

Long-term Data Retention: Challenges, Standards and Best Practices

Simona Rabinovici-Cohen, IBM Haifa
Sam Fineberg, Fineberg Consulting
Phillip Viana, IBM US
February 16 2017

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

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

Today's Presenters



Sam Fineberg
Co-chair, SNIA Long Term
Retention Technical Working
Group



Simona Rabinovici-Cohen
IBM Research - Haifa



Phillip Viana
Co-chair, SNIA Long Term
Retention Technical Working
Group
IBM US



SNIA-At-A-Glance



160

unique member
companies



3,500

active contributing
members



50,000

IT end users & storage
pros worldwide

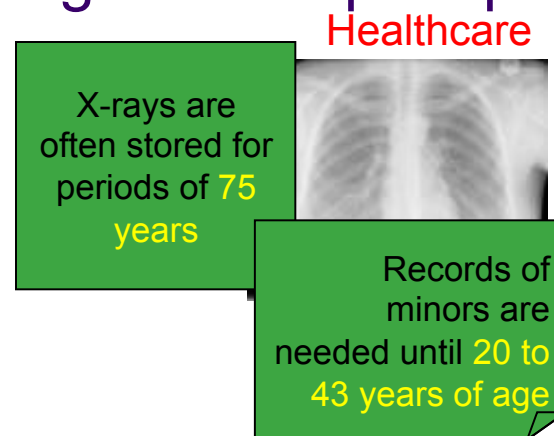
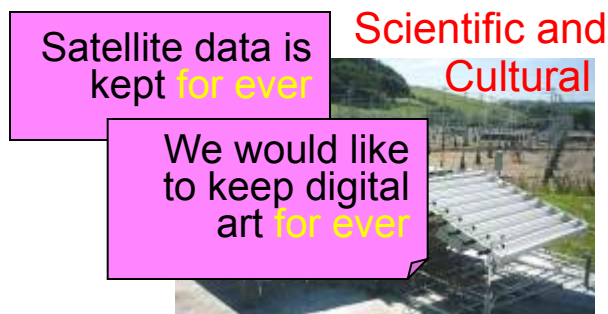
Learn more: snia.org/technical

 **@SNIA**

- **Introduction**
- **SNIA Long Term Retention technology**
 - ◆ Self-contained Information Retention Format (SIRF)
- **OpenSIRF**
- **Summary**

The Need for Digital Preservation of Big Data

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



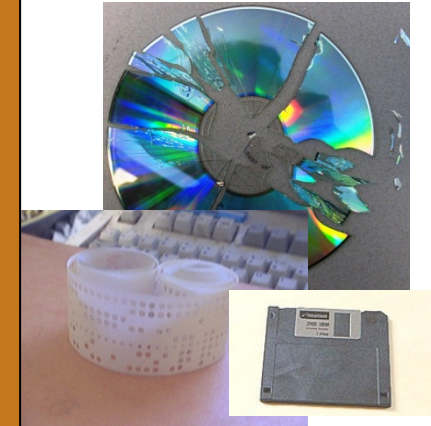
Goals of Digital Preservation

- Digital assets stored now should remain
 - ◆ Accessible
 - ◆ Undamaged
 - ◆ Usable
- For as long as desired – beyond the lifetime of
 - ◆ Any particular storage system
 - ◆ Any particular storage technology
- And at an *affordable cost*

