Long Term Information Retention

Sam Fineberg (HP Software)
Simona Rabinovici-Cohen (IBM)
With lots of help from other members of the SNIA LTR TWG including Mary Baker, Roger Cummings, John Marberg, Gene Nagle, Michael Peterson, Don Post and Bob Rogers

Outline

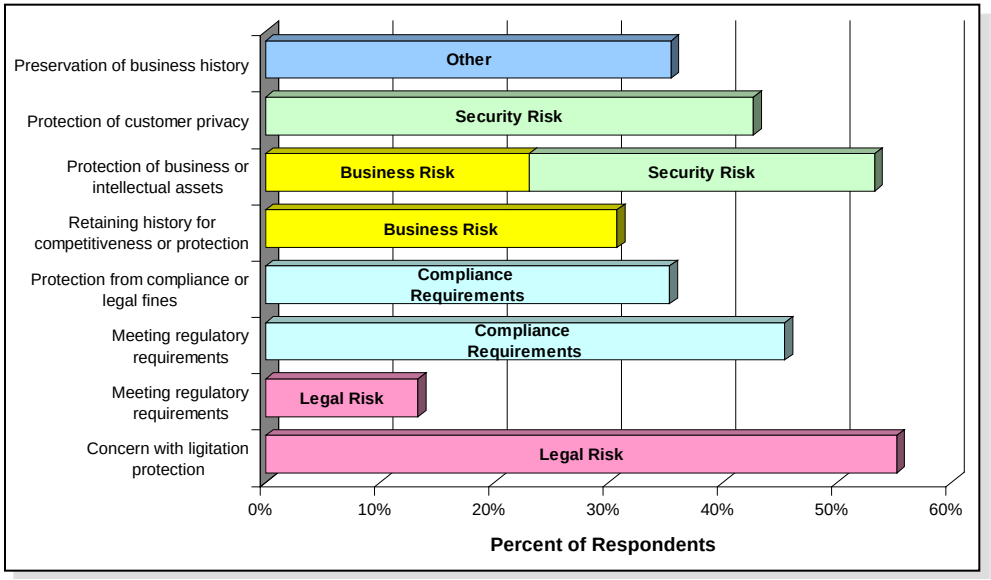
- Introduction to digital preservation
- Preservation Technologies
- SNIA SIRF: Self-contained Information Retention Format
- EU ENSURE: Enabling kNowledge, Sustainability, Usability and Recovery for Economic Value
- Summary

Avoiding the Digital Dark Age

- More and more critical information is created digitally and never sees paper
  - Documents, Web Pages, videos, music, photos, …
  - Medical data, business records, historical documents, …
- We have known for years that digital information is easy to lose
  - But preservation is hard, expensive, and poorly defined
  - Governments and libraries are just starting to grapple with the problem, businesses have largely ignored it
- As a consequence, most businesses and individuals are in danger of losing information, and may not even know it is happening
- **We are at risk of losing decades of digital content before we ever get around to preserving it.**

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.

Key findings

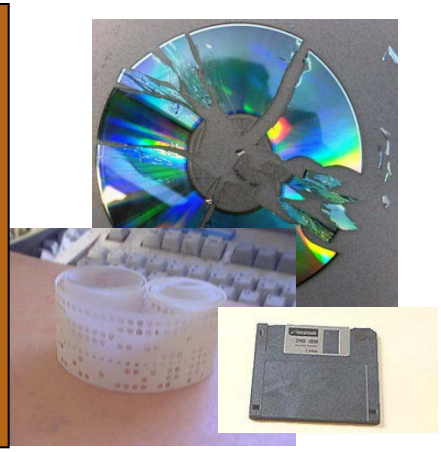
- 68% had to retain data more than 100 years
- 83% had to retain data more than 50 years
- Less than 20% were satisfied that they could access their retained data more than 50 years in the future

100 YEAR Archive Task Force

Threats to long-term assets

- Large-scale disaster
- Human error
- Media faults

Long-term content suffers from more threats than short-term content



- Component faults
- Economic faults
- Attack
- Organizational faults

- Media/hardware obsolescence
- Software/format obsolescence
- Lost context/metadata

Even preserving the bits is hard

- Large scale & long time periods
  - Extremely unlikely that a single copy of a large corpus can be completely error free
  - Even improbable events will have an effect
- Now try to keep
  - The bits usable - physical preservation
  - The information reusable - logical preservation

Practices vary by time

- Can't predict what will change – only know it will
- This means processes are key
  - Must be evolvable
    - Current processes get us to the next step
    - At that point we will likely need new processes to take over
  - Must not destroy what we are trying to protect
  - Standards make evolution easier
- A good archive is almost always in motion
  - Digital preservation is not a static activity!

