ANALYTICS & BIG DATA SUMMIT

LONG TERM RETENTION OF BIG DATA

David Pease
Simona Rabinovici-Cohen
IBM Research

Outline

- Introduction and Challenges
- LTFS: Linear Tape File Systems
- SIRF: Self-contained Information Retention Format
- EU ENSURE: Enabling kNowledge, Sustainability, Usability and Recovery for Economic Value
- LTFS and SIRF
- Summary

Big Data for Long Term

- 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:

Healthcare

X-rays are often stored for periods of 75 years

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

M&E

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

Satellite data is kept for ever

Scientific and Cultural

We would like to keep digital art for ever

The Long Term Retention Challenge

ANALYTICS & BIG DATA SUMMIT

These documents were created by pre-digital societies.
The media and information content are still interpretable.

Mayan Glyph,

Palenque ***~630AD.***

Dead Sea Scroll, ~70AD.
Media: Copper.
Language: Hebrew.

This information was created a few years ago.

Will the media last for 20 years?

Will it be possible to access, interpret and present the data in 20 years? 50? 100?

Challenges in the Long Term Retention of Big Data

- Provide economically scalable storage systems that
  - Efficiently store and preserve big data archives
  - Enable far future search, access, and analytics

- There are many financial and practical reasons to prefer tape storage for such big data archives
  - Longer media life expectancy
  - Lower cost per PB over time
  - Greater bandwidth for transfers and migrations