Threats to long-term assets

- Large-scale disaster
- Human error
- Media faults

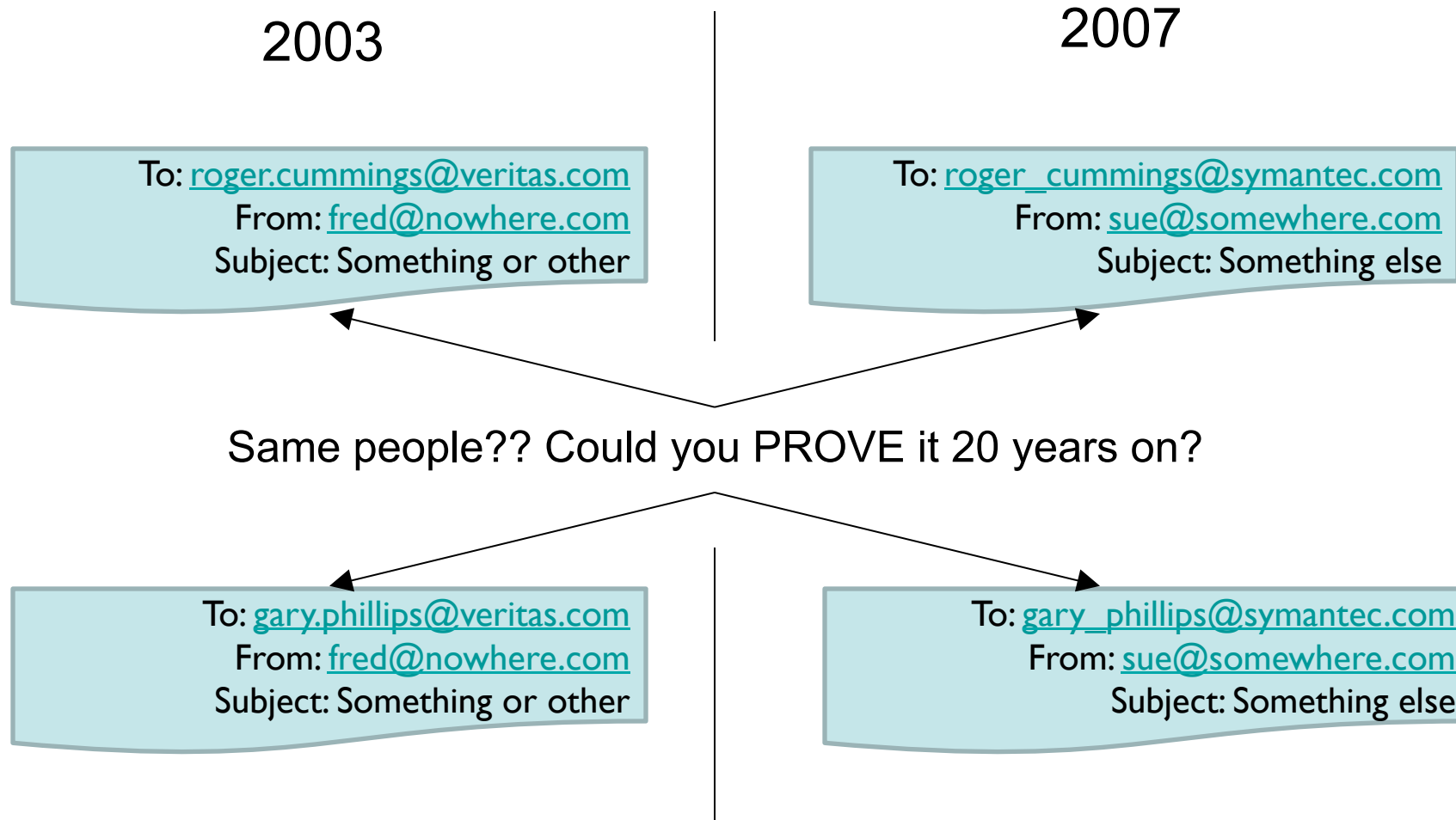
Long-term
content
suffers from
more threats
than short-
term content



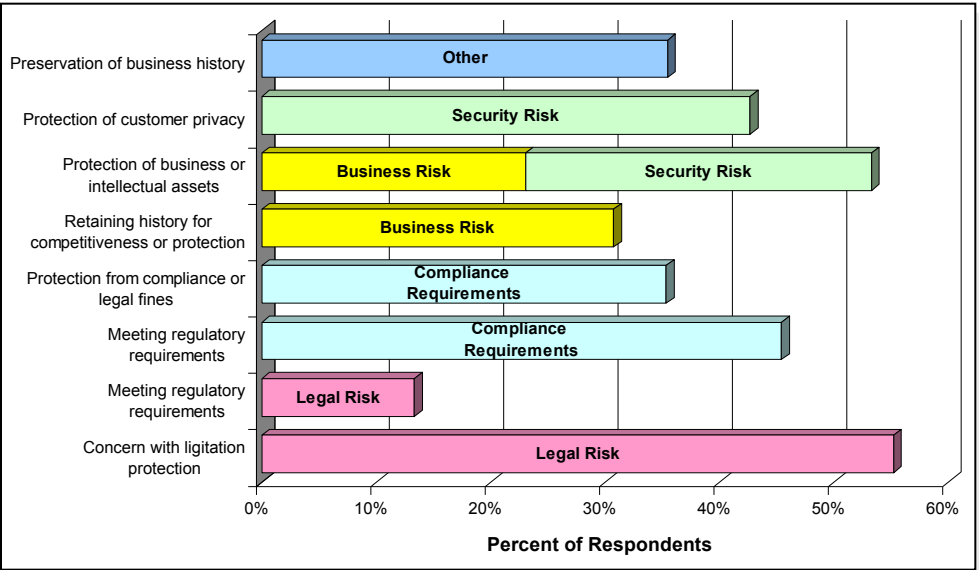
- Component faults
- Economic faults
- Attack
- Organizational faults

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

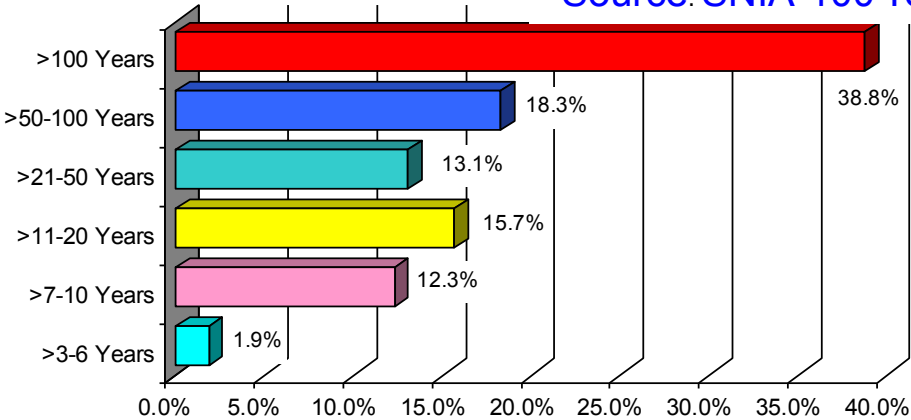
Real Life Example Problem



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



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



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

➤ Solutions are now becoming available

- ◆ Standards – OAIS, VERS, MoReq, ...
- ◆ Storage formats - SIRF, OpenAXF, PREMIS, BagIt....
- ◆ Software – Fedora, LOCKSS, DSPace, Arkivum, iRods, Rosetta,
- ◆ Cloud Services – Preservica, Duracloud, Chronopolis, Dternity, Glacier,

