Self-contained Information Retention Format For Future Semantic Interoperability

Simona Rabinovici-Cohen[1], Roger Cummings[2], and Sam Fineberg[3]

[1] IBM Research – Haifa
simona@il.ibm.com
[2] Antesignanus
roger@antesignanus.com
[3] HP Storage
fineberg@hp.com

Abstract. Long term preservation of digital information, including machine generated large data sets, is a growing necessity in many domains. A key challenge to this need is the creation of vendor-neutral storage containers that can be interpreted over time. We describe SIRF, the Self-contained Information Retention Format, which is being developed by the Storage Networking Industry Association (SNIA) to support this challenge. We define the SIRF components, its metadata, categories and elements, along with some security guidelines. SIRF metadata includes the semantic information as well as schema and ontological information needed to preserve the physical integrity and logical meaning of preservation objects. We also describe how the SIRF logical format is serialized for storage containers in the cloud and for tape based containers. Aspects of SIRF serialization for the cloud are being experimented with OpenStack Swift object storage in the ForgetIT EU project.

1 Introduction

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 genomics, medical records, astronomy, atmospheric science, 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. This includes both the data itself as well as semantic metadata necessary to enable search, access, and analytics on that data in the far future.

The Storage Networking Industry Association (SNIA) conducted a "100 year archive" survey. It found that 83% of the organizations surveyed have digital assets they need to retain for over 50 years, and 53% have information they need to retain "permanently". Recognizing these challenges, SNIA formed the Long Term Retention (LTR) group [1] to address storage aspects of digital retention. LTR is working on the Self-contained Information Retention Format (SIRF), to create a standardized vendor neutral storage format that will help its users 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.

Both cloud storage and tape technologies are viable alternatives for storage of data for the long term. Cloud technology is emerging as an infrastructure suitable for building large and complex systems, presenting a scalable and cost-effective alternative to the traditional storage systems. 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 [2].

Tapes are attractive for long term data retention as their expected lifetime is higher than that of other types of media and their cost is considerably lower. Moreover, 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. This paper combines SIRF with cloud technology, as well as separately combines it with tape technology.

A core standard for digital preservation systems is the Open Archival Information System (OAIS)[4], an ISO standard since 2003 (ISO 14721:2003 OAIS). OAIS metadata can also include semantic metadata [3] to facilitate the preservation of schemas and ontological information. However, OAIS is a high-level reference model, which means it is flexible enough to be used in a wide variety of environments. More detailed steps and workflow stages need to be developed for the implementation of an OAIS based system. SIRF adds more detail to the metadata needed in the storage container.

SIRF uses cases and functional requirements were described in [4] along with the substantial differences from other formats. Our main contribution in this paper includes the definition of the SIRF format for long term storage containers. We define the SIRF catalog metadata, its categories and elements along with the rationale behind them. To show that SIRF can be combined with different types of underlying storage containers, we describe SIRF serialization for the cloud and SIRF serialization for tapes. We also provide some implementation overview of SIRF aspects in OpenStack cloud object storage[5] that is being examined in the context of the ForgetIT[6] European Union integrated research project.

The rest of this paper is organized as follows. In section 2, we discuss the business need of storage containers for long term retention. In section 3, we introduce the SIRF container format, its components and metadata. Section 4 defines the serialization for cloud and for tapes. Section 5 describes some aspects of experimental usage of SIRF in ForgetIT project for concise managed preservation of personal data and organizational web sites. In section 6, we review related work and conclude with a summary and some future work.

[4] http://public.ccsds.org/publications/archive/650x0m2.pdf

[5] http://www.openstack.org/software/openstack-storage

[6] http://www.forgetit-project.eu

2 Business Need for Long Term Retention

While no one wants to lose their digital content, the cost of maintaining integrity and access is significant, in both money and effort. And unlike paper based content, the lifespan of digital content can be very short unless if proactive steps are being taken to protect it. The use of a storage container format like SIRF adds little expense and greatly increases the sustainability of data. However, this is not adequate unless if the cost of preserving content is less than the (potential) cost of losing it.

In a business context, there are three major reasons why content is preserved. These are: to preserve history, to mitigate risk or meet a legal mandate, and for future value of information. One or more of these may apply, and the amount an entity is willing to spend will differ depending on how well these reasons are aligned with the business goals of an organization.