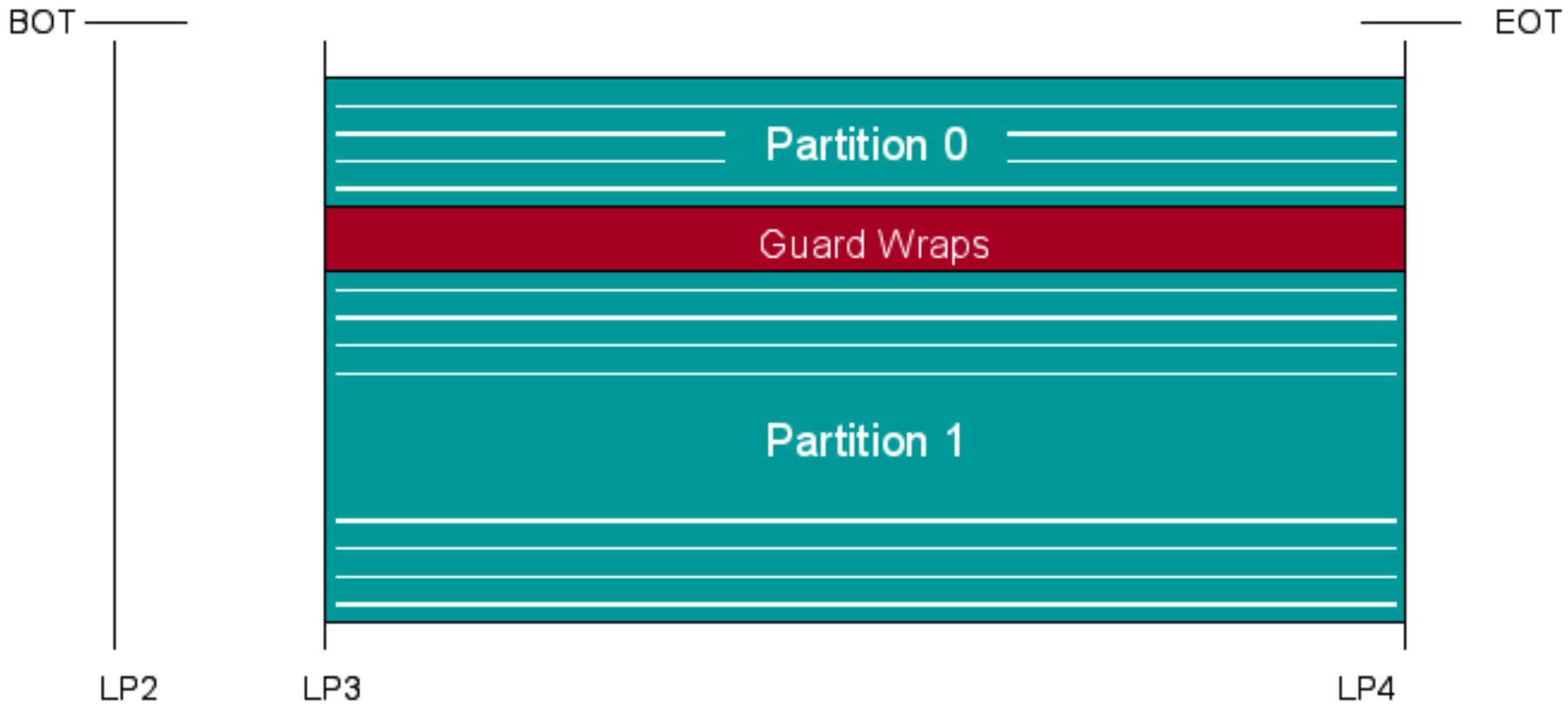
Outline

- Introduction and Challenges
- LTFS: Linear Tape File Systems
- SIRF: Self-contained Information Retention Format
- EU ENSURE: Enabling kNowledge, Sustainability, Usability and Recovery for Economic Value
- LTFS and SIRF
- Summary

Linear Tape Open (LTO) Tape

- LTO Consortium
  - Defines an industry standard for tape drives and media
    - IBM, HP, Quantum, many media manufacturers
- LTO Tape
  - Serpentine recording, shingled writing
  - Block-addressable
  - Essentially an append-only media
- LTO Generation 5 (LTO-5)
  - Released April 2010
  - 1.5 terabytes per cartridge (uncompressed)
  - 140 MB/sec streaming data rate
  - Dual-partition capability

Dual-Partition Tape – logical view



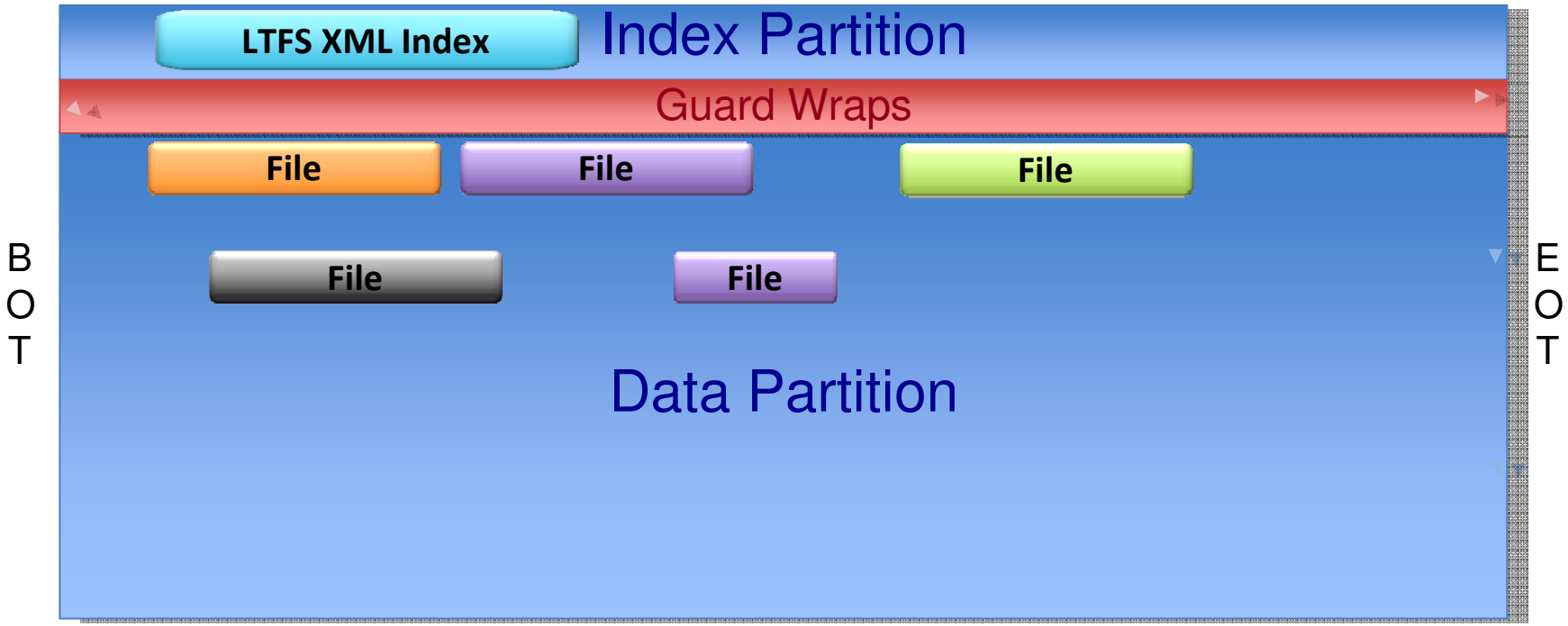
What is LTFS?