Practices vary by context

- What do we preserve?
  - Bits? Applications? Logical connections? Context? Etc.?
  - Depends on customer domain
    - Example: digital copy of old book
      - words? wear and tear on the paper? political context?
  - Can't always predict the eventual use
    - Affordability may force some decisions
- What do we use?
  - Techniques
    - Virtual machines? Emulation? Canonical formats?
    - Self-describing formats? Standardized data models?
    - Loss-tolerant formats? Format migration?
    - Preservation of ancient equipment?
  - Yes: all could play a role for different domains

SNIA's effort to address preservation

- Formation of the Long Term Retention (LTR) TWG
- Goals of digital preservation
  - Digital assets stored now should remain accessible, usable, undamaged
  - For as long as desired – beyond the lifetime of any particular storage system & any particular storage technology (or any application!!)
  - And at an affordable cost (or a range of cost/performance)
- LTR TWG Program of Work addresses both "bit preservation" and "logical preservation"
  - Both are absolutely necessary to retain usability of information
  - Cannot make either reliable enough by itself @ reasonable cost
  - **Migration** is a potentially affordable approach for both

Migration

- Move a set of information from an old device or technology growing less reliable (e.g. LTO-2 tape) ……
- … or from an application no longer supported or in general use (e.g. WordPerfect 4.2)…..
- to a new device and/or a new format
- Requirements for migration
  - Preserve not only all the data but all related metadata too
  - Maintain provenance, authenticity & integrity
  - Be auditable and traceable
- Need a "container" to encapsulate all of the related information … and a way to automate much of migration

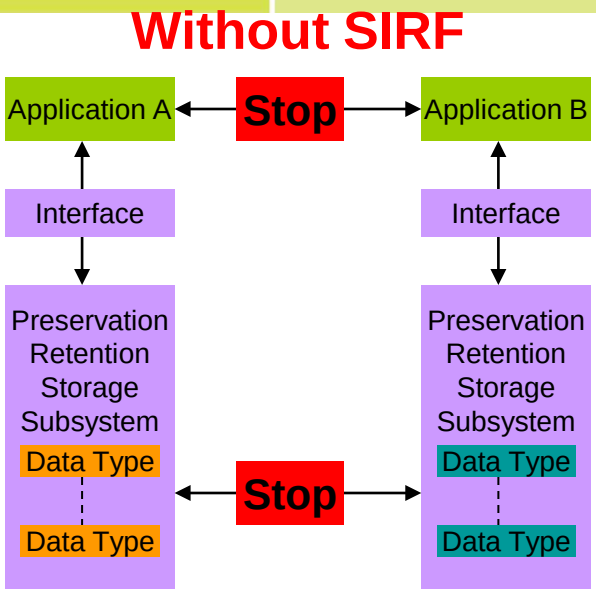
An Analogy

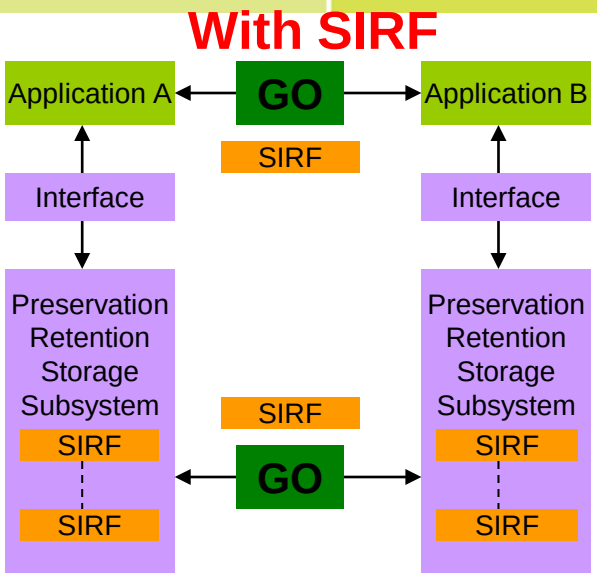
- Standard archival box
  - Archivists gather together a group of related items, known as a collection
  - Collection is placed in a physical box container
  - The box is labeled with information about its content e.g., name and reference number, date, contents description, destroy date

Photo courtesy Oregon State Archives

- 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 the information in the catalog

Problems SIRF addresses





Sets of linked objects moved individually; referential integrity and context may be lost	Sets of linked objects moved between systems maintaining referential integrity and full context
Only original application that created the objects can read and interpret them	Any SIRF compliant application can read and interpret the objects
Export and import needed to migrate objects	Objects migrated without export and import
Preservation Objects cannot be sustained long-term	Preservation Objects can survive longer

Outline

- Introduction to digital preservation
- Preservation Technologies
- SNIA SIRF: Self-contained Information Retention Format
- EU ENSURE: Enabling kNowledge, Sustainability, Usability and Recovery for Economic Value
- Summary

Self-contained Information Retention Format (SIRF)

- SIRF is a logical container format appropriate for long-term storage of digital information
  - Preserves collections of objects and their relationships
  - Includes generic metadata that can be extended with domain specific information
  - Can be mapped to and physically migrated between a wide variety of underlying storage systems
- SIRF use cases and requirements document is released for public review
  - http://www.snia.org/tech_activities/publicreview
- More information on SIRF is available at
  - ***http://www.snia.org/ltr***

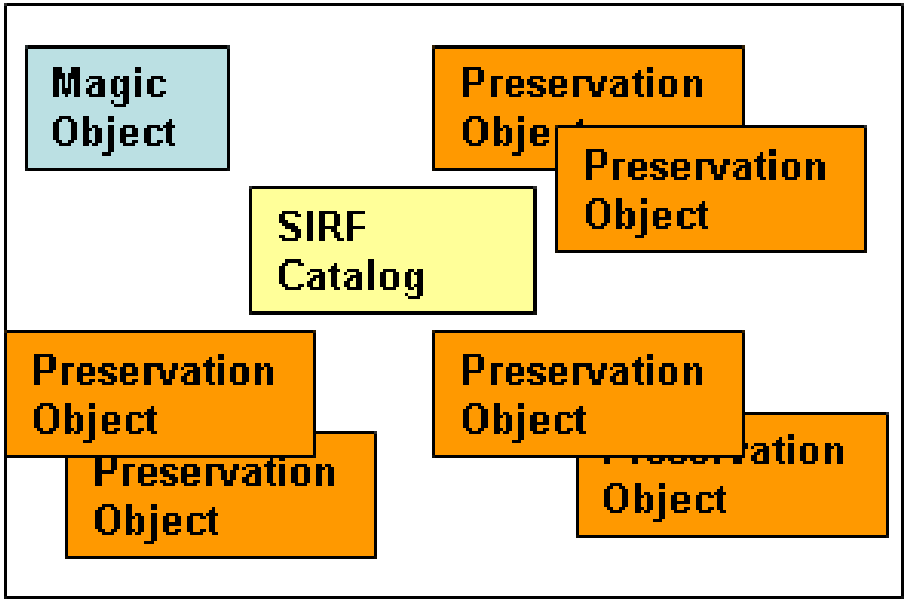
What is SIRF?