One of the main reasons why people and organizations preserve content is to preserve history. In the case of an individual, it may be photos, videos, and other content preserving one's life history. In a business context, libraries, national archives, historians, and others have a primary mission to preserve history.

Another often cited reasons for preserving data is for "risk mitigation", or in some cases for "legal mandate". These are closely related reasons because legal mandate is often looked at through the lens of legal risk. For example, an often cited legal mandate is in healthcare, where medical organizations are required to retain information for the lifetime of a patient. This seems like a difficult requirement, especially since records are often maintained in private doctors' offices and other places that may not exist 50 or 75 years into the future. Anecdotal evidence shows that medical records are not maintained that long. So, why is this happening? It is because records retention is expensive, and there are no penalties for losing information. That is not to say that doctors and hospitals don't try, rather they won't spend the necessary money.

Regarding future value of information, one obvious example is in the entertainment industry. Movies, TV shows, music, and other content can be re-sold and repurposed decades after its creation. This can result in many dollars in revenue. So not surprisingly, organizations like the Motion Picture Expert's Group are at the leading edge of digital preservation. Entertainment companies spend significant amounts of money retaining their content so that they will have it available to repurpose. However, this does not mean they can retain everything. With the advent of digital movie production, the amount of data that can be generated during the creation of a single film is immense. Therefore, even here where future value is tangible, some hard choices need to be made.

So, how does SIRF help? SIRF brings down the expense of preservation, because data can remain accessible even if the software that created the data no longer exists. SIRF reduces the complexity of logical and physical migration, making it easier for businesses to justify. By using SIRF today, it becomes possible to retain more information, and to retain information with a lower perceived future value. This is unlike proprietary and undocumented formats, which become useless soon after a business stops paying for support.

3 The SIRF Format

3.1 SIRF Components

Archivists and records managers of physical items such as documents, objects, 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 that group as a unit. Once assembled, an archivist will place the collection in a physical container (e.g. a file folder or a filing box of standard dimensions), and that container is attached with a label that gives an overview of the container content e.g. name and reference number, date, contents description, destroy date.

We propose an approach to digital content preservation that leverages the knowledge of the archival profession and helps archivists remain comfortable with the digital domain. We define a digital equivalent to the physical container - the archival box or file folder - that defines a collection, and which can be labeled with standard information in a defined format to allow retrieval when needed. SIRF is intended to be that equivalent - a storage container format for a set of (digital) preservation objects that includes 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.

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 cloud container, 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.

Figure 1 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 media and has an agreed defined name and a fixed size. It also includes the means to access the SIRF catalog (for example, the catalog's location).
- Preservation objects that contain the actual data to be preserved. An example preservation object can be the OAIS Archival Information Package (AIP). The container may include multiple versions of a preservation object and multiple copies of each version, but each specific preservation object is generally immutable.
- A catalog that is updateable and contains semantically enriched metadata needed to make the container and its preservation objects portable, accessible, and understandable into the future without relying on metadata external to the storage subsystem.

While traditional storage systems include only limited standardized metadata about each object, SIRF provides the semantically rich metadata needed for long term preservation and interpretation of information, and ensures its grouping with the data. This rich metadata is defined in the catalog in a logical format to allow its serialization for different storage technologies. We show its mapping to some of today's storage containers (cloud storage and tapes), but as new storage technologies become prevalent in the future, additional mappings will need to be defined.

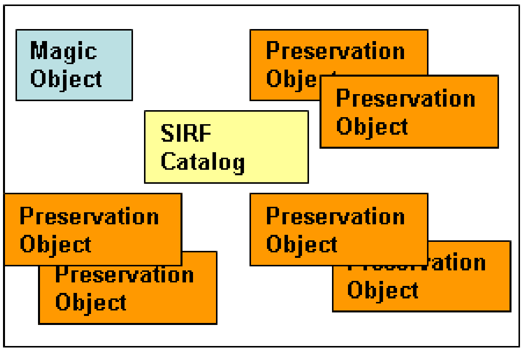

Fig. 1: SIRF Components

3.2 SIRF Catalog Metadata Schema

The SIRF catalog is an object that includes metadata about the preservation objects (POs) in the container and their schema and interelationships. It has a well-defined standardized format so it can be understandable in the future. The SIRF catalog is separated from the metadata contained in the POs themselves because a strict standardized format is difficult to impose on the POs that are generated by different applications and domains. Additionally, the SIRF catalog includes some metadata that is not included in the PO e.g. fixity value of the whole PO. Including this metadata within the PO changes the fixity value of the PO making this metadata inherently incorrect.