- A file system implemented on dual-partition linear tape
  - Makes tape look and work like any removable media (e.g., USB drive, removable disk)
    - Files and directories show up on desktop, directory listings, etc.
    - Drag-and-drop files to/from tape, double-click to open
    - Run any application written to use disk files
  - Includes file- and directory-level user metadata
    - Unlimited extended attributes
  - Supports automated libraries as well as stand-alone drives
    - In library mode, allows listing contents and searching of all volumes in library without mounting tapes
  - IBM single-drive implementations released as open source
    - Linux and Mac OS X versions released as Open Source
    - Windows version freely downloadable
    - www-03.ibm.com/systems/storage/tape/ltfs/index.html

What is LTFS?

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

Logical View of LTFS Volume



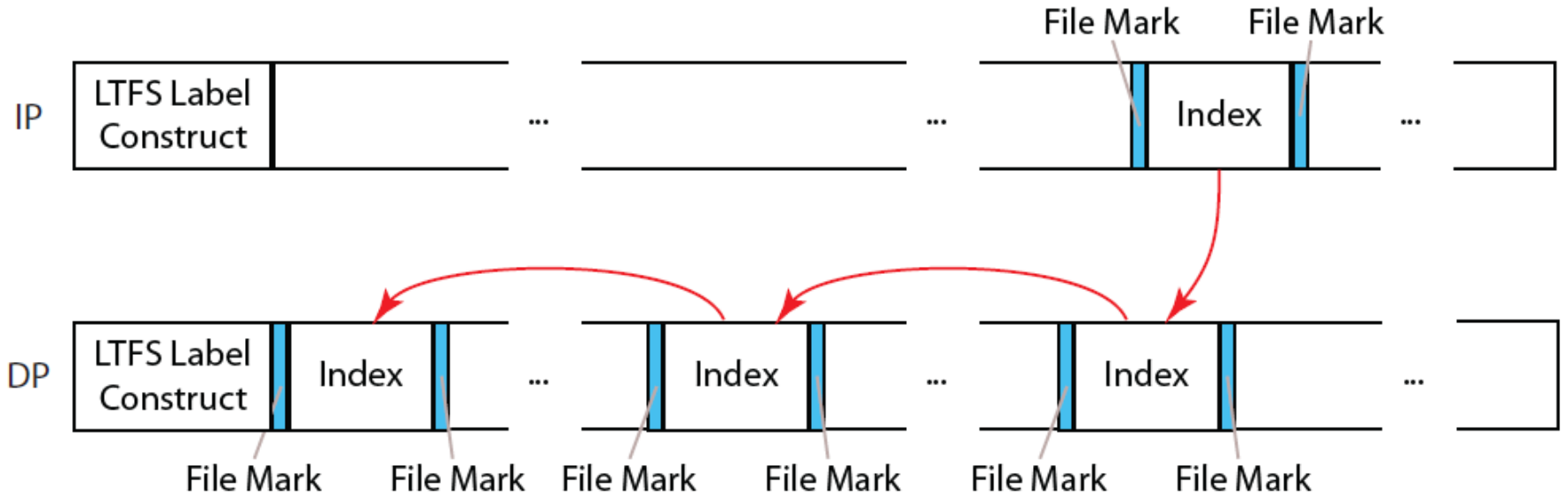
XML Index Schema

- Similar to information in disk-based file system
  - Files
    - Name, dates, extent pointers, extended attrbutes, etc.
  - Directories
- Designed to be simple, cross-platform
  - Tags and values easy to read, “human” format
  - No platform-specific data
    - Supports Unix/Linux, MacOS, and Windows
- Format becoming the standard for linear tape
  - Formal standardization through SNIA
- Format specfication on LTO Consortium site:
  - www.trustlto.com/LTFS_Format_To Print.pdf