- SIRF is a logical data format of a storage container.
  - A storage container may comprise a logical or physical storage area considered as a unit.
    - For example, a storage container may comprise a mountable data storage unit, a file system, a tape, a block device, a stream device, an object store, a data bucket in a cloud storage
  - A SIRF container holds a set of preservation objects to be understood in future
- Required Properties
  - Self-describing – can be interpreted by different systems
  - Self-contained – all data needed for the preservation objects interpretation is in the container
  - Extensible – so it can meet future needs

SIRF Components

A SIRF container includes:

- A **magic object**: identifies SIRF container and its version
- Numerous **preservation objects** that are immutable
- A **catalog** that is
  - Updatable
  - Contains metadata to make container and preservation objects portable into the future without external functions



Layered Approach

- A SIRF container follows a well defined logical data format
  - The format should be understandable by any SIRF compatible application
  - Maps to multiple underlying object interface layers
  - Example object layers
    - Advanced: OSD, Cloud, XAM
    - Lower level: UDF, CDFS, FAT, LTFS

- SIRF metadata is defined at two levels
  - Level 1 catalog (L1) – unique metadata, not in the preservation objects, that is mandatory to make preservation objects portable into the future
  - Level 2 catalog (L2) – information that is probably also in the preservation objects, that is needed for fast access to the preservation objects

What is a Preservation Object?

- SIRF Containers Store Collections of Preservation Objects (POs)

- A Preservation Object is
  - the raw data to be preserved,
  - plus additional embedded or linked metadata, and
  - includes everything needed to enable the sustainability of the information encoded in the raw data for decades to come

- Attributes of a PO
  - may be subject to physical and logical migrations
  - may be dynamic and change over time
  - an updated PO is a new version of the original, and its audit log records the changes that have occurred so authenticity may be verified

- An example of a PO is OAIS Archival Information Package (AIP)
  - An AIP includes recursive representation information that enables future interpretation of the raw data

SIRF's Relation to Existing Preservation Standards

- Generic formats
  - Bagit
  - JHOVE
- Domain specific packaging formats
  - XML Formatted Data Unit (XFDU)
  - VERS Encapsulated Object (VEO)
  - Metadata Encoding and Transmission Standard (METS)
  - Preservation metadata: Implementation Strategies (PREMIS)
- SIRF can be used with many of these, but it is unique because it
  - Preserves collections of objects and their relationships
  - Includes generic metadata that can be extended with domain specific information for fast access
  - Can be mapped to and physically migrated between a wide variety of underlying storage systems

SIRF and CDMI

- Cloud Data Management Interface (CDMI) specifies a standard API for clouds
- CDMI API can be used to access the various preservation objects and the catalog object in a SIRF-compliant container
- Example
  - Assume we have a cloud container named "Patient X" that is SIRF-compliant. This means, the container has several medical records of this patient where each medical record is a preservation object. Additionally, the container has a catalog object.
  - We can read the various medical records (preservation objects) and the catalog object via CDMI REST API as follows:

    GET <root URI>/<ContainerName>/<DataObjectName>

    GET <root URI>/Patient X/catalog

    GET <root URI>/Patient X/MedicalRecord1

Specification Methodology

- ✓ Define use cases and flows among the actors involved in SIRF
  - 4 generic uses cases and 5 Workload-based use cases
- ✓ For each use case, find the derived functional requirements

- Define the semantic metadata items in the SIRF catalog via a hierarchical numbered list representation
  - Add cardinally to each metadata item
- Define a detailed table for each metadata item
  - Include definition, rationale, obligation, usage notes, etc.
- Perform iteratively until convergence
  - Validate the already specified metadata items against a use case
  - Update metadata items

- Inspired by the PREMIS specification - http://www.loc.gov/standards/premis/v2/premis-2-1.pdf

SIRF Level 1

- The SIRF catalog includes metadata such as:
  - General information:
    - Spec ID and version
    - Provenance of container e.g. cloud container name
    - Audit Object ID
  - For each Preservation Object:
    - Preservation object ID (unique over offline/online storage)
    - Preservation object copy number/ID ?
    - Preservation object children's ID
    - Create date using auditable time stamp
    - Modify date? (maybe can be deduced from level 0)
    - Last accessed? (maybe can be deduced from level 0)
    - Last fixity check date
    - Fixity algorithm (may be multiple)
    - Fixity value (may be multiple)
    - Retention/Litigation hold reference count
    - Retention/Litigation hold date ?
    - Deletion hold reference count
    - Retention date
    - Audit object ID
    - Extension