➤ But, their usage is still limited

- ◆ Primarily used in government agencies, libraries, and highly regulated industries

➤ Why

- ◆ Lack of education or understanding?
- ◆ Lack of need, will, funding, etc.? Lack of penalties?
- ◆ Short term focus?

100 Year Archive Survey 2017



- The SNIA LTR SIG and DPCO committee have recently begun an effort to develop a new 100 Year Archive survey.
- The goal of this new survey is to assess:
 - ◆ What has changed in 10 years, are the key tenants of the original survey still true?
 - ◆ Have business drivers changed? Have businesses raised or lowered the priority?
 - ◆ What preservation methods are being used? What systems, and how consumed?
 - ◆ Are users meeting their goals for data preservation? What challenges remain?
- To better define the surveys goals, target populations, and specific questions, SNIA is soliciting input
 - ◆ Please join the LinkedIn SNIA Archive Survey group:
 - ◆ <https://www.linkedin.com/groups/8590697>
 - ◆ Birds of a feather session at the FAST conference in Santa Clara, March 1

- Introduction
- **SNIA Long Term Retention technology**
 - ◆ Self-contained Information Retention Format (SIRF)
- OpenSIRF
- Summary

SIRF: Self-contained Information Retention Format



A specification by SNIA Long Term Retention (LTR) TWG

An Analogy

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

Photo courtesy Oregon State Archives



SIRF Properties

- ❑ SIRF is a logical data format of a **storage container** appropriate for long term storage of digital information
 - ❑ A storage container may comprise a logical or physical storage area considered as a unit.
 - ❑ Examples: a file system, a tape, a block device, a stream device, an object store, a data bucket in a cloud storage

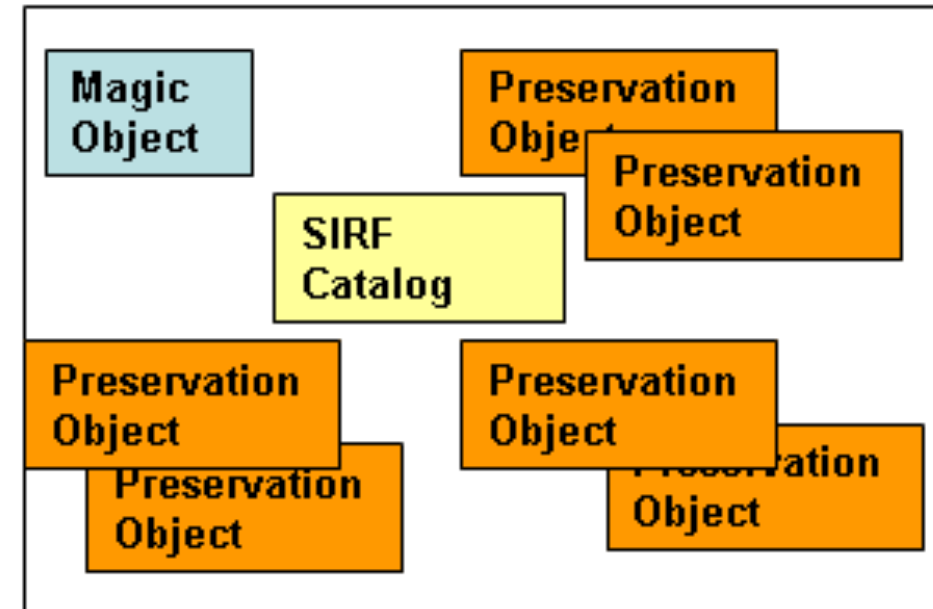
- ❑ Required Properties
 - ❑ **Self-describing** – can be interpreted by different systems
 - ❑ **Self-contained** – all data needed for the interpretation is in the container
 - ❑ **Extensible** – so it can meet future needs



SIRF Components

A SIRF container includes:

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



SIRF is inspired by the Open Archival Information System (OAIS) - ISO 14721:2003

SIRF Categories



The SIRF catalog includes metadata organized in a hierarchy of categories, elements and attributes. The categories are:

- ◆ Container information:
 - › Specification
 - › Container ID
 - › State
 - › Provenance
 - › Audit Log
- ◆ For each Preservation Object:
 - › Object IDs
 - › Related Objects
 - › Dates
 - › Packaging Format
 - › Fixity
 - › Retention
 - › Audit Log
 - › Extension