Sample XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<index version="0.9">
    <creator>IBM LTFS 0.20 - Linux - ltfs</creator>
    <volumeuuid>9710d610-5598-442a-8129-48d87824584b</volumeuuid>
    <generationnumber>3</generationnumber>
    <directory>
        <name>LTFS Volume Name</name>
        <creationtime>2010-01-28 19:39:50.715656751 UTC</creationtime>
        <modifytime>2010-01-28 19:39:55.231540960 UTC</modifytime>
        <accesstime>2010-01-28 19:39:50.715656751 UTC</accesstime>
        <contents>
            <directory>
                <name>directory1</name>
                <contents>
                    <file>
                        <name>binary_file.bin</name>
                        <length>10485760</length>
                        <extentinfo>
                            <extent>
                                <partition>b</partition>
                                <startblock>8</startblock>
                                <byteoffset>0</byteoffset>
                                <bytecount>720000</bytecount>
                            </extent>
                            <extent>
                                <partition>b</partition>
                                <startblock>18</startblock>
                                <byteoffset>0</byteoffset>
                                <bytecount>9765760</bytecount>
                            </extent>
                        </extentinfo>
                        <extendedattributes>
                            <xattr>
                                <key>uservalue</key>
                                <value>fred</value>
                            </xattr>
                        </extendedattributes>
                    </file>
                    <file>