Hierarchical Representation Example

1 containerInformation (1-1: M, NR)
  1.1 sirfSpecification (1-1: M, NR)
    1.1.1 sirfLevel (1-1: M, NR)
    1.1.2 sirfIdentifier (1-1: M, NR)
  1.2 containerIdentifier (0-*: O, R)
    1.2.1 containerIdentifierType (1-1: M, NR)
    1.2.2 containerIdentifierValue (1-1: M, NR)
  …
2 objectInformation (1-*: M, R)
  2.1 objectIdentifier (1-*: M, R)
    2.1.1 objectIdentifierType (1-1: M, NR)
    2.1.2 objectIdentifierValue (1-1: M, NR)
  …
  2.7 objectFixity (0-*: O, R)
    2.7.1 objectDigestAlgorithm (1-1: M, NR)
    2.7.2 objectDigest (1-1: M, NR)
    2.7.3 objectDigestOriginator (0-1: O, NR)
  …

objectIdentifier Table

Item	2.1 objectIdentifier
Components	2.1.1 objectIdentifierType 2.1.2 objectIdentifierValue
Definition	A designation used to uniquely identify the object within the container in which it is stored.
Rationale	
Repeatability	Repeatable
Obligation	Mandatory
Creation / Maintenance notes	
Usage notes	Identifiers must be unique within the container. They may be preexisting (?), and in use in other digital object management systems.

objectIdentifierType Table

Item	2.1.1 objectIdentifierType
Components	None
Definition	A designation of the domain within which the object identifier is unique.
Rationale	Identifier values cannot be assumed to be unique across domains; the combination of *objectIdentifierType* and *objectIdentifierValue* should ensure uniqueness.
Repeatability	Not Repeatable
Obligation	Mandatory
Examples	DLC, DRS, hdl:4263537
Usage notes	The type of the identifier may be implicit within the container as long it is can be explicitly communicated when the digital object is disseminated outside of it.

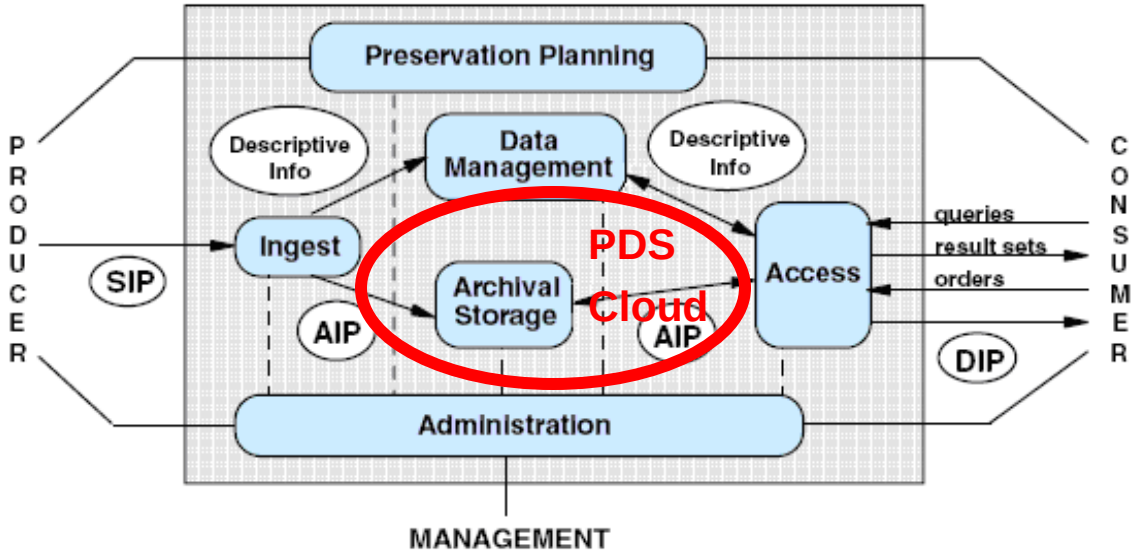
Outline

- Introduction to digital preservation
- Preservation Technologies
- SNIA SIRF: Self-contained Information Retention Format
- EU ENSURE: Enabling kNowledge, Sustainability, Usability and Recovery for Economic Value
- Summary

