
- No single technology will be usable over the timespans mandated by current digital preservation needs
  - SNIA CDMI and LTFS technologies are among best current choices
    - Are good for perhaps 10-20 years
  - SIRF provides a vehicle for collecting all of the information that will be needed to transition to new technologies in the future
    - SIRF can be serialized for the future technologies as they come
- OpenSIRF is an open source SIRF implementation for OpenStack Swift

For further information

- SIRF Specification for public review (coming soon)
  - http://www.snia.org/tech_activities/publicreview
- OpenSIRF: SIRF Implementation for OpenStack Swift
  - http://200.144.254.4:4061/opensirf-jsp
  - http://200.144.254.4:4061/opensirf-core-javadoc
    - User: lttwg
    - Password: 100years
- SIRF use cases and requirements document is released for public review
  - http://www.snia.org/tech_activities/publicreview
- More information on SIRF (& other SNIA LTR activities) is available at
  - http://www.snia.org/ltr

Attribution & Feedback

The SNIA Education Committee thanks the following Individuals for their contributions to this Tutorial.

Authorship History

Authors (Winter 2015)

Simona Rabinovici-Cohen
Sam Fineberg
Phillip Viana
Roger Cummings
Mary Baker

(incorporating materials from earlier tutorials dating back to 2008, and with particular thanks to the 100 Year Archive Task Force (2007))

Additional Contributors

Gene Nagle
Mark Carlson (& the Cloud TWG)
Eric Hibbard
David Pease
Joseph White
Alan Yoder

Please send any questions or comments regarding this SNIA Tutorial to ***tracktutorials@snia.org***