                        <name>read_only_file</name>
                        <length>0</length>
                        <readonly/>
                    </file>
                </contents>
            </directory>
        </contents>
    </directory>
</index>
```

Index Arrangement on LTFS Tape

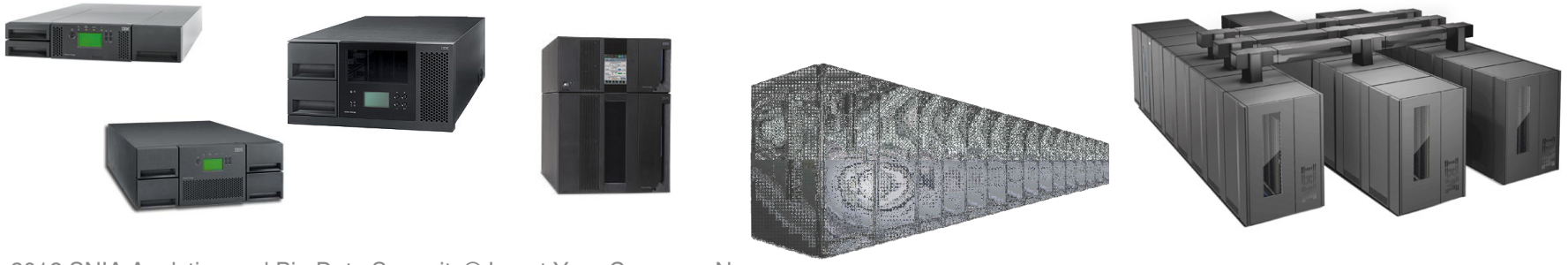


LTFS in Single Drive Mode

- Shows up like any standard (i.e., disk) file system
  - Directories
  - Files
- Tape contains File/Directory Index in Index Partition
  - XML schema
    - Keeps multiple “generations” (older versions of XML schema)
  - Data files written to Data Partition (usually)
- Small data files can optionally be written to Index Partition
  - Quick access, can be cached at mount time
- Tape index “forgotten” at unmount

LTFS in Library Mode (LTFS LE)

- Mount Library, not Drive
- LTFS Caches Index of each tape read/written
  - Each volume shows as separate file system folder/directory
  - After mount, all tape directories are viewable, searchable
    - Without mounting any tape
    - LTFS drives automation to mount tape on file read/write
- LTFS can recognize when tape leaves, reenters library
  - Performs consistency check to see if tape index has changed



Outline

- Introduction and Challenges
- LTFS: Linear Tape File Systems
- SIRF: Self-contained Information Retention Format
- EU ENSURE: Enabling kNowledge, Sustainability, Usability and Recovery for Economic Value
- LTFS and SIRF
- Summary

Self-contained Information Retention Format (SIRF)

Being developed by SNIA Long Term Retention (LTR) TWG

An Analogy

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

- SIRF is the digital equivalent
  - Logical container for a set of (digital) preservation objects and a catalog