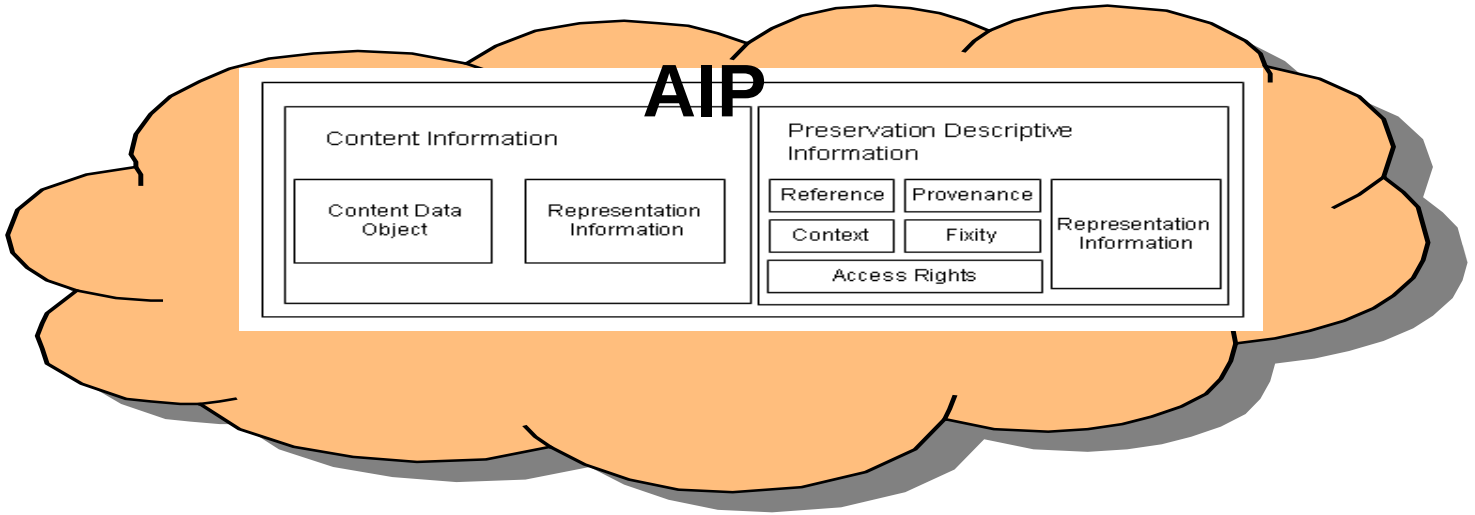
- ENSURE is FP7 EU Project in the area of preservation
  - Three year Integrated Project (IP) started Feb. 1, 2011
  - Consortium of 13 partners (industry and academic)
- ENSURE has a business/industry-oriented focus
  - Drivers for preservation are both regulatory and business value
  - Past efforts on digital preservation have focused on memory institutions addressing the good of society
- Demonstrated with three use case: Health Care, Clinical Trials and Finance
  - Create a financially viable preservation solution
  - Handle regulation and lifecycle management
  - Expand use of emerging technologies for digital preservation
  - Maintain access and privacy rights to information and IP

PDS Cloud: Preservation DataStores in the Cloud

- Provides preservation-aware storage services for ENSURE
- Based on OAIS Archival Storage entity but provides more automation of preservation processes
- Built on top of multiple clouds concurrently, while taking advantage of each one's special capabilities
- Includes a SIRF Handler component for future implementation

