

Towards SIRF: Self-contained Information Retention Format

Simona Rabinovici-Cohen
IBM Research – Haifa
simona@il.ibm.com

Mary G. Baker
HP Labs
mary.baker@hp.com

Roger Cummings
Symantec Research Labs
roger_cummings@symantec.com

Sam Fineberg
HP Software
fineberg@hp.com

John Marberg
IBM Research – Haifa
marberg@il.ibm.com

ABSTRACT

Many organizations are now required to preserve and maintain access to large volumes of digital content for dozens of years. There is a need for preservation systems and processes to support such long-term retention requirements and enable the usability of those digital objects in the distant future, regardless of changes in technologies and designated communities. A key component in such preservation systems is the storage subsystem where the digital objects are located for most of their lifecycle. We describe SIRF (Self-contained Information Retention Format) – a logical storage container format specialized for long term retention. SIRF includes a set of digital preservation objects and a catalog with metadata related to the entire contents of the container as well as to the individual objects and their interrelationship. SIRF is being developed by the Storage Networking Industry Association (SNIA)¹ with the intention of creating a standardized vendor-neutral storage format that will be interpretable by future preservation systems and that will simplify and reduce the costs of digital preservation.

Categories and Subject Descriptions

H.3.7 [Information Storage and Retrieval]: Digital Libraries – standards, system issues; H.3.2 [Information Storage and Retrieval]: Information Storage; E.2 [Data Representations]: object representation; H.5.0 [Information Interfaces and Presentation]: General; D.2.0 [Software Engineering]: General – standards.

General Terms

Standardization, management.

Keywords

Long term retention, digital preservation, digital archiving, storage subsystem, cloud storage, metadata, provenance, storage container catalog, standards.

¹Portions of this paper are based on documents that the authors have created as part of their work in SNIA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

1. INTRODUCTION

A growing number of organizations now have a requirement to preserve large volumes of digital content for dozens of years, and to maintain access to it for reasons such as sustainability of business assets, retention of intellectual property, and appreciation of cultural and scientific history. Regulatory compliance and legal issues require preservation of email archives, medical records, financial accounts, aircraft designs, oil-field data, and more. Web services and applications compete to provide storage, organization and sharing of consumer photos, movies, and other creations. Many other fixed-content repositories are charged with collecting and providing access to scientific data, intelligence, books, movies and music.

Long-term digital retention and preservation is the ability to sustain the accessibility, understandability and usability of digital objects in the distant future regardless of changes in technologies and in the “designated communities” (data consumers) that use these digital objects. Unfortunately, preserving and maintaining access to long-lived digital information is still difficult, error-prone, and expensive. Long-term digital content suffers from many threats, including corruption of the digital content, attacks, organizational changes, and obsolescence of hardware and software. Frequent news stories cover organizations’ failures to be able to preserve important content, one such story being NASA’s near loss of the original video/data of the first moon landing, eventually recovered from a set of neglected tape reels [1].

The digital preservation challenge can be divided into “bit preservation” and “logical preservation”. Bit preservation is the ability to retrieve the bits in the face of physical media degradation or obsolescence, corruption or destruction due to errors or malicious attacks, or even environmental catastrophes such as fire and flooding. Logical preservation involves preserving the understandability and usability of the data, despite unforeseeable changes that will take place in servers, operating systems, data management products, applications and even users. Additionally, logical preservation needs to maintain the provenance of the data, along with its authenticity and integrity, so that current and future systems can ensure that only legitimate users access that data. We need preservation systems and processes to enable and support bit and logical preservation. For

affordability and efficiency, all such processes must be performed at scale.

A core standard for digital preservation systems is the Open Archival Information System (OAIS) [2], an ISO standard since 2003 (ISO 14721:2003 OAIS). This standard concentrates on logical preservation and specifies the terms, concepts, and reference models to be used in a system dedicated to preserving digital assets for a dedicated user group (designated community) that needs to access and understand the information preserved. OAIS is a high-level reference model, which means it is flexible enough to be used in a wide variety of environments. However, more detailed steps and workflow stages need to be developed for its implementation.

OAIS defines logical preservation as a recursive problem. In addition to storing the raw data, one must also store the sometimes separately-born (in time and place) metadata that helps interpret and use the raw data. Moreover, this metadata (representation information) may recursively need additional metadata to help interpret it. The recursion ends when the representation information is non-digital and preserved by the general understanding or learning of the designated community. To further support logical preservation, OAIS defines additional metadata that is associated with the raw data and describes its context, logs its provenance, and ensures its fixity (data integrity).

While preserving business, public and personal assets involves various stakeholders and technologies, storage has a key role in this spectrum as it is where the data resides for most of its lifecycle. The term “preservation-aware storage” [3] denotes a storage component with built-in support for preservation. This is the layer of the system that manages the long-term storing and maintenance of digital material. Digital preservation systems can be less costly, more robust and have lower probability of data corruption or loss if they separate out preservation-related functionality (e.g., fixity checks, provenance maintenance, transformation execution) to the preservation-aware storage layer [3].