  - The SIRF catalog contains metadata related to the entire contents of the container as well as to the individual objects
  - SIRF standardizes the information in the catalog

Photo courtesy Oregon State Archives

SIRF Properties

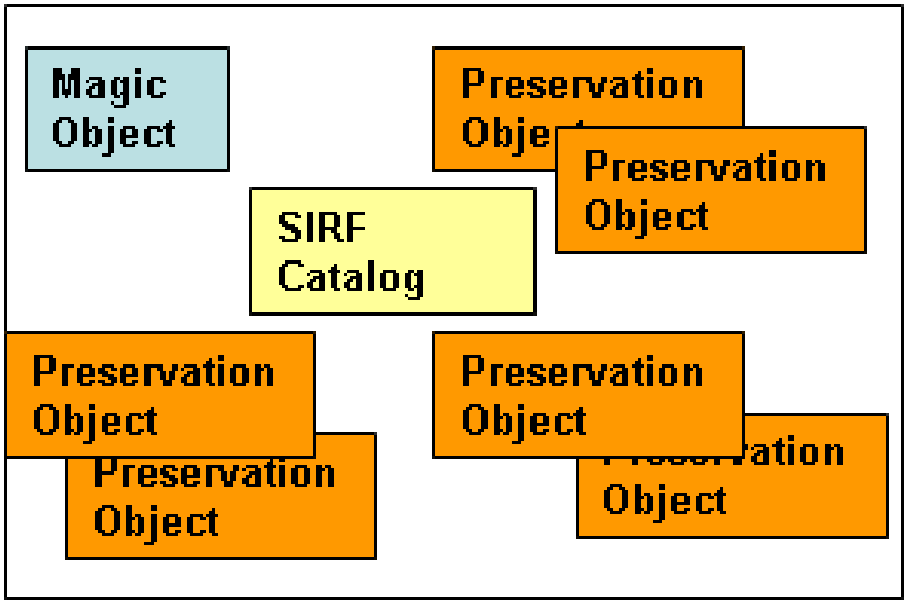
- SIRF is a logical data format of a **storage container** appropriate for long term storage of digital information
  - A storage container may comprise a logical or physical storage area considered as a unit.
    - Examples: a file system, a tape, a block device, a stream device, an object store, a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

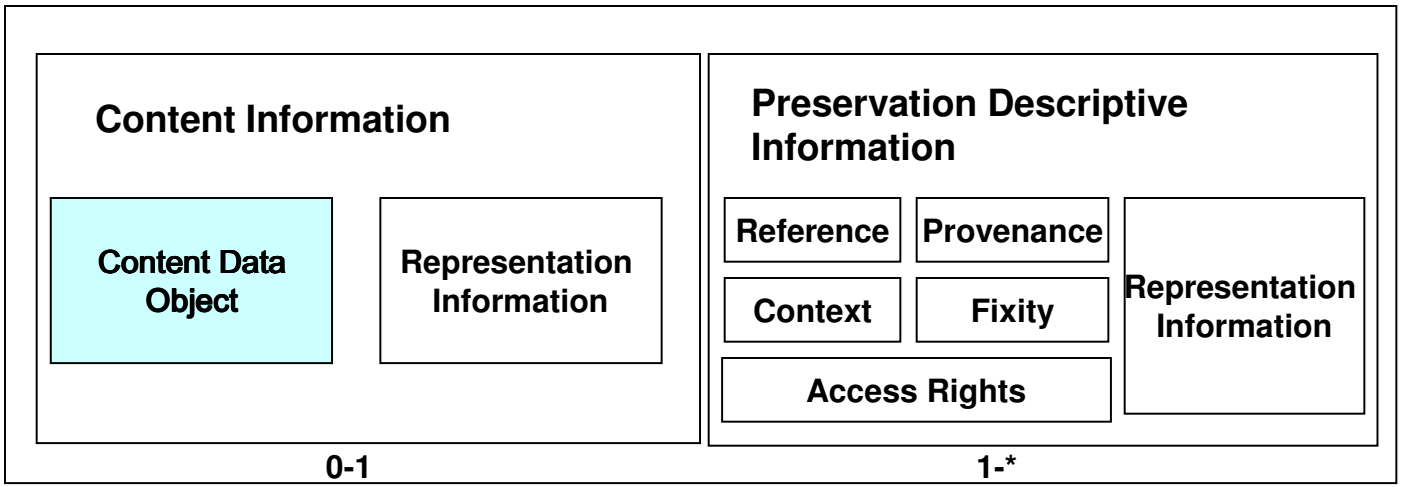
SIRF Components

A SIRF container includes:

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



OAIS Archival Information Package (AIP)



- OAIS AIP is an example of preservation object
- Domain specific packaging formats for preservation objects:
  - XML Formatted Data Unit (XFDU)
  - VERS Encapsulated Object (VEO)
  - Metadata Encoding and Transmission Standard (METS)
  - Preservation metadata: Implementation Strategies (PREMIS)

Layered Approach

- A SIRF container assumes an underlying object interface
  - Example object layers
    - Advanced: OSD, Cloud, XAM
    - Lower level: UDF, CDFS, FAT, LTFS
- SIRF metadata is defined at two levels
  - Level 1 catalog (L1) – unique metadata, not in the preservation objects, that is mandatory to make preservation objects portable into the future
  - Level 2 catalog (L2) – information that is probably also in the preservation objects, that is needed for fast access to the preservation objects

SIRF Level 1 – Work in Progress

The SIRF catalog includes metadata such as:

- Container information:
  - Spec ID and version
  - SIRF level
  - Container ID
  - State
  - Container provenance