Storlets can use SIRF catalog metadata for preservation processes

- The Storlet Engine performs computation modules within the storage
- http://rd.springer.com/chapter/10.1007%2F978-3-319-15895-2_6
- [http://domino.research.ibm.com/library/cyberdig.nsf/papers/7233ABCCEC84F0BF85257D3100559FD6/\\$File/H-0320.pdf](http://domino.research.ibm.com/library/cyberdig.nsf/papers/7233ABCCEC84F0BF85257D3100559FD6/$File/H-0320.pdf)

PO Information – IDs Category



➤ Elements:

- ◆ **PO name (objectName)** – non unique identifier e.g. file name
- ◆ **PO version ID (objectVersionIdentifier)** – unique identifier that identifies the specific version of the PO
- ◆ **PO logical ID (objectLogicalIdentifier)** - a unique identifier that identifies the various versions that originate from the same ancestor
- ◆ **PO parent ID (objectParentIdentifier)** - a unique identifier that identifies the parent PO from which this PO version was created. Parent PO shares the same logical ID as the current PO, but has different version ID.

Goals of SIRF Serialization for Cloud/FS

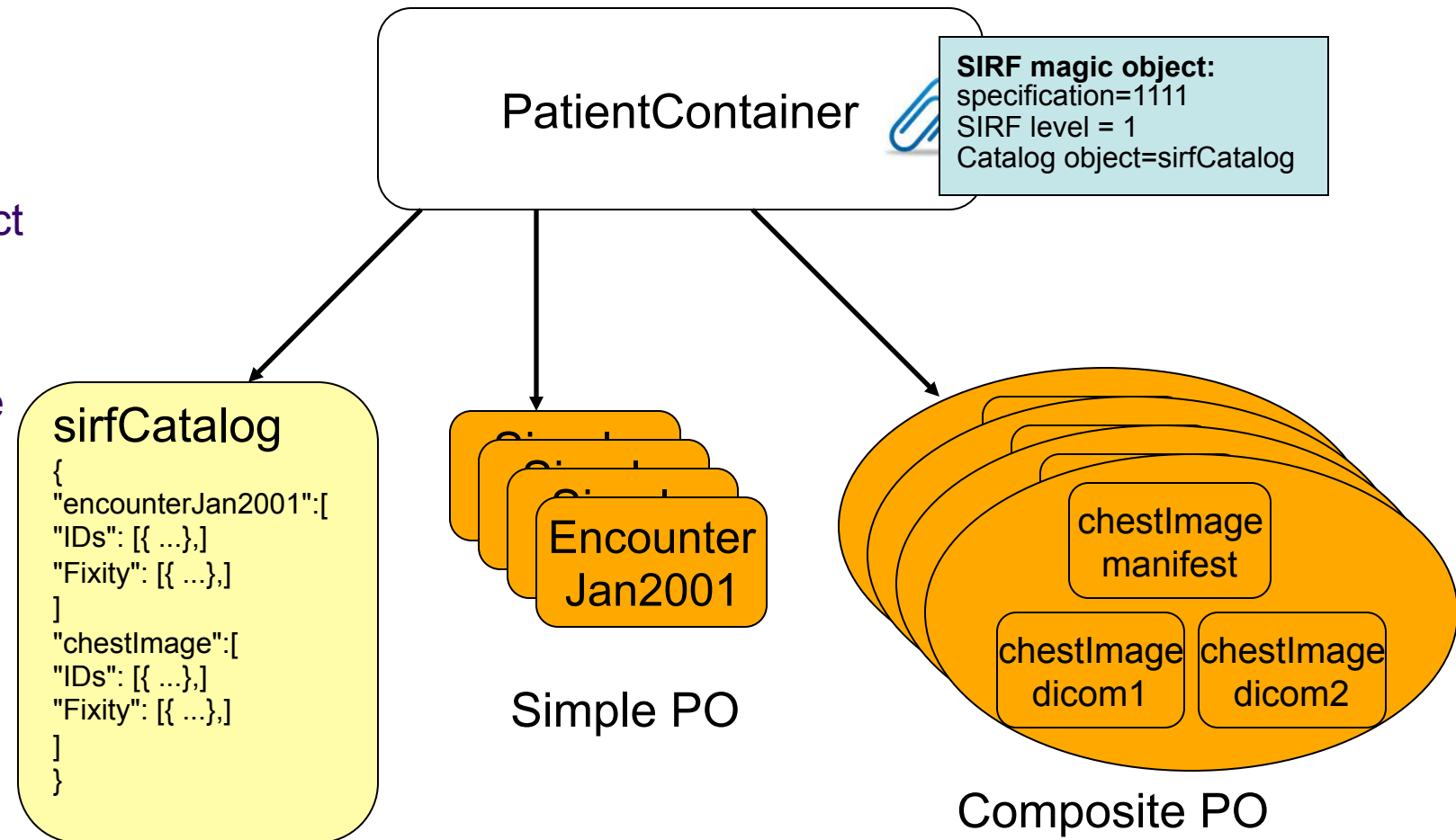


- SIRF serialization for Cloud/FS specifies how a SNIA Cloud Data Management Interface (CDMI) cloud container or Linear Tape File System (LTFS) tape also becomes SIRF-compliant
 - ◆ This can be generalized to any cloud container and file system container
 - ◆ OpenSIRF implemented serialization for OpenStack Swift container and regular file system logical volume