Recognizing the significance of preservation-aware storage functionality, the Storage Networking Industry Association (SNIA) formed the Long Term Retention (LTR) Technical Working Group (TWG) [4] in 2008 to address storage aspects of digital retention. The LTR TWG is working on Self-contained Information Retention Format (SIRF) [5] to create a standardized vendor-neutral storage format that will help its customers interpret preservation objects in the future, even by systems and applications that do not exist today. SIRF provides strong encapsulation of large quantities of metadata with the data at the storage level, and enables easy migration of the preserved data across storage devices.

The SIRF format is intended for a mountable unit storage container but should be agnostic to the actual medium. While some data is preserved on CDs and DVDs, tape and disk storage systems are currently the predominate types of media on which large quantities of data are preserved. In

many cases, the preservation data tends to be cold (inactive) and is seldom accessed over time. SIRF is agnostic to the type of storage medium as well as to any type of data. SIRF enables the mountable unit storage container to be self-describing and self-contained to the extent possible. This capability is particularly useful for offline storage media, which are often deployed for cold data, but SIRF also works well for storage that remains online.

This document describes SIRF and the motivation for it, along with some of its use cases and requirements. We derive the desired functional requirements of the SIRF format and the system that implements and uses it from the use cases. We also lay out the basis for the specification by introducing SIRF levels and some of the metadata in each level. SIRF is a significant advance over traditional storage formats, which are oblivious to the needs of long-term retention. In contrast to the semantics of traditional file systems, which include only limited metadata about each file, SIRF provides for the rich metadata needed for preservation and ensures its grouping with the data.

The rest of this paper is organized as follows. In section 2, we review related work. In section 3, we introduce the SIRF container format and discuss its properties. We also position SIRF with respect to other related specifications. Section 4 provides several workload-based example use cases, revealing various requirements for SIRF. Section 5 describes the SIRF levels. Section 6 concludes with a discussion of future work.

2. RELATED WORK

A growing number of studies focus on the storage aspects of digital preservation. Long-term preservation systems differ from traditional storage applications with respect to goals, characteristics, threats, and requirements. Baker et al. [6, 7] examine these differences and suggest bit preservation guidelines and alternative architectural solutions that focus on replication across autonomous sites, reduced per-site engineering costs, and the ability to scale over time and technologies. They take into account faults due to humans and organizations in addition to hardware and software, and present an extended reliability model along with several strategies for reducing the probability of irrecoverable data loss.

Storer et al. [8, 9] discuss security threats that arise when storing data for long periods of time. This includes common threats such as loss of integrity, failure of authentication and compromise of privacy, as well as new specific threats such as slow attacks. The papers examine how existing systems address these concerns and suggest methods to ensure secured long term survivability.

Preservation DataStores (PDS) [10, 11] is storage architecture for OAIS-based preservation-aware storage that focuses on supporting logical preservation. It is motivated by the notion that digital preservation systems will be more robust and have lower probability for data

corruption or loss if they offload preservation-related functionality to the storage layer. Offloading enables the storage layer to reduce the amount of data transferred to applications and to improve its own data placement in favor of preservation. The requirements for preservation-aware storage are characterized in a position paper by Factor et al. [3]. PDS was partially developed in the framework of CASPAR [12], an EU FP6 project concerned with preservation of cultural, artistic, and scientific knowledge.

Offloading data maintenance functions from the application to the storage system is an ongoing trend. Functionality such as bit-to-bit data migration, block-level data integrity, and encryption are now carried out by advanced intelligent disk and tape subsystems. For instance, Muniswamy-Reddy et al. [13] introduce PASS (Provenance-aware Storage Systems), a storage system that collects and tracks the provenance of data objects. PASS is further described below.

Dappert and Enders [14] discuss the importance of metadata in a long-term preservation solution. The authors identify several categories of metadata, including descriptive, preservation-related, and structural, arguing that no single existing metadata schema accommodates the representation of all categories. The work surveys metadata specifications contributing to long-term preservation.

The LOCKSS (Lots of Copies Keep Stuff Safe) project, originating at Stanford University and described by Maniatis et al. [15], provides an open-source, peer-to-peer decentralized digital preservation infrastructure. It includes software that turns an ordinary personal computer into a digital preservation appliance and implements a protocol for auditing and repairing the online content that is resistant to both sudden and slow attacks

The National Digital Information Infrastructure and Preservation Program (NDIIPP) [16] is a collaborative initiative run by the US Library of Congress. Its goal is to develop a national strategy for digital preservation. NDIIPP develops a rich set of formats appropriate for preservation and provides a high-level architecture.

Cloud technology is emerging as an infrastructure suitable for building large and complex systems. Storage and computing resources hosted and provisioned by a third party present a scalable and cost-effective alternative to traditional in-house computing. Thus, the cloud is clearly an attractive platform for long term preservation solutions, and in particular, cloud storage can be leveraged for preservation-aware storage. However, cloud preservation strategies must take into account new threats, such as the possible discontinuation of any chosen cloud service.