  - Audit log object ID

For each Preservation Object (PO):

IDs

Children's ID

Dates

Packaging format

Fixity

Retention

Preservation profile

Audit log object ID

Extension

Preservation Object (PO) – IDs Category

- Fields:
  - **PO name** – non unique identifier e.g. file name
  - **PO version ID** – unique identifier that identifies the specific version of the PO
  - **PO logical ID** - a unique identifier that identifies the various versions that originate from the same ancestor
  - **PO parent ID** - 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.

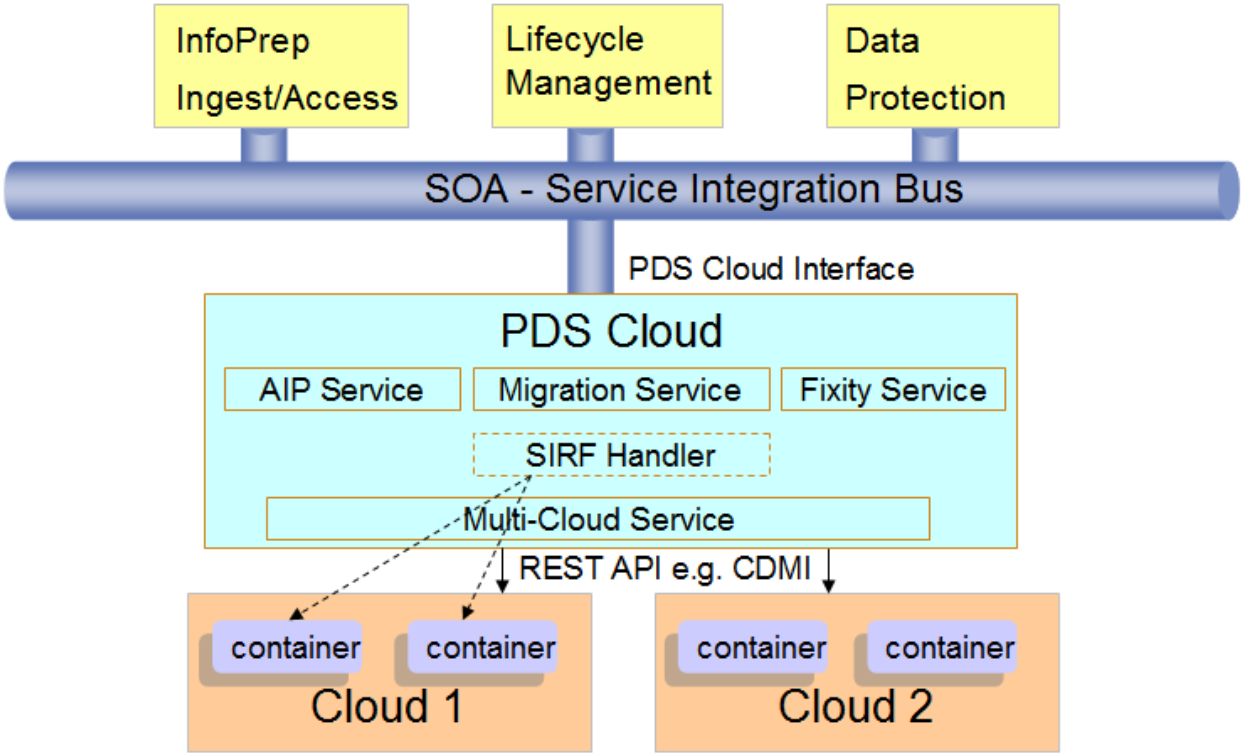
ENSURE

ANALYTICS & BIG DATA SUMMIT

- ENSURE is FP7 EU Project in the area of preservation
  - Three year Integrated Project (IP) started Feb. 1, 2011
  - Consortium of 13 partners (industry and academic)
- ENSURE has a business/industry-oriented focus
  - Drivers for preservation are both regulatory and business value
- Demonstrated with three use case: Health Care, Clinical Trials and Finance
- Contributions to standards is a goal of the project

SEVENTH FRAMEWORK PROGRAMME

PDS Cloud and SIRF in ENSURE

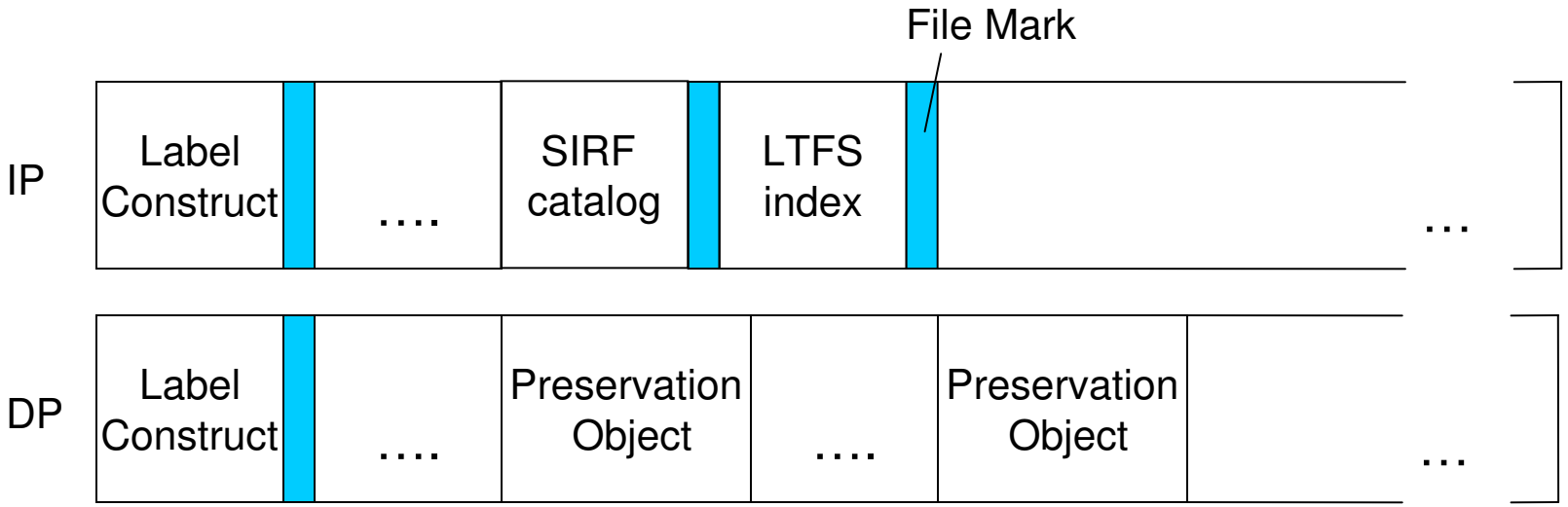


•Preservation DataStores (PDS) in the Cloud provides preservation-aware storage services for ENSURE based on OAIS
•The SIRF Handler component in PDS Cloud is for future implementation

Outline

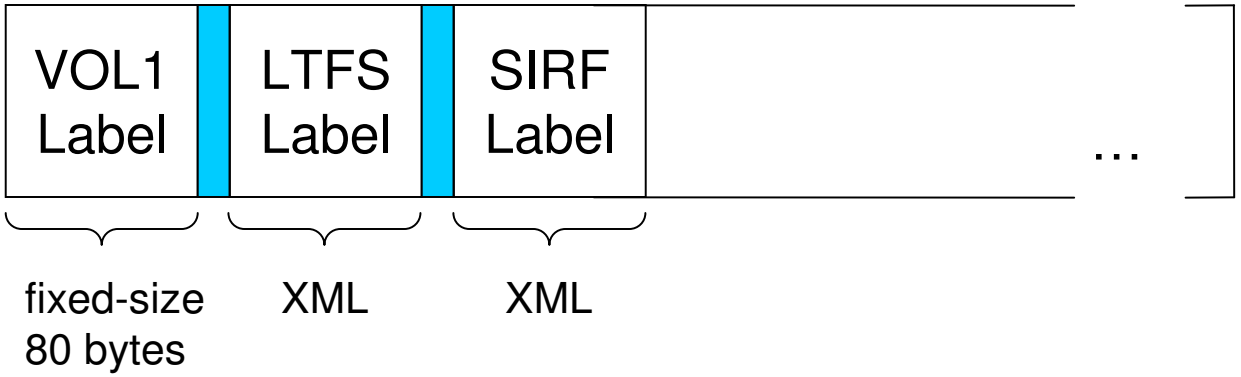
- Introduction and Challenges
- LTFS: Linear Tape File Systems
- SIRF: Self-contained Information Retention Format
- EU ENSURE: Enabling kNowledge, Sustainability, Usability and Recovery for Economic Value
- LTFS and SIRF
- Summary

SIRF and LTFS: Volume



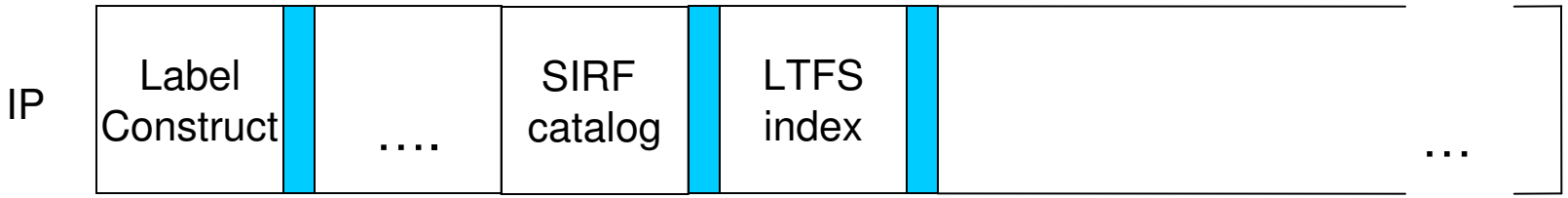
- The SIRF catalog resides in the index partition