The SIRF catalog includes metadata related to the whole container as well as metadata related to each preservation object within the container. Both types of metadata are divided into categories, elements and attributes organized in a hierarchical representation. The full metadata definitions and the rationale behind them are defined in SIRF draft specification[7]. Here we provide some example categories for the whole container metadata in subsection 3.2.1 and for each preservation object within the container in subsection 3.2.2 below.

3.2.1 Container Information Metadata Schema. The metadata for the whole container includes the categories Specification, Container ID, State, and Container Provenance.

[7] http://www.snia.org/tech_activities/publicreview, to appear

The Specification category includes information about the specification used. As the specification may evolve over time and distinct storage containers may use different SIRF specifications, it's important to include the exact version of the specification in the SIRF catalog including specification ID and specification version.

The Container ID category includes the container unique identifier such as the tape ID for tape based storage containers or cloud container ID in case of cloud storage.

The State category is an indication of the progress of any activities that are to be carried out against a container. For example, if a container holds many preservation objects, state may indicate whether all of the objects intended for a container have been included or not. Or, state may indicate an in-process migration of a container. Multiple state entries are allowed in case if there are multiple pending activities.

The Container Provenance category is metadata describing the history of the information in a SIRF container (e.g., its origins, chain of custody, preservation actions and effects). The Provenance information may vary depending on the type of information being preserved or its intended audience and it may be large. Therefore, it is included in the catalog by reference, and the actual information is stored in another preservation object. The container provenance information stored in SIRF may be in the W3C-PROV format, or any other well known provenance format. Regardless of the perspective from which provenance metadata is derived, it is critical for understanding the container, its history, its context and meaning.

3.2.2 Object Information Metadata Schema. The metadata for each preservation object includes several categories; from which we'll describe here: Object IDs, Fixity, and Audit log.

The Object Identifiers (IDs) category is used to identify a PO and to link to other POs. Managing identifiers over the long term raises issues such as: how to ensure uniqueness of identifiers over long term, how to handle evolution of identifiers over time, how to ensure scalability of identifiers.

SIRF helps to address these issues by enabling redundancy in identifiers and registering the evolution (genealogy) of POs. Hence, a PO in a SIRF container can have multiple identifiers as redundant identifiers. This increases the chances that at least one of the identifiers will survive for the long term. Nevertheless, at any time, at least one of the identifiers should be persistent and unique.

Fixity is used to demonstrate that the content information has not been altered in an undocumented or unauthorized manner. The fixity information can be seen as an integrity check value. Fixity is sometimes computed via simple cheap functions such as a CRC, or it can include a stronger and more expensive (in execution time and space) cryptographic hash function such as MD5 or SHA-512. No matter how strong the fixity computation functions are, they are likely to become obsolete in the far future when larger amounts of storage and stronger computing power are available. Thus, the preservation system should be allowed to update fixity functions in the future, as existing ones become obsolete. Consequently, the SIRF catalog allows for multiple fixity algorithms and values for a given PO.

The audit log category is provided as a place for preserving any important information about how an object has been accessed or modified. The extent and contents of an audit log depend on the needs of the specific preservation data store and its use case. Distinct domains have different audit logs regulations e.g., SEC is for the US financial market domain, FDA is for the US medical domain. In SIRF, audit logs are stored in the catalog as links to preservation objects.

3.3 SIRF Container Security Guidelines

Some of the legal mandates for information retention also incorporate requirements for privacy and access protection. Where such security-based requirements exist, they add another level of complexity to long-term retention of the SIRF container. Much of this additional complexity results from the fact that the security-based requirements tend to mitigate against other retention requirements. For instance, while retention generally seeks to make information widely available and usable, security tends to restrict access to ensure that information privacy is maintained.

Information security also adds significantly to the amount of metadata that must be maintained within the container to ensure future usability of the information. Most obvious is the need to identify the encryption scheme used, and the need to maintain information about the different types of access that should be granted to the information. All access information needs to be based on the definition of abstract roles rather than specific people because, given the time periods being addressed by long-term retention, people will change job functions, organizations will grow, merge, or disappear, and uses for the information may significantly alter. A long-term retention system must be able to continuously add new users and associate them with existing roles, and change the roles assigned to existing users.