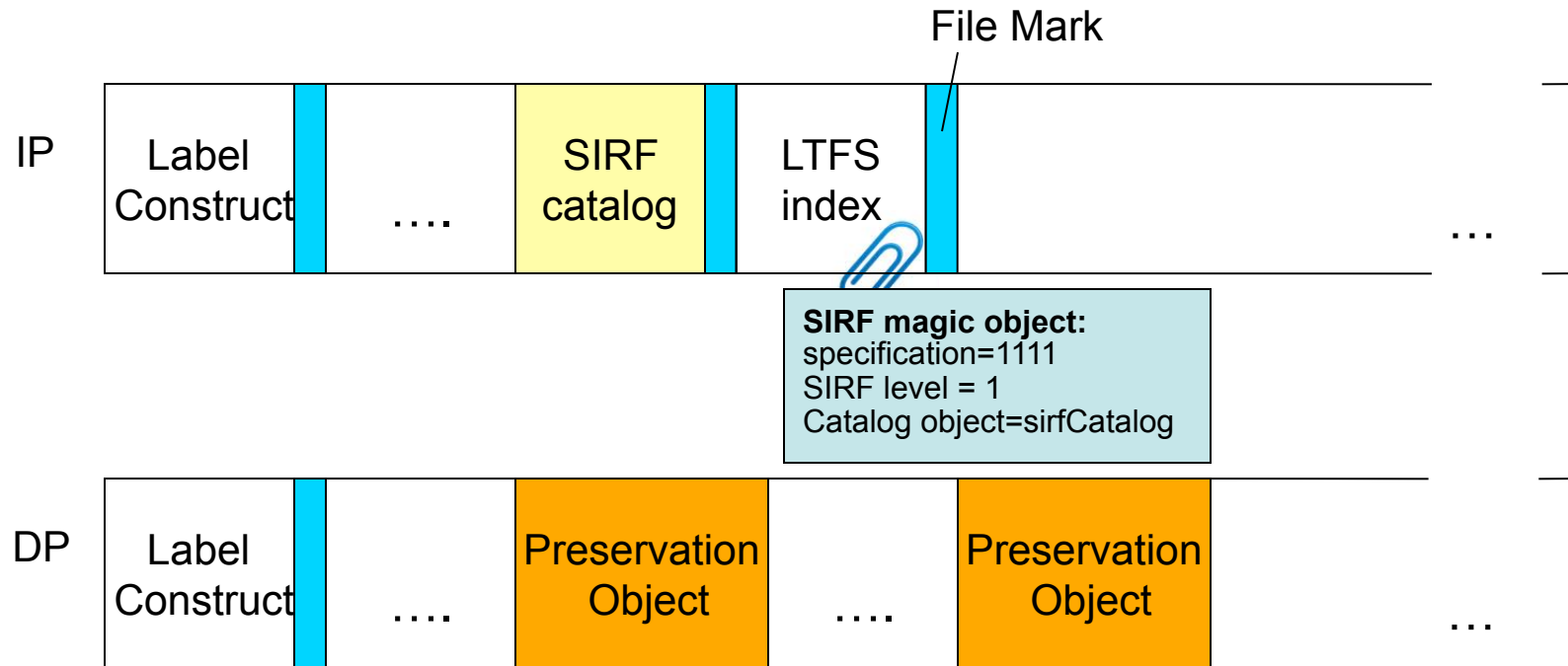
- A SIRF-compliant cloud or file system containers enable a future storage client to “understand” containers created by today’s storage client
 - ◆ The properties of the future client is unknown to us today
 - ◆ “understand” means identify the preservation objects in the container, the packaging format of each object, its fixities values, etc. (as defined in the SIRF catalog)

SIRF Serialization for Cloud

- SIRF magic object is mapped to the CDMI container metadata
- SIRF catalog is an object in the CDMI container formatted in JSON
- SIRF Simple/Composite PO is mapped to CDMI data object/set of data objects



SIRF Serialization for LTFS Tape








































- SIRF magic object is mapped to extended attributes of the “LTFS index” root directory
- SIRF catalog resides in the index partition and formatted in XML
- SIRF Simple/Composite PO is mapped to a LTFS file/set of files

- Introduction
- SNIA Long Term Retention technology
 - ◆ Self-contained Information Retention Format (SIRF)
- **OpenSIRF**
- Summary

- ❑ Open source reference implementation of SIRF
 - ❑ MIT license
 - ❑ Java
 - ❑ Available in GitHub: [opensirf.github.com](https://github.com/opensirf)
- ❑ Components
 - ❑ Core (Java classes that model SIRF)
 - ❑ Server (JAX-RS implementation – REST API)
- ❑ Supports
 - ❑ Regular file systems
 - ❑ OpenStack Swift