  - LTFS application has rules to indicate what to store in the index partition.
  - This is used to indicate to store the SIRF catalog in the index partition.
- A preservation object (PO) is mapped to an LTFS file
- We don't maintain generations of SIRF catalog – there are versions of the POs

SIRF and LTFS: Label Construct



- VOL1 Label includes for example volume identifier (6 bytes), implementation identifier (13 bytes), owner identifier (14 bytes).
- LTFS Label includes for example creator, volume UUID, blocksize, compression, partitions ids.
- The SIRF Label is the “magic object” and includes for example specification ID and version, SIRF level.

SIRF and LTFS: IDs Category



The SIRF catalog is an XML including:

- Container information
- PO information for each PO
  - The IDs category in each PO information includes a unique identifier - PO Version ID
  - The PO Version ID will be “LTO” + Tape UUID + LTFS <fileuid>
  - The LTFS fileuid is a 64 bits sequence number generated for each LTFS file

Summary

- Generating and collecting very large data sets is becoming a necessity in many domains that also need to keep that data for long periods
- There are many financial and practical reasons to prefer tape storage for such big data archives
- LTFS provides a file system implemented on dual-partition linear tape
  - being standardized by SNIA LTFS TWG
- SIRF is a logical data format of a storage container appropriate for long term information retention
  - being developed by SNIA LTR TWG and will be used in ENSURE
- SIRF serialization for LTFS provides economically scalable storage containers for long term retention of big data
- More information on LTFS: ***www.trustlto.com/LTFS_Format_To Print.pdf***
- More information on LTR and SIRF: ***http://www.snia.org/ltr***