The management of keying information, whether related to the encryption of information or to the authentication of the roles assigned to specific users, presents a specific challenge in terms of long-term retention. Clearly such information cannot directly be located within the container itself, but sufficient metadata must be included in the container to allow the keying information to be located, validated, and verified.

ISO/IEC 27040 draft is being created to address the security of both local and cloud-based security systems. It emphasizes that there are integrity, authentication, and privacy threats that are particular to long-term storage systems. It also notes that the long lifetime of information within such systems enables attacks that require a large amount of access to the information but which can be disguised as many small requests over an extended period of time. It highlights the importance of maintaining a log of attack attempts, compromises, and system and user changes, and notes that such a log must also be maintained for the long-term. In the current version of SIRF, we support some initial security guidelines via e.g., the Fixity and the Audit Log categories.

4 SIRF Serialization for Cloud and for Tape

The SIRF serialization for cloud/tape specifies how a cloud container or a tape container becomes SIRF-compliant. A SIRF-compliant cloud container or tape container enables future's cloud/tape clients to "understand" containers created by today's cloud/tape clients even though the properties of the future client is unknown today. By "understand", we mean we can identify the preservation objects in the container, the packaging format of each object, its fixity values, etc. (as defined in the SIRF catalog).

For the concrete serialization we chose specific standard based storage containers. For the cloud, we chose CDMI[8] and OpenStack object storage while for tapes we chose LTFS[9] based tapes. No single technology will be usable over the time spans mandated by current digital preservation needs. SNIA CDMI and LTFS technologies are among best current choices, but 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, and it can be serialized for future technologies as they emerge.

For the serialization step, we classify the preservation objects as either simple preservation object or composite preservation object. A simple PO contains just one element and is mapped to one object in the CDMI cloud or one file in the LTFS tape. A simple PO can be for example a jpg photo or a tar file. A composite PO contains several elements and a manifest that combines the elements. The composite PO is mapped to several objects in the CDMI cloud or a number of files in the LTFS tape.

4.1 Serialization For Cloud Storage

The Cloud Data Management Interface (CDMI) is an ISO/IEC 17826:2012 standard created by SNIA that defines an interoperable format for moving data and associated metadata between cloud providers. CDMI has several implementations including an open source implementation for OpenStack Swift[10] cloud storage.

A CDMI cloud container can be qualified as a SIRF container when:

- The SIRF magic object is mapped to the CDMI container metadata.
- The SIRF catalog is an object in the CDMI container formatted in JSON (self-describing) that includes one containerInformation section and multiple objectInformation sections - one for each PO within the container (self-contained). This object should be indexed (if possible). There is a CDMI extension to support indexing with object granularity.

[8] Cloud Data Management Interface - http://www.snia.org/cdmi

[9] Linear Tape File System - http://www.snia.org/ltfs

[10] Swift - https://wiki.openstack.org/wiki/Swift

- A SIRF PO that is a simple object (contains one element) is mapped to a CDMI data object.
- A SIRF PO that is a composite object is mapped to a set of data objects (one for each element) and a manifest data object that includes information about the elements.

The interface to the SIRF-compliant CDMI container is the ordinary CDMI Application Program Interface (CDMI API). In addition, the CDMI API can be used to store and access the various preservation objects and the catalog object.

For example, figure 2 depicts a CDMI container named ”Patient Container” that is SIRF-compliant and includes medical encounters and images for the patient. Assume each encounter is a simple preservation object; each image is a composite preservation object; and since the container is SIRF-compliant, it also includes a catalog object.