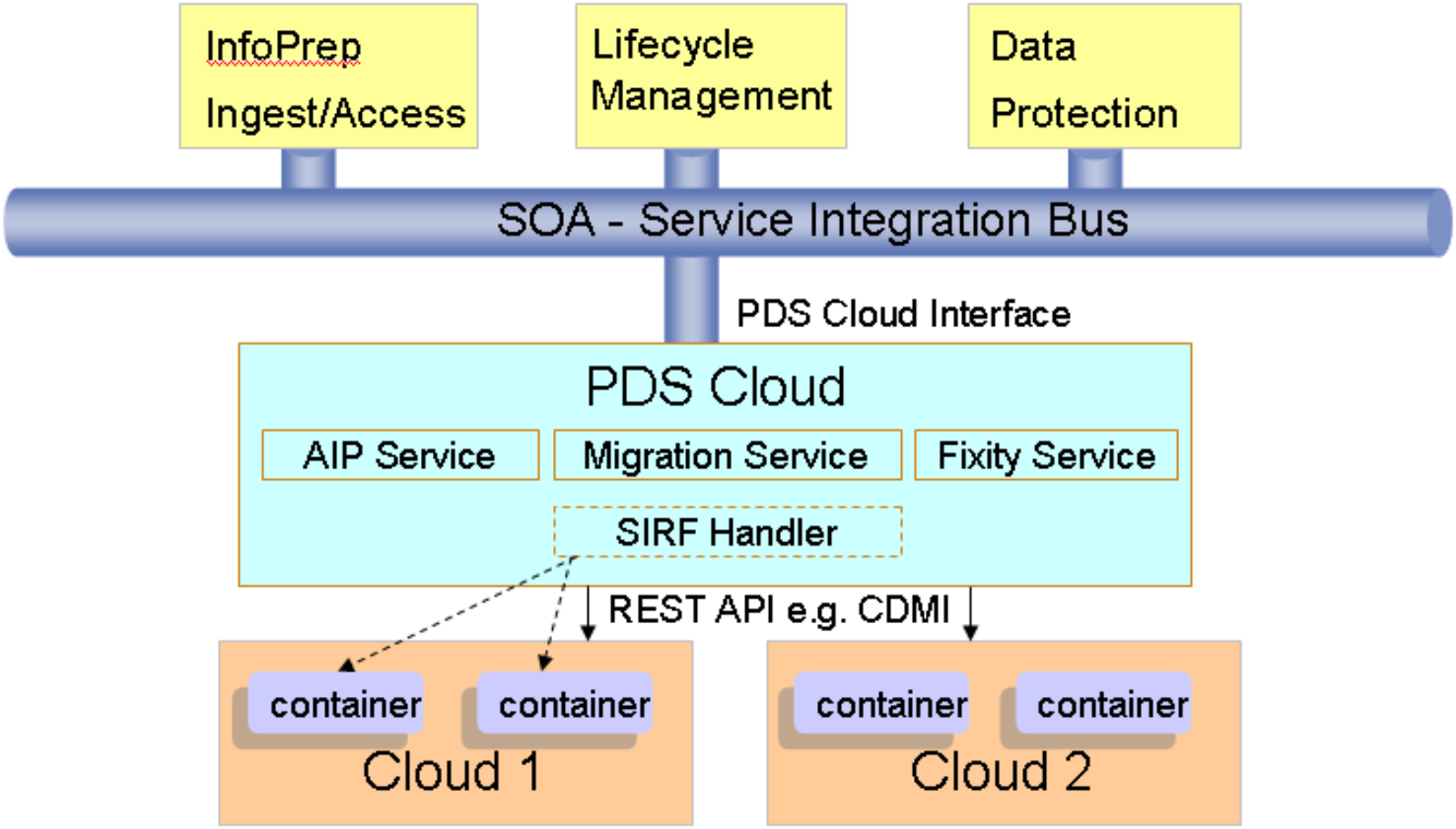


PDS Cloud Functionality



- Map OAIS AIP and the links among AIPs to the cloud data model
- Multi cloud support while considering self-containment and self-describing implications
- Migrate data in its entirety including all its metadata, provenance, context, representation information
- Support multiple integrity (fixity) checks with updatable algorithms
- Support preservation object copies over multiple clouds and reduce the data lock-in problem

PDS Cloud and SIRF in ENSURE



Note: SIRF Handler is for future implementation

Summary

- Digital preservation is an important problem that is only growing in importance over time
  - Long term is different from short
  - Digital preservation is different from preserving physical objects
  - Best practices exist, but it is not a solved problem
- Solutions are being developed, but they are
  - Domain specific
  - Based on assumptions about future needs
- The SNIA LTR-TWG is trying to improve the state of the art by developing SIRF
  - An extensible storage format, not for a specific domain
  - Suitable for long term preservation
  - Storable on a wide range of media and technologies
  - SNIA is seeking input on SIRF

Summary

- SIRF container is key to long term information retention, but also of interest to cloud & compliance activities etc.
  - Not trying to re-invent the wheel, leveraging existing work to the maximum extent possible
  - When combined with bit preservation activities will provide a comprehensive set of tools to address long term information retention

About the SNIA LTR TWG

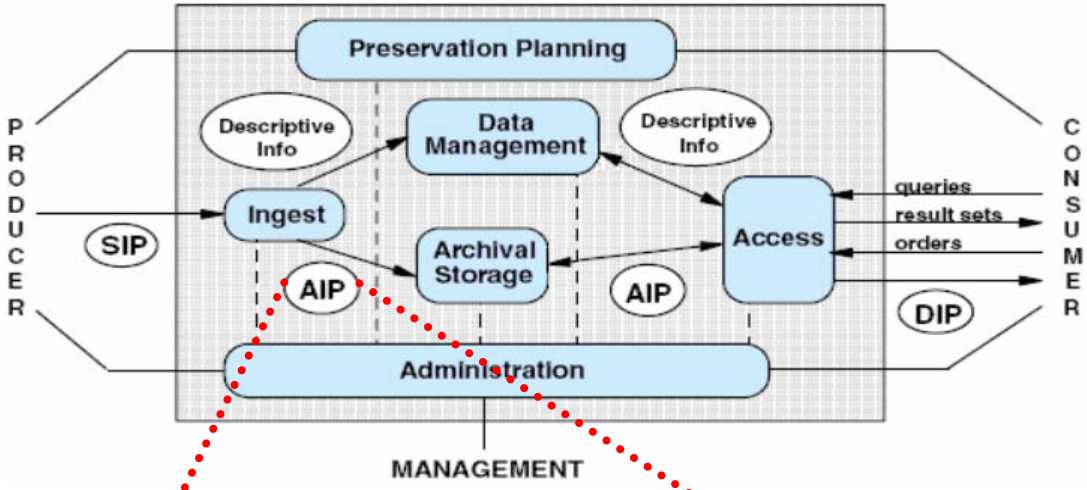
- This presentation has been developed by members of the SNIA Long Term Retention Technical Working Group (LTR TWG)
  - http://www.snia.org/tech_activities/workgroups
- Mission
  - The TWG will lead storage industry collaboration with groups concerned with, and develop technologies, models, educational materials and practices related to, data & information retention & preservation.
- Charter
  - The TWG will ensure that SNIA plays a full part in addressing the "grand technical challenges" of long term digital information retention & preservation, namely both physical ("bit") and logical preservation.
  - The TWG will generate reference architectures, create new technical definitions for formats, interfaces and services, and author educational materials. The group will work to ensure that digital information can be efficiently and effectively preserved for many decades, even when devices are constantly replaced, new technologies, applications and formats are introduced, consumers (designated communities) often change, and so on.
- Please join us!

Backup

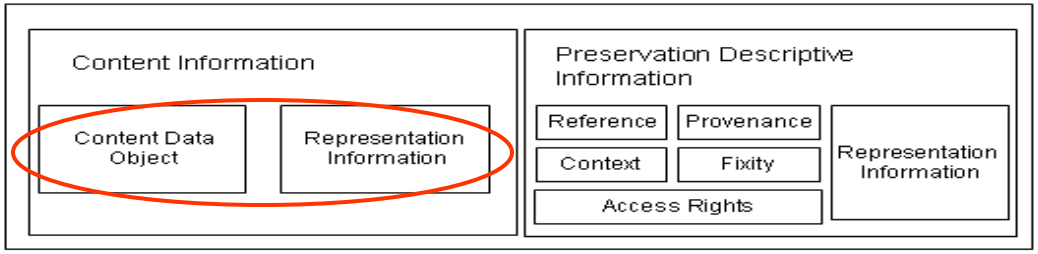
Open Archival Information System (OAIS)

- ISO standard reference model (ISO:14721:2002)
- Provide fundamental ideas, concepts and a reference model for long-term archives
- Includes a functional model that describes all the entities and the interactions among them in a preservation system
- Archival Information Package (AIP) - a logical structure for the preservation object that needs to be stored to enable future interpretation