Muniswamy-Reddy et al. [17] make the case that provenance is crucial for data stored on the cloud, and they identify the properties of provenance that enable its utility. Several alternative protocols for maintaining data provenance in current cloud stores are presented, which were implemented using PASS augmented to use Amazon Web Services. Based on an evaluation of the implemented

protocols, the authors discuss challenges for providing native provenance support on the cloud. Challenges of provenance in the cloud are also examined by Sakka et al. [18]. This work identifies multiple categories of constraints on provenance, and consequently proposes alternative approaches to organizing provenance management in the cloud.

The DuraCloud open source platform [19] developed by DuraSpace aims to provide a fully integrated environment where services and data can be managed across multiple cloud providers. DuraCloud will be offered as a hosted service providing data storage, replication and access, and services to support data preservation, such as data format transformation and fixity checking.

Recently, the ENSURE (Enabling kNowledge Sustainability, Usability and Recovery for Economic Value) project [20] was launched by an EU FP7 consortium led by IBM. ENSURE aims at extending the state of the art in digital preservation, focusing on business and scientific use cases, such as health care and financial data. The architecture is consistent with the OAIS reference model. Preservation oriented storage services are built as add-ons to available cloud storage infrastructures. ENSURE will leverage concepts from PDS and SIRS in the cloud storage environment.

3. WHAT IS SIRS?

Archivists and records managers of physical items such as documents, records, etc., avoid processing each item individually. Instead, they gather together a group of items that are related in some manner - by usage, by association with a specific event, by timing, and so on - and then perform all of the processing on the group as a unit. The group itself may be known as a series, a collection, or in some cases as a record or a record group. Once assembled, an archivist will place the series in a physical container (e.g., a file folder or a filing box of standard dimensions), mark the container with a name and a reference number and place the container in a known location. Information about the series will be included in a "finding aid" such as an online catalog that conforms to a defined schema and gives the name and location of the series, its size, and an overview of its contents.

We propose an approach to digital content preservation that leverages the processes of the archival profession thus helping archivists remain comfortable with the digital domain. One of the major needs to make this strategy possible is a digital equivalent to the physical container - the archival box or file folder - that defines a series, and which can be labeled with standard information in a defined format to allow retrieval when needed. Self-contained Information Retention Format (SIRS) is intended to be that equivalent - a storage container format for a set of (digital) preservation objects that also provides a catalog with metadata related to the entire contents of the container as well as to the individual objects and their

interrelationship. This logical container makes it easier and more efficient to provide many of the processes that will be needed to address threats to the digital content. Easier, more efficient preservation processes in turn lead to more scalable and less costly preservation of digital content.

3.1 Preservation Object

As SIRF containers comprise preservation objects, we first define what these objects are. A preservation object is a digital information object that includes the raw data to be preserved plus additional embedded or linked metadata needed to enable sustainability of the information encoded in the raw data for decades to come. The preservation object is the basic unit in the storage of a preservation system. It may be subject to physical and logical migrations, making it an updateable object over time. An updated preservation object is a new version of the original, and its audit log records the changes that have occurred so authenticity may be verified.

The OAIS Archival Information Package (AIP) standard [2] is an example of a preservation object. The main elements of OAIS AIP are depicted in Figure 1.

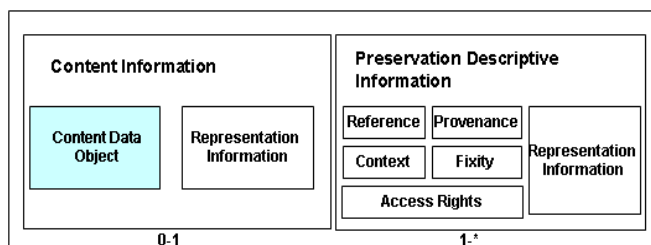


Figure 1. OAIS AIP logical structure.

An AIP contains zero or one Content Information part, and one or more Preservation Description Information (PDI) parts. The Content Information part includes the raw data along with its representation information. The PDI parts provide additional metadata related to the content's logical preservation. More specifically, Content Information comprises the Content Data Object (the raw data) that is the focus of the preservation, as well as the Representation Information (RepInfo) that is needed to render the object intelligible to its designated community. This may include information regarding the hardware and software environment needed to view the content data object.

Note that the RepInfo is a recursive object and may have an additional RepInfo to interpret itself. This recursion ends when facing a RepInfo that is non-digital and preserved by the designated community. For example, astronomical data represented in a FITS file is associated with RepInfo that includes a dictionary to describe the FITS keywords. The FITS dictionary is associated with a RepInfo that includes the dictionary structure specification. Assuming the dictionary specification is in XML, its RepInfo includes the XML specification, which is associated with a RepInfo that includes the Unicode specification. We assume that the

Unicode specification is preserved by the designated community, and thus it does not need an additional RepInfo.

The PDI is further divided into four sections: reference (globally unique and persistent identifiers), provenance (chain of custody, the history and the origin of the content information custody), context (relationships of the content information to its environment), fixity (a demonstration that the particular content information has not been altered in an undocumented manner), and access rights.