OpenSIRF Core classes

- ▼  org.opensirf.audit
 - ▷  AuditLogReference.java
 - ▷  ContainerAuditLog.java
 - ▷  ContainerAuditLogReference.java
 - ▷  PreservationObjectAuditLog.java
- ▼  org.opensirf.catalog
 - ▷  IndexedObjectInformationSet.java
 - ▷  SIRFCatalog.java
 - ▷  SIRFCatalogMarshaller.java
- ▼  org.opensirf.container
 - ▷  ContainerIdentifier.java
 - ▷  ContainerInformation.java
 - ▷  ContainerProvenanceReference.java
 - ▷  ContainerSpecification.java
 - ▷  MagicObject.java
 - ▷  Provenance.java
 - ▷  ProvenanceInformation.java
 - ▷  SIRFContainer.java
 - ▷  State.java
- ▼  org.opensirf.obj
 - ▷  ContainerIdentifierElement.java
 - ▷  DigestInformation.java
 - ▷  Extension.java
 - ▷  ExtensionPair.java
 - ▷  FixityInformation.java
 - ▷  ObjectIdentifierElement.java
 - ▷  PackagingFormat.java
 - ▷  PreservationObjectIdentifier.java
 - ▷  PreservationObjectInformation.java
 - ▷  PreservationObjectLogicalIdentifier.java
 - ▷  PreservationObjectName.java
 - ▷  PreservationObjectParentIdentifier.java
 - ▷  PreservationObjectVersionIdentifier.java
 - ▷  ReferenceElement.java
 - ▷  RelatedObjectReference.java
 - ▷  RelatedObjects.java
 - ▷  Retention.java

- ❑ Easy-to-install development environment
 - ❑ Requires Vagrant and Virtualbox
 - ❑ 3 virtual machines:
 - OpenSIRF Server (JAX-RS running on Tomcat)
 - File system storage container
 - OpenStack Swift storage container
 - ❑ Chef cookbook configures the connectivity between servers

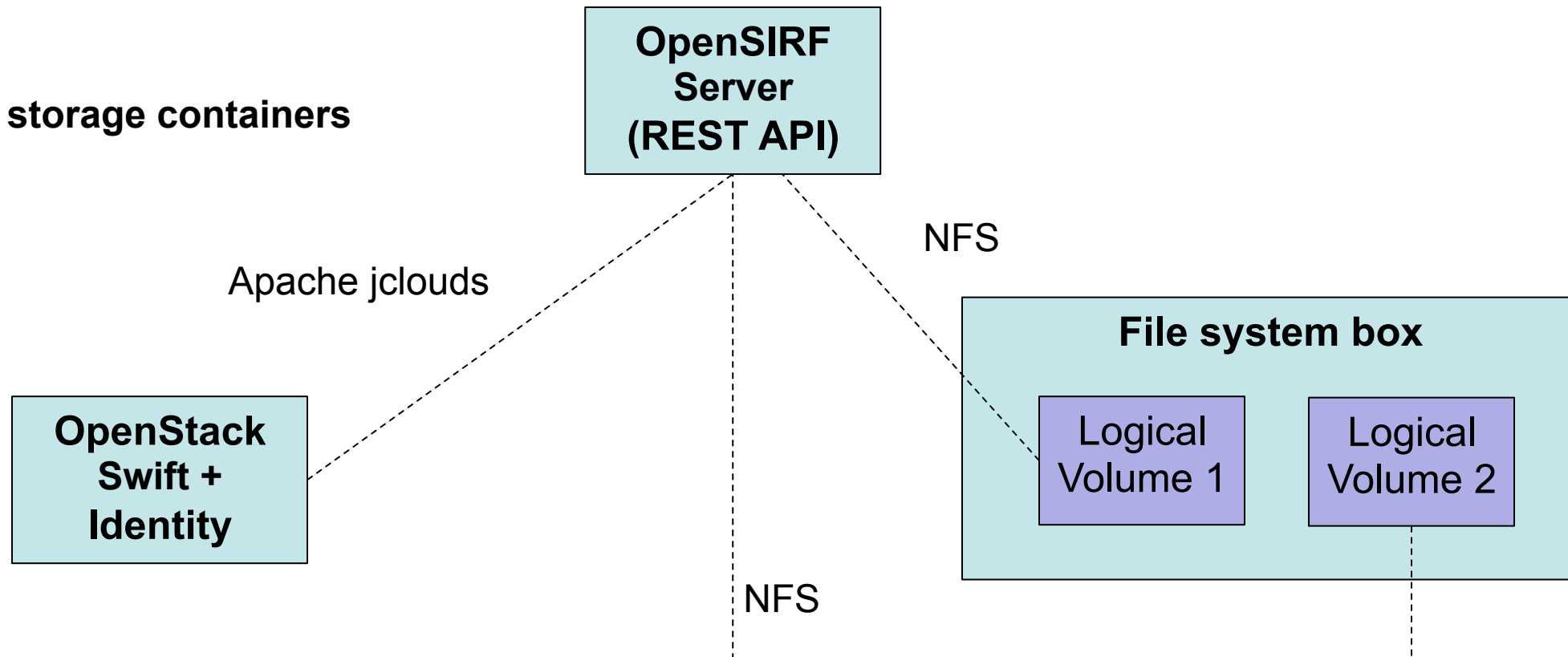
```
wget https://git.io/vDKP8 -O install.sh && bash install.sh
```

A red arrow originates from the text below and points upwards and to the left, ending at the URL in the command line above.