*** OAIS Functional Model**

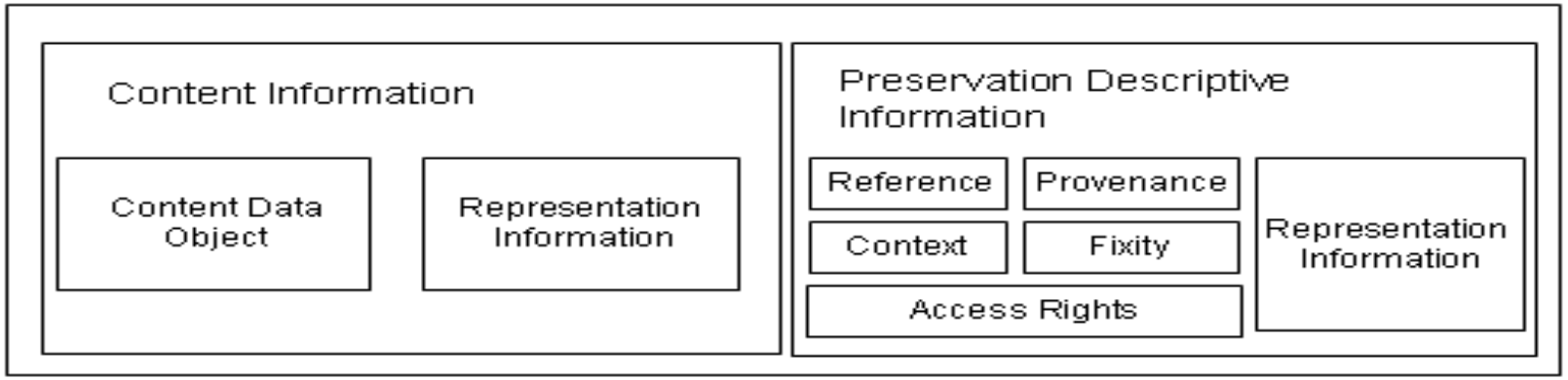


Preservation Object (AIP)



* Figure taken from the OAIS spec

OAIS AIP Logical Structure



- **Content Data Object** - the raw data that is the focus of the preservation.
- **Representation Information** – the information required to interpret the raw data to its designated community.
- **Reference** – globally unique and persistent identifiers for the content information.
- **Provenance** – the history and the origin of the content information and any changes that may have taken place since it was originated, and who has had custody of it since it was originated.
- **Context** – documents reason for creation of the content information and relationship to its environment.
- **Fixity** – a demonstration that the particular content information has not been altered in an undocumented manner.
- **Access Rights** - the information that identifies the access restrictions pertaining to the Content Information, including the legal framework, licensing terms, and access control.

SIRF and XAM

- The eXtensible Access Method specifies a standard API for content addressable storage systems
- XAM API can be used to store and access the various preservation objects and the catalog object in a SIRF-compliant container
- Example
  - ….

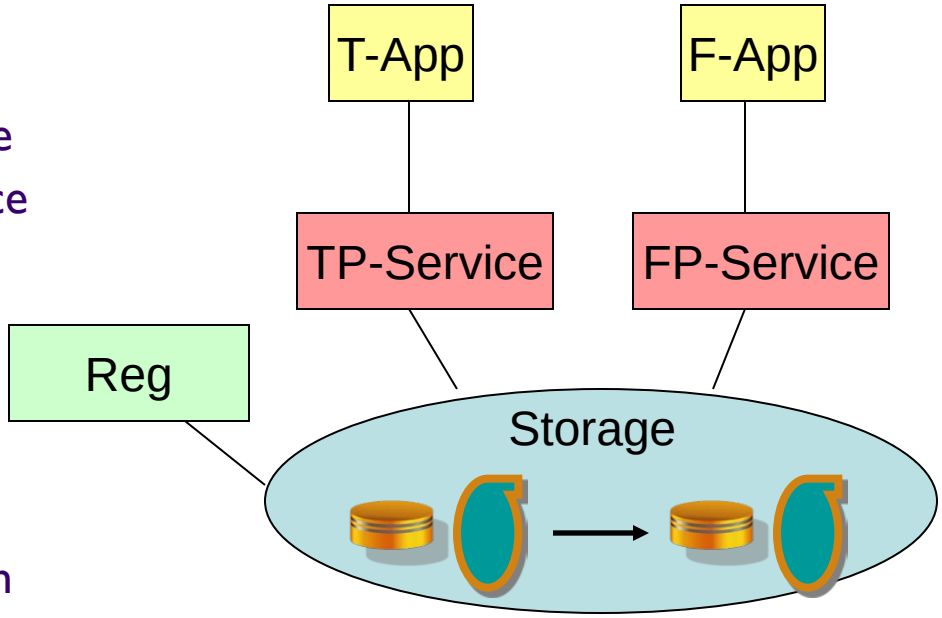
SIRF Use Cases and Functional Requirements Methodology

- Define Actors involved in SIRF
- Define use cases and flows among the actors
  - 4 generic uses cases
    - Unlinked to specific type of data or application
    - Technological changes in the environment
  - 5 Workload-based use cases
    - Specialized for concrete workloads
    - Additional non-technological changes in the environment
- For each use case, find the derived functional requirements
- Aggregate all functional requirements and map use cases to them
- Categorize the functional requirements
  - general requirements, format requirements, data model requirements, performance requirements, etc.
- Prioritize the functional requirements
  - Some of the requirements may conflict each other