OAIS describes the elements that should be within an AIP without specifying their format or how they are packaged together. Some standards are emerging for specific designated communities that provide specification for the actual format and packaging of a preservation object. Examples of such standards are the XML Formatted Data Unit (XFDU) [21] for space data, the VERS Encapsulated Object (VEO) [22] for electronic records, the Metadata Encoding and Transmission Standard (METS) [23] for digital libraries, PREservation Metadata: Implementation Strategies (PREMIS) [24], and Long Term Archiving and Retrieval (LOTAR) for aerospace data.

SIRF does not specify the preservation object format. Preservation objects are generally created by applications and services defined outside of the storage subsystem, and their formats tend to be domain-specific. Moreover, the storage subsystem may include multiple formats of preservation objects, and must be supported by SIRF. Specifically, SIRF is scoped to define the metadata and format in its catalog, which includes information about the preservation objects, the relationship among these objects and information to support implementation of preservation processes.

One of the processes performed upon a preservation object is migration, which is essential for long-term digital retention and preservation. The migration process includes the act of moving data from one system to another because of a change. The nature of the change may include (but is not limited to): possible decay of the storage media, impending obsolescence of hardware or software, change in availability of software or documentation (copyright issues), and change in external environment such as organization or staff. Once created, the preservation objects are generally immutable, but new versions may be created over time, for instance when a migration is performed. SIRF needs to support these immutable objects and migration processes.

3.2 SIRF Definition

SIRF is a logical container format for the storage subsystem, appropriate for the long-term storage of digital information. It is a logical data format of a mountable unit, e.g., a filesystem, a block device, a stream device, an object store, a tape, etc. It assumes the mountable unit includes an object interface layer that constructs objects out of the sectors and blocks. Some advanced storage subsystems

provide a built-in object interface as in the case of Object storage, Cloud storage and Extensible Access Method (XAM) storage. Other, more lower level storage subsystems, have specialized media-dependent standards to expose object interfaces, as in the case of UDF (Universal Disk Format - ISO/IEC 13346) for DVDs, CDFS (Compact Disc File System - ISO 9660) for CDs, FAT (File Allocation Table) for disks, and LTFS (Long Term File System) for tapes.

Figure 2 illustrates the SIRF container, which includes the following components:

- A magic object that identifies whether this is a SIRF container and gives its version. The magic object is independent of the storage medium and has an agreed defined name and a fixed size. It includes means to access the SIRF catalog.
- Preservation objects, which are immutable. The container may include multiple versions of a preservation object.
- A catalog that is updateable and contains metadata needed to make the container and its preservation objects portable into the future without relying on functions external to the storage subsystem.

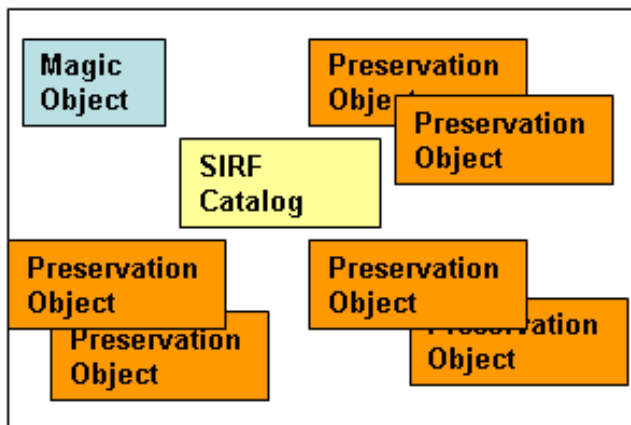


Figure 2. SIRF components.

SIRF is defined using a layered approach, with two levels. The levels differ in their catalog. The SIRF level 1 catalog contains unique metadata, some of which is not included within the preservation objects, but is mandatory to make those preservation objects portable into the future. Examples of such metadata are retention hold, reference counts, preservation object fixity algorithms, fixity values and fixity calculation dates. The SIRF level 2 catalog includes additional information that may also be included within the preservation objects, and is used for fast access to the preservation objects. Examples of such metadata are links to representation information needed to assure referential integrity, metadata about the relationship among the preservation objects, packaging format, etc.

3.3 SIRF Properties

To make it easier to move content between systems and technologies, while ensuring it remains complete and interpretable, we need a standard way to store the content that is self-contained, self-described, and extensible. The metaphor we use is a closed bottle that includes all the information needed to understand the bottle's contents at a future point in time. The SIRF icon is depicted in Figure 3.



Figure 3. SIRF icon.

Thus, the following are the key properties of the SIRF long-term storage container format:

Self-contained: Long-term retention requires the preservation of both data and its metadata, which can become disaggregated. To prevent this from happening, the unit of storage for an object should include both the data and its metadata, so that they are treated and moved together as a single storable unit that will be kept intact for the life of the object. Similarly, the unit of storage for the objects' container should include both the objects and the metadata about the objects and their interrelationship.

Self-described: It should be possible to look at a data package and determine what it is, so that we can interpret it correctly. For example, it should be possible to determine the objects within the container and their associated metadata. One problem is that the self-description of the container must also be interpretable. If it is complex, then it too must be self-describing. Because of this recursive problem, a completely self-describing format is impossible to achieve. However, self-describing formats remain useful if at the root of the recursion they use only very widely used formats, such as ASCII, and the self-description itself can be updated over time. While it is possible to create self-describing proprietary formats, widely used industry standard formats are more likely to have a long life.