[https://raw.githubusercontent.com/
OpenSIRF/opensirf-server/develop/install-
dev.sh](https://raw.githubusercontent.com/OpenSIRF/opensirf-server/develop/install-dev.sh)


```
wget https://git.io/vDKP8 -O install.sh && bash install.sh
```

Multiple storage containers



- ❑ Abstract driver factory
 - ❑ SwiftDriver (extends ISirfDriver)
 - ❑ FilesystemDriver (extends ISirfDriver)
- ❑ Target storage container is decided in runtime according to distribution policy

```
public interface ISirfDriver {  
    public void createContainerAndMagicObject(String containerName);  
    public MagicObject containerMetadata(String containerName);  
    public InputStream getFileInputStream(String container, String filename) throws IOException;  
    public void uploadObjectFromString(String containerName, String fileName, String content);  
    public void uploadObjectFromByteArray(String containerName, String fileName, byte[] b);  
    public void deleteContainer(String containerName);  
    public void deleteObject(String containerName, String objectName);  
}
```


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


```
[phil@oc0364286225 ~]$ curl -i -X GET http://devsirfserver:8088/sirf/config; echo
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 515
Date: Tue, 14 Feb 2017 23:36:26 GMT

{"containerConfiguration":{"containerName":"sirfContainer","driver":"multi","endpoint":
"localhost","distributionPolicy":"evenlyFree","subconfigurations":[{"mountPoint":"lv1",
"containerName":"lv1","driver":"fs","endpoint":"devsirffs"}, {"mountPoint":"lv2","contai
nerName":"lv2","driver":"fs","endpoint":"devsirffs"}, {"identity":"sirf:sirfadmin","cred
ential":"100years","provider":"openstack-swift","region":"RegionOne","containerName":"s
wiftContainer1","driver":"swift","endpoint":"http://devsirfswift:5000/v2.0/"}]}}
```


A screenshot of a terminal window with a black background and white text. The text shows a successful PUT request to a devsirfserver. The output includes the HTTP status 201 Created, the server version Apache-Coyote/1.1, the location of the resource, and the date and time of the request.

```
[phil@oc0364286225 ~]$ curl -i -X PUT http://devsirfserver:8088/sirf/container/myContainer
HTTP/1.1 201 Created
Server: Apache-Coyote/1.1
Location: http://devsirfserver:8088/sirf/container/myContainer
Content-Length: 0
Date: Tue, 14 Feb 2017 23:38:01 GMT
```



```
[phil@oc0364286225 ~]$ curl -i -X GET http://devsirfserver:8088/sirf/container/myContainer/catalog
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Content-Length: 1551
Date: Tue, 14 Feb 2017 23:39:08 GMT
```

```
{
  "catalogId": "catalog.json",
  "containerInformation": {
    "containerSpecification": {
      "containerSpecificationIdentifier": "SIRF-1.0",
      "containerSpecificationSirfLevel": "1",
      "containerSpecificationVersion": "1.0"
    },
    "containerIdentifier": {
      "containerIdentifierLocale": "en",
      "containerIdentifierType": "containerIdentifier",
      "containerIdentifierValue": "myContainer"
    },
    "containerState": {
      "containerStateType": "ready",
      "containerStateValue": "true"
    },
    "containerProvenanceReference": {
      "referenceRole": "Provenance",
      "referenceType": "internal",
      "referenceValue": "provenance.po.json"
    },
    "containerAuditLog": [],
    "objectsSet": {
      "objectInformation": [
        {
          "objectIdentifiers": [
            {
              "objectName": [
                {
                  "objectIdentifierLocale": "en",
                  "objectIdentifierType": "name",
                  "objectIdentifierValue": "provenance.po.json"
                }
              ],
              "objectLogicalIdentifier": {
                "objectIdentifierLocale": "en",
                "objectIdentifierType": "logicalIdentifier",
                "objectIdentifierValue": "provenance.po.json"
              },
              "objectParentIdentifier": {
                "objectIdentifierLocale": "en",
                "objectIdentifierType": "parentIdentifier",
                "objectIdentifierValue": "null"
              },
              "objectVersionIdentifier": {
                "objectIdentifierLocale": "en",
                "objectIdentifierType": "versionIdentifier",
                "objectIdentifierValue": "provenance.po.json"
              }
            }
          ],
          "objectCreationDate": "2017-02-14T23:38Z",
          "objectLastModifiedDate": "2017-02-14T23:38Z",
          "objectLastAccessedDate": "2017-02-14T23:38Z",
          "objectRelatedObjects": [],
          "packagingFormat": {
            "packagingFormatName": "none"
          },
          "objectRetention": {
            "retentionType": "time_period",
            "retentionValue": "forever"
          },
          "objectAuditLog": [],
          "objectExtension": [],
          "versionIdentifierUUID": "provenance.po.json"
        }
      ]
    }
  }
}
```



```
[phil@oc0364286225 ~]$ cat aaa
jfdsaofeifjdsalkfndsalkfjdsalkfjdsalkfjdsjakfjdoq
[phil@oc0364286225 ~]$ curl -i -X POST -H "Content-type:multipart/form-data" -F inputstream=@/home/p
hil/aaa -F poi="@poi.json;type=application/json" http://devsirfserver:8088/sirf/container/myContaine
r/po
HTTP/1.1 100 Continue

HTTP/1.1 201 Created
Server: Apache-Coyote/1.1
Location: http://devsirfserver:8088/sirf/container/myContainer/philPo1
Content-Length: 0
Date: Tue, 14 Feb 2017 23:46:06 GMT
```



```
[phil@oc0364286225 ~]$ curl -i -X GET http://devsirfserver:8088/sirf/container/myContainer/philPol/d
ata
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
content-disposition: attachment;filename=philPol
Content-Type: application/octet-stream
Content-Length: 48
Date: Tue, 14 Feb 2017 23:49:52 GMT

jfdsaoifeifjdsalkfndsalkfjdslkjfdskalfdsjakfjdoq
```