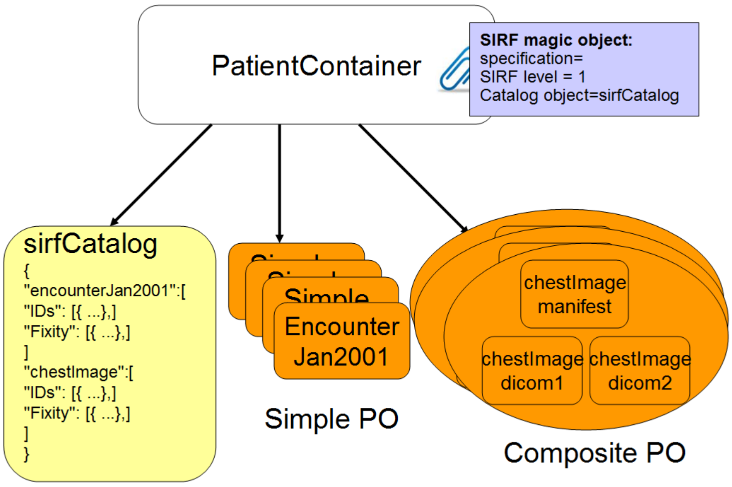

Fig. 2: SIRF Seralization for CDMI Example

4.2 Serialization For Tapes

The Linear Tape File System (LTFS) format specification defines LTFS Volumes. An LTFS Volume holds data files and corresponding metadata to completely describe the directory and file structures stored on the volume. Files can be written to, and read from, an LTFS Volume using standard POSIX file operations. The LTFS Volume includes an index in XML that contains metadata similar to information in disk-based file systems such as file name, dates, extent pointers, extended attributes, etc. LTFS is becoming the standard for linear tape and is being formalized through SNIA.

An LTFS volume is comprised of a pair of LTFS partitions: a data partition (DP) and an index partition (IP). Each partition contains a Label Construct followed by a Content Area. As depicted in figure 3, a LTFS tape container can be qualified also as a SIRF container when the volume format is as follows:

- The SIRF magic object is mapped to extended attributes of the LTFS index root directory.
- The SIRF catalog resides in the index partition and formatted in XML (self-describing) that includes one containerInformation section and multiple objectInformation sections - one for each PO within the container (self-contained). LTFS application has rules to indicate what to store in the index partition. That method can be used to indicate to store the SIRF catalog in the index partition. Alternatively, the index partition can include a reference to the SIRF catalog that will reside in the data partition.
- A preservation object (PO) is mapped to an LTFS file or set of files. In case the PO is a simple object composed of one element, it is mapped to a LTFS file. In case the PO is a composite object composed of several elements, it is mapped to a set of LTFS files (one for each element) and a manifest file that its content includes information about the elements.

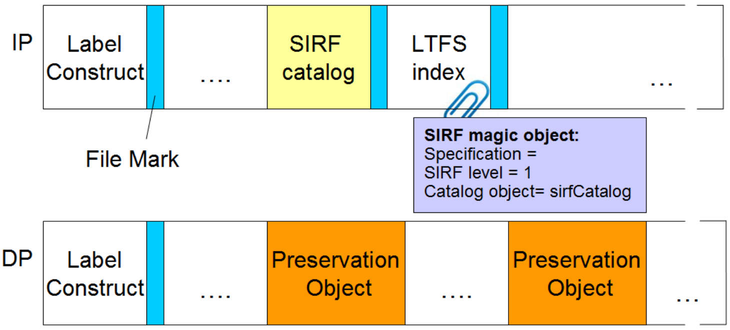

Fig. 3: SIRF Seralization for LTFS Volume

5 SIRF in ForgetIT

The European Union integrated project ForgetIT investigates ways for concise long term digital preservation and its adoption for personal data and organizational web sites. It combines three new concepts: managed digital forgetting inspired from human brain and cognitive psychology; smooth transition between data active use and its preservation; contextualized remembering keeping the archive understandable and useful.

The ForgetIT Preserve-or-Forget framework uses the DSpace open source as its preservation system, where the archival storage is Preservation DataStores (PDS) in the Cloud [2] that provides preservation-aware storage services based on the OAIS model. PDS includes the Preservation Engine and the Storlet Engine. The Preservation Engine transforms the logical OAIS functions and information objects into processes and physical storage objects. The Preservation Engine sometimes requires performing data-intensive computational tasks, such as transformation, migration, fixity checks, and data analysis. When the Preservation Engine requires performing such tasks, it uses storlets - computational modules running in a sandbox close to the data. Offloading OAIS-based functionality to the storage decreases probability of data loss, simplifies the applications and supports automation of preservation processes.

The Storlet Engine [5] provides the cloud storage with a capability to include storlets that run within the storage in a sandbox that provides isolation. It is plugged into a private cloud or object storage such as OpenStack Swift and provides a powerful extension mechanism that makes the storage flexible, customizable and extensible. By using storlets, the client benefits of reduced bandwidth (reduce the number of bytes transferred over the WAN), enhanced security (reduce exposure of sensitive data), cost saving (reduce infrastructure at the client side), and compliance support (improve provenance tracking).