Extensible: It is impossible to predict all of the changes likely to be needed for information retained for decades. As these changes occur, we want to preserve information about what changes we made and when. For example, we need to record information about format migrations and may also want to keep the original container tied to its rendition in a new format. As another example, we may want to add information about changes in custody of the container or be able to add new types of information to existing content. A

good long-term storage container format must allow for additions and extensions while preserving the integrity of the original data.

3.4 SIRF and Related Specifications

SIRF is related to several existing specifications, including the OAIS reference model for digital preservation that we previously described. SIRF is OAIS-aware, and preservation objects in SIRF may utilize the OAIS AIP. Below we describe SIRF's relationship to other existing specifications.

Implementations of SIRF may use Extensible Access Method (XAM) [25], a SNIA and ISO standard that defines an interface between consumers (application and management software) and providers (storage systems). XAM can be used to provide an object interface for SIRF, and the XAM interface can be used to access SIRF containers and the contained preservation objects.

The open source JHOVE (JSTOR/Harvard Object Validation Environment) characterization tool is an important component of many digital repositories and preservation workflows. The Library of Congress is currently developing the next-generation JHOVE2 architecture [26] for format-aware characterization. JHOVE2 employs the DROID (Digital Record Object Identification) tool [27] developed by the UK National Archives, which performs automatic format identification of a file. JHOVE is orthogonal to SIRF and the combination of the two can be very powerful. JHOVE2 is a tool to be used to identify the characterization of a Content Data Object (CDO, see Figure 1), including its format. This characterization can be used as additional representation information to enrich the preservation object of that CDO stored in a SIRF-compliant storage subsystem. Or, it can be used to identify the format of the CDO after it was read from SIRF-compliant storage.

BagIt [28] is a hierarchical file packaging format currently being developed by the US Library of Congress and published as an internet draft of the IETF Network Working Group. A bag consists of a payload that is the custodial focus of the bag and is treated as semantically opaque. The bag also includes tags that are metadata files intended to facilitate and document the storage and transfer of the bag. The tags include information such as the listing of payload files and corresponding checksums, the organization transferring the content, the date that the content was prepared for delivery.

While BagIt is more intended for a single preservation object, SIRF is more focused on a storage container of a mountable unit that includes multiple preservation objects. It includes metadata in its catalog and numerous preservation objects. The catalog metadata includes much broader information than provided in BagIt to help interpret the preservation objects as well as the interrelationship among those preservation objects in the container.

4. SIRF USE CASES

We now present some workload-based use cases and the requirements derived from them. These use cases have been described to us by members of the communities from which they come. The requirements we derive are often targeted at the underlying storage system, but SIRF must provide support for enabling these requirements. First, we define the actors in a preservation system in relation to SIRF.

4.1 Actors

The actors in a preservation system that relate to SIRF, as illustrated in Figure 4, are

- Storage – Storage subsystem that stores numerous preservation objects.
- TP-Service – Today's preservation service, e.g., ingest service, transformation service.
- FP-Service – Future preservation service which may be unknown today.
- T-App – Today's application that generates digital data, e.g., a word processor, email application.
- F-App – Future application which may be unknown today.
- Registry – Registry that stores representation information of the used storage formats, e.g., the specification documents of the used formats.

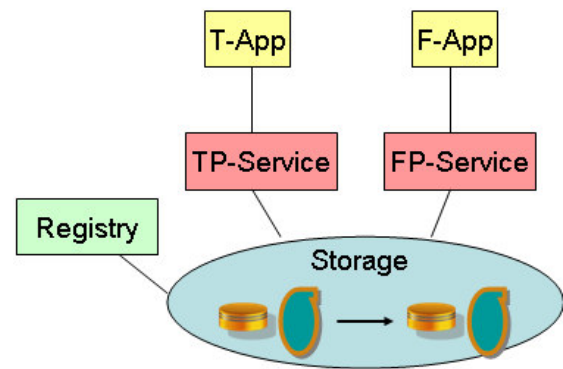


Figure 4. Preservation system actors.

4.2 eDiscovery Use Case

Discovery is the formal legal process of finding information relevant to a legal matter and delivering it to opposing counsel. eDiscovery is discovery as applied to electronically stored information. Preservation Objects (POs), like any other electronic information, can be subject to eDiscovery.

The use case flow is

1. T-App ingests a PO today via TP-Service.
2. Time passes and the data becomes subject to eDiscovery.

3. Potentially responsive POs are identified using provenance, context and content information stored with POs.
4. Identified POs are put on “legal hold,” preventing deletion or modification.
5. Identified POs are copied from the preservation system and collected to a case repository for processing, review, and analysis.
6. At some future date the “legal hold” is removed. A PO may become subject to other legal holds or retention/disposition policies at any time.

The main requirements derived from this use case are

- Support for retention holds on POs that prevent their deletion or modification.
- 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 storing audits. The audits can include records about modification, possibly records about access, etc.
- It needs to be possible to identify, collect and preserve POs that are relevant to a legal matter.