❑ Resources

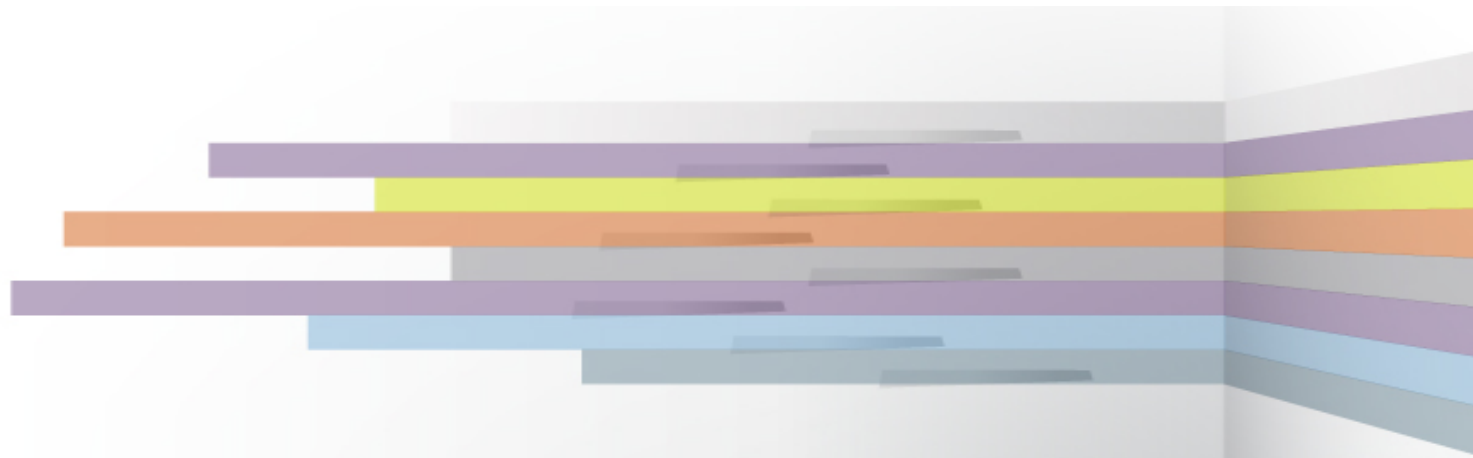
- ❑ Google Group: <https://groups.google.com/forum/#!forum/opensirf>
- ❑ Source code: <https://github.com/opensirf>
- ❑ Documentation: <https://github.com/OpenSIRF/opensirf-doc>
- ❑ Vagrant boxes: <https://atlas.hashicorp.com/opensirf/>

- Introduction
- SNIA Long Term Retention technology
 - ◆ Self-contained Information Retention Format (SIRF)
- OpenSIRF
- **Summary**

- Need to retain not only information of interest but ALL other information to make it fully usable in future
 - ◆ Put it all in the SIRF “digital box”, preserve that as a unit
 - ◆ SIRF includes metadata about the storage container, to help “understand” the contents of the container in the future
- No single technology will be usable over the timespans mandated by current digital preservation needs
 - ◆ SIRF provides a vehicle for collecting all of the information that will be needed to transition to new technologies in the future
 - › SIRF can be serialized for the future technologies as they come

For further information

- SIRF specification
http://www.snia.org/tech_activities/standards/curr_standards/sirf
- Self-contained Information Retention Format For Future Semantic Interoperability
 - ◆ <http://ceur-ws.org/Vol-1306/paper2.pdf>
- More information on SIRF & SNIA LTR activities (including these slides)
 - ◆ <http://www.snia.org/ltr>
- OpenSIRF is available at:
 - ◆ <http://github.com/opensirf>
- LinkedIn SNIA Archive Survey group:
 - ◆ <https://www.linkedin.com/groups/8590697>
- FAST Conference BoF session March 1 in Santa Clara
<https://www.usenix.org/conference/fast17/bofs#snia>



Thank You for Attending

Please remember to rate this webcast

Slides are available at <http://snia.org/ltr>