PDS in ForgetIT implements some aspects of SIRF. It creates the various identifiers used for maintaining the evolution of POs, which can be stored in the Object IDs category in the SIRF catalog.

Regarding the fixity category in the SIRF catalog, PDS developed a fixity storlet that can compute multiple fixity values for each PO, and new hash functions can be uploaded to the storage as older ones become too weak or even obsolete.

While the ForgetIT POs are generated by different applications and domains (personal and organizational use cases), the SIRF catalog presents a standardized format that can be interpreted in the future.

6 Discussion and Conclusions

6.1 Related Work

Storage aspects of archiving and preservation systems have been the focus of a growing number of studies. You et al. [6] present PRESIDIO, a scalable archival storage system that efficiently stores diverse data. Adams et al. [7] studied scientific and historical archives, covering a mixture of purposes, media types, and access models. Based on this study, they identify areas for improving the efficiency and performance of archival storage systems.

Long-term preservation systems differ from traditional storage applications with respect to goals, characteristics, threats, and requirements. Baker et al. [8] examine these differences and suggest bit preservation guidelines and alternative architectural solutions that focus on replication across autonomous sites. Storer et al. [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.

Dappert and Enders [10] 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.

6.2 Conclusions and Future Work

Moving forward, digital content preservation will have many technical and cultural challenges. As digital technologies continue to replace physical ones, these challenges must be solved to prevent us from losing a generation of content.

SIRF, the Self-contained Information Retention Format, was developed to address the growing necessity to preserve digital information over long periods of time. SIRF does this by acting as the digital equivalent of an archivist's "box". SIRF preserves data and metadata as a single unit and provides a catalog containing the basic metadata needed to access and preserve content. This aids in the future understanding of data, and in the migration to new storage devices and formats.

We have shown that the SIRF can be serialized for a variety of storage technologies including LTFS based tape and CDMI cloud containers. This should provide a means for preserving information for the next years, and a vehicle for migrating to whatever new storage technologies become prevalent in the future.

In future work, we would like to improve support for the security guidelines developed in ISO/IEC 27040. Also, we would like to experiment SIRF in other projects and serialize it for additional storage containers.

Acknowledgments. This work was partially funded by the European Commission in the context of the FP7 ICT project ForgetIT (under grant no: 600826).

References

1. SNIA Long Term Retention (LTR) group. URL: `http://www.snia.org/ltr`
2. Rabinovici-Cohen, S., Marberg, J., Nagin, K., Pease, D.: PDS Cloud: Long Term Digital Preservation in the Cloud. In: IC2E 2013: Proceedings of the IEEE International Conference on Cloud Engineering, San Francisco, CA (March 2013)
3. Brunsmann, J.: Product Lifecycle Metadata Harmonization with the Future in OAIS Archives. In: DC 2011: Proceedings of the International Conference on Dublin Core and Metadata Applications, Hague, The Netherlands (2011)
4. Rabinovici-Cohen, S., Baker, M., Cummings, R., Fineberg, S., Marberg, J.: Towards SIRF: Self-contained Information Retention Format. In: SYSTOR 2011: Proceedings of the International Systems and Storage Conference, Israel (2011)
5. Rabinovici-Cohen, S., Henis, E., Marberg, J., Nagin, K.: Storlet Engine: Performing Computations in Cloud Storage. IBM Technical Report H-0320 (August 2014)
6. You, L., Pollack, K., Long, D., Gopinath, K.: PRESIDIO: A Framework for Efficient Archival Data Storage. ACM Transactions on Storage **7**(2) (July 2011)
7. Adams, I.F., Storer, M.W., Miller, E.L.: Analysis of Workload Behavior in Scientific and Historical Long-Term Data Repositories. TOS **8**(2) (2012)
8. Baker, M., Shah, M., Rosenthak, D., Roussopoulos, M., Maniatis, P., Giuli, T., Bungale, P.: A Fresh Look at the Reliability of Long-Term Digital Storage. In: Proceedings of the 1st ACM SIGOPS European Systems Conference. (2006)
9. Storer, M.W., Greenan, K.M., Miller, E.L., Voruganti, K.: POTSHARDS - A Secure, Recoverable, Long-Term Archival Storage System. TOS **5**(2) (2009)
10. Dappert, A., Enders, M.: Digital Perservation Metadata Standards. Information Standards Quarterly, Special Issue on Digital Preservation **22**(2) (2010) 4–12