4.3 Email Archive Use Case

Email data may include interrelated objects and many repetitions. An email thread includes one or more messages where each message is an email by itself and can contain zero or more attachments. The following flow is one method of preserving emails, used here to derive SIRF requirements, but other methods may exist.

The use case flow is

1. T-App ingests an e-mail thread today via TP-Service. This includes ingesting a collection of several interrelated POs as follows
 - Ingest a new PO for the thread. The PO metadata should include all mail header information, auditable date information, keywords, etc., including allowance for organizationally-unique metadata.
 - For each message within the thread, check if a PO already exists for that message. If it does, create a link from the thread PO to the existing message PO. If not, ingest a new PO for the message and a link from the thread PO to the newly created message PO.
 - For each file attachment within the message, ingest a PO for that attachment and a link from the message PO to the attachment PO.
 - Ingest one or more POs for information upon which the thread depends, such as a PO for the address

book, POs for organizational processes, POs for data leakage policies, etc.

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. These include a new version PO for the address book, new version POs for the new organizational processes, and new version POs for data leakage policies. Note that the thread, message and attachment POs created in step 1 are not affected.
3. More time passes and F-App searches the metadata of threads, messages and attachments in parallel to find relevant POs. F-App creates POs for the search results to raise the performance of future searches and ingests them into the preservation system via the FP-service. These new POs may contain links to the thread, messages and attachments created in step 1.

The main requirements derived from this use case are

- Support for links between POs that are as immutable as the objects themselves. The links can be either “hard links” that require the existence of the object linked to within the SIRF container, or “soft links” that can refer to an object external to SIRF. Soft links may be desirable or even necessary at times, but they violate our goal of self containment.
- Support for auditable time stamps that are immutable and created by a known authority.
- Support for “special” POs such as a PO that includes an address book, or a PO that includes search results.
- Generic support for organizationally unique metadata.

4.4 Consumer Archive on the Cloud Use Case

An individual wants to preserve his family photos and documents in a cloud that provides preservation services, so that forthcoming generations will be able to access that data and study their roots. For simplicity, we assume here that the cloud service continues to operate during all the generations described in this use case. We examine another use case that also involves changes in the cloud service itself in Section 4.6.

The use case flow is

1. A user creates a genealogy container for his genealogy-related documents on a cloud that provides SLAs for preservation.
2. The user uses T-App to ingest a genealogy-related document via TP-service on the cloud.
3. TP-service on the cloud ingests the PO with the original document and also transforms the document to a standardized format believed to be more sustainable (such as PDF/A [29]) and ingests the resulting PO version to the same genealogy container.
4. Time passes and the grandchildren would like to get that document.

5. FP-service will validate the grandchildren's identity and will provide appropriate credentials to access the genealogy container and the document.
6. F-App accesses via FP-Service the latest version of the document and renders the PDF/A document.

The main requirements derived from this use case are

- Support for transformations of the PO, e.g., support for various versions of the PO and the tree structure they create.
- Support for managing identifiers over time
- Support secured access to the data that is updatable over time e.g., when a security mechanism becomes weak.
- Support for cloud containers to be SIRF-compliant, so containers can be migrated to other clouds with all the required preservation information.
- Support for verification of document provenance and authenticity, regardless of migrations whether logical or physical.

4.5 Biomedical Bank Use Case

Assume a large hospital that has an adjacent academic medical research center. It stores the patients' biomedical data in a biomedical bank in which data is preserved for decades. The data is used at the point of care as well as for biomedical research by the adjacent research center.

The use case flow is

1. T-App ingests via TP-service a PO that includes a standardized Digital Imaging and Communications in Medicine (DICOM) image of the leg of a patient that is a minor.
2. Time passes and the patient, who is now an adult, schedules an appointment to check a new problem he has encountered in his leg.
3. FP-service will identify the data needed for the scheduled appointment using reference, context and provenance information.
4. The identified POs will be brought ahead of the appointment from offline media to an online system to be quickly accessible for the appointment.
5. F-App at the point of care accesses the identified POs for the patient via FP-Service.
6. More time passes and a researcher from the adjacent academic medical research center wants to access that image for research purposes. According to HIPAA regulations, the researcher can get only an anonymous image.
7. F-App accesses the anonymized PO via FP-Service.

The main requirements derived from this use case are

- Support hierarchical storage management, e.g., support unique IDs for the POs regardless of the storage tier, and support on-line and off-line storage.

- Support masking of sensitive data, e.g., support storing POs for anonymization modules within the SIRF container.
- Support verification of document provenance and authenticity, regardless of migrations whether logical or physical.

4.6 Merged Cloud Repositories Use Case

Assume we have two cloud storage providers named "FirstCloud" and "SecondCloud" that provide preservation services. We consider the case where they merge their services at some point. Further requirements would be derived from considering a use case where a deployed service is suddenly discontinued.

The use case flow is

1. T-App ingests via TP-service a PO in a cloud that is provided by "FirstCloud".
2. T-App also ingests via TP-service a second PO in a second cloud provided by "SecondCloud".
3. Time passes and the two companies "FirstCloud" and "SecondCloud" are merged and their two cloud repositories are combined. This is possible as the POs identifiers are globally unique.
4. F-App accesses via FP-Service the two POs in the combined cloud provided by the merged company.