SIRF Actors

Non-human actors:

- Storage - Storage subsystem
- TP-Service - Today's preservation service
- FP-Service - Future's preservation service
- T-App - Today's application e.g. Office
- F-App - Future's application
- Reg – Registry



- The storage persists sets of preservation objects

Example Use Case : eMail archive

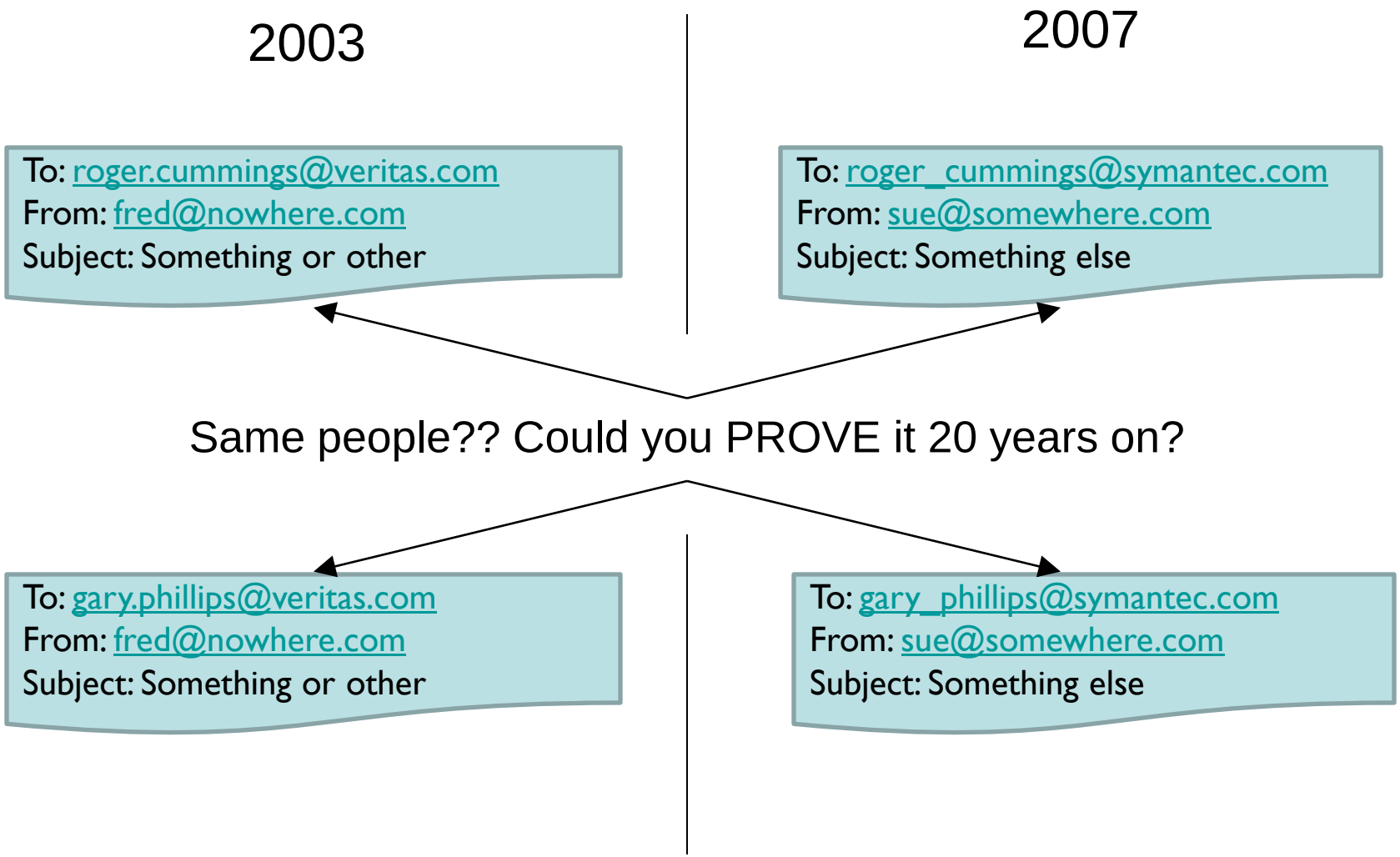
Flow:

1. T-App ingests an e-mail thread today via TP-Service. This includes ingesting a collection of several interrelated Preservation Objects (POs) - thread PO, message POs, attachments POs, PO for the address book, POs for organizational processes, POs for data leakage policies
2. Time passes and the organization changes scope, name, undergoes a merger, etc. As a result, FP-Service creates a set of new version POs for the address book and the organizational processes
3. More time passes and F-App searches the repository and creates POs for the search results to raise performance of future searches. Those new POs may contain soft links to the thread, messages and attachments created in step 1

Main Requirements:

- Support for time stamps (required quality is work-in-progress)
- Support for "special" POs e.g. address book PO, search results PO
  - For lack of a better name, we call these "special" POs - secondary catalog
- Support for hard links and soft links
- Generic support for organizational unique metadata

Real Life Example Problem



Derived SIRF Requirements – a sample

- Support for verification of document provenance and authenticity
  - Regardless of migrations whether logical or physical.
- Support methodology for verification of completeness and correctness
- Support for retention holds that prevent POs being modified or deleted
- Support for links between POs that are as immutable as the objects themselves
  - Either "soft" or "hard" links
- Support for "special" POs, auditable time stamps