The main requirements derived from this use case are

- Support cloud containers to be SIRF-compliant, so containers can be interpreted by other clouds.
- Persistent globally unique identifiers for POs.
- Optional deduplication of POs in the merged repositories.

5. SIRF LEVELS

SIRF is defined using a layered approach. There are two levels, which differ in the specification of metadata in the catalog. The SIRF level 1 catalog contains mandatory information for preservation, parts of which cannot be deduced from other data in the storage container. SIRF level 2 contains additional information that can be used for fast access to the Preservation Objects (POs), but is not mandatory for preservation.

We now illustrate some of the metadata in each level. This should be considered preliminary, whereas the SIRF specification is still under development.

SIRF level 1 information describing the container includes details such as specification ID and version, provenance of the container including who created it and for what purposes, and the container's audit object ID.

In addition, the SIRF level 1 catalog includes for each PO in the container, metadata such as:

- Identifiers – PO ID that is unique over offline and online storage, PO copy ID, and PO derived versions IDs.
- Dates – creation date using auditable time stamp.
- Fixity – last fixity check date, fixity algorithm (may be multiple), fixity value (may be multiple).
- Retention – retention hold reference count, retention date, deletion hold reference count.
- Audit – audit object ID.

The SIRF level 2 catalog must be SIRF level 1 compliant. It includes additional information on the container, such as fixity algorithms and fixity values of the whole medium, and identifiers of special POs, (such as address book PO, search result PO, de-identification PO).

SIRF level 2 also specifies interlinks between POs, categorized to “hard links” (the object linked to must exist in the container), and “soft links” (the object linked to can be external to the container).

For each PO in the container, the SIRF level 2 catalog includes metadata such as:

- Names – preservation object name and title.
- Packaging format and standards used in the PO.
- Links – link to representation information, link to provenance, link to context, link to access rights.

Dividing the SIRF specification into two levels makes it simpler to construct a SIRF compliant container of level 1, while deferring more complex optional details to level 2. It should be observed that in some situations, even information not considered in level 1 could be deemed mandatory. For example, if we maintain level 2 metadata of each object in the SIRF catalog, the system can perform object de-duplication even if it loses the duplicate object unique metadata.

6. CONCLUSIONS AND FUTURE WORK

There is an increasing need for preservation systems that can preserve myriad types of information for decades or even centuries. Using open standards in such systems is a basic design principle; open standards are more likely to result in solutions that are robust, system-independent, longer-lived, and interoperable.

In this paper, we have described the standardization work of SIRF, a long-term storage container format. SIRF provides strong encapsulation of large quantities of metadata together with the data at the storage level, and allows easy migration of the preserved data across storage devices. It enables a mountable storage container to be self-describing and self-contained to the extent possible. SIRF is currently under development in the SNIA Long-Term Retention (LTR) Technical Working Group. We have published the SIRF use cases and requirements draft for public review [5], and we have laid out the basics for the actual specification.

One of the difficult issues in preservation systems is validation methods. How does one validate today that the storage format will be interpretable by storage subsystems in the unforeseeable future? This is a challenge that the preservation community continues to explore. In addition, validation processes need to consider costs factors. Some projects, such as CASPAR and ENSURE, develop validation methods for preservation systems; we can consider these methods for SIRF validation.

Another direction for future exploration is the use of SIRF in a cloud storage environment. The emerging cloud storage is highly distributed, scalable and offers availability, accessibility and sharing of data at low cost. However, cloud storage platforms are designed for general use, and are not particularly tailored for preservation needs. As of today, cloud providers offer minimal SLAs, and rely on the customer's trust that the service is performed properly and will continue to exist. Some open questions in this domain include: how to utilize SIRF in a cloud storage environment; what additional metadata is needed in the SIRF catalog for the cloud environment; can a storage container be federated over multiple storage clouds, and how would this affect SIRF?

ACKNOWLEDGEMENTS

We would like to acknowledge Gary Zasman and Michael Peterson who contributed to the initiation of SIRF in SNIA.

REFERENCES

- [1] NASA, “The Apollo 11 Telemetry Data Recordings: A Final Report,” December 2009, http://www.hq.nasa.gov/alsj/a11/Apollo_11_TV_Tapes_Report.pdf.
- [2] ISO 14721:2003, Blue Book. Issue 1. CCSDS, 650.0-B-1: Reference Model for an Open Archival, Information System (OAIS), 2002.
- [3] M. Factor, D. Naor, S. Rabinovici-Cohen, L. Ramati, P. Reshef, and J. Satran, “The need for preservation aware storage - a position paper,” *ACM SIGOPS Operating Systems Review*, Special Issue on File and Storage Systems, 14(1):19-23, January 2007.
- [4] SNIA Long Term Retention (LTR) Technical Working Group <https://www.snia.org/apps/org/workgroup/ltrtwg>.
- [5] Self-contained Information Retention Format (SIRF) use cases and functional requirements, working draft – version 0.5a, SNIA, September 2010, http://www.snia.org/tech_activities/publicreview/SIRF_Use_Cases_V05a_DRAFT.pdf.
- [6] M. Baker, K. Keeton, and S. Martin, “Why traditional storage systems don’t help us save stuff forever,” Technical Report 2005-120, HP Laboratories Palo Alto, June 2005.
- [7] M. Baker, M. Shah, D. Rosenthal, M. Roussopoulos, P. Maniatis, TJ Giuli, and P. Bungale, “A fresh look at the reliability of long-term digital storage,” in *EuroSys*

- 2006: *1st ACM SIGOPS European Systems Conference*, Leuven, Belgium, pp. 221-234, April 2006.
- [8] M.W. Storer, K.M. Greenan, and E.L. Miller. "Long-term threats to secure archives," in *StorageSS 2006: 2nd International Workshop on Storage Security and Survivability*, Alexandria, VA, pp. 9-16, October 2006.
- [9] M.W. Storer, K.M. Greenan, E.L. Miller, K. Voruganti. "POTSHARDS - a secure, recoverable, long-term archival storage system," *ACM Transactions on Storage*, 5(2), article 5, June 2009.
- [10] M. Factor, D. Naor, S. Rabinovici-Cohen, L. Ramati, P. Reshef, J. Satran, and D.L. Giaretta. "Preservation DataStores: Architecture for Preservation Aware Storage," in *MSST 2007: 24th IEEE Conference on Mass Storage Systems and Technologies*, San Diego, CA, pp. 3-15, September 2007.
- [11] S. Rabinovici-Cohen, M.E. Factor, D. Naor, L. Ramati, P. Reshef, S. Ronen, J. Satran, and D.L. Giaretta, "Preservation DataStores: New storage paradigm for preservation environments," *IBM Journal of Research and Development*, Special Issue on Storage Technologies and Systems, 52(4/5):389-399, July/September 2008.
- [12] CASPAR: Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval, EU FP6 Project, <http://www.casparpreserves.eu>.
- [13] K-K. Muniswamy-Reddy, D.A. Holland, U. Braun, and M. Seltzer, "Provenance-aware storage systems," in *USENIX'06: 2006 USENIX Annual Technical Conference*, Boston, MA, pp. 43-56, May 2006.
- [14] A. Dappert, and M. Enders, "Digital preservation metadata standards," *Information Standards Quarterly*, Special Issue on Digital Preservation, 22(2):4-12, Spring 2010.
- [15] P. Maniatis, M. Roussopoulos, T.J. Giuli, D.S.H. Rosenthal, and M. Baker, "The LOCKSS peer-to-peer digital preservation system," *ACM Transactions on Computer Systems*, 23(1):2-50, February 2005.
- [16] NDIIPP: National Digital Information Infrastructure and Preservation Program, a Collaborative Initiative of the Library of Congress, <http://www.digitalpreservation.gov>.
- [17] K-K. Muniswamy-Reddy, P. Macko, and M. Seltzer, "Provenance for the Cloud," in *FAST'10: 8th USENIX Conference on File Technologies*, San Jose, CA, pp. 197-210, February 2010.
- [18] M.A. Sakka, B. Defude, and J. Tellez, "Document provenance in the cloud: constraints and challenges," in *EUNICE 2010: 16th International EUNICE/IFIP WG 6.6 Workshop: Networked Services and Applications – Engineering, Control and Management*, Trondheim, Norway, LNCS 6164, pp. 107-117, June 2010.
- [19] DuraCloud: Technology and Storage by DuraSpace, <http://www.duraspace.org/duracloud.php>.
- [20] ENSURE: Enabling Knowledge Sustainability, Usability and Recovery for Economic Value, EU FP7 Project, http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION=D&CAT=PROJ&RCN=98002.
- [21] XML Formatted Data Unit (XFUDU) Structure and Construction Rules, Recommended Standard CCSDS-661.0-B-1, The Consultative Committee for Space Data Systems (CCSDS), September 2008, <http://public.ccsds.org/publications/archive/661x0b1.pdf>.
- [22] VERS: Management of Electronic Records PROS 99/007 (Version 2), The Victorian Electronic Records Strategy, <http://www.prov.vic.gov.au/vers/standard/version2.asp>.
- [23] METS: Metadata Encoding and Transmission Standard, <http://www.loc.gov/standards/mets>.
- [24] PREMIS: PREservation Metadata: Implementation Strategies, <http://www.loc.gov/standards/premis>.
- [25] XAM: eXtensible Access Method, SNIA XAM Initiative, <http://www.snia.org/forums/xam>.
- [26] JHOVE2: The Next Generation Architecture for Format-Aware Characterization, <http://www.jhove2.org>.
- [27] DROID: Digital Record Object Identification, <http://sourceforge.net/projects/droid>.
- [28] The BagIt File Packaging Format (0.96), IETF Network Working Group Internet Draft, October 2010, <http://tools.ietf.org/html/draft-kunze-bagit-05>.
- [29] ISO 19005-1:2005, Document Management - Electronic document file format for long term preservation - Part 1: Use of PDF 1.4 (PDF/A-1), ISO International Organization for Standardization, June 2005.