

Storage Standards Work

Joel Williams

October 15, 1996

**<http://www.ses-inc.com>
joel.williams@ses-inc.com**

Systems Engineering and Security, Inc.

Standards Participation

- **File-level Metadata for Portability of Sequential Storage (FMP) Study Group**
 - **Fernando Podio, NIST, is the chairman**
 - **Meeting Schedule**
 - **December 12 - 13 in Silver Spring**
 - **February 4 - 5 in Silver Spring**
 - **April 14 in NYC (AIIM Conference)**
- **IEEE Storage Systems Standards Working Group**
 - **Jack Cole, ARL, is the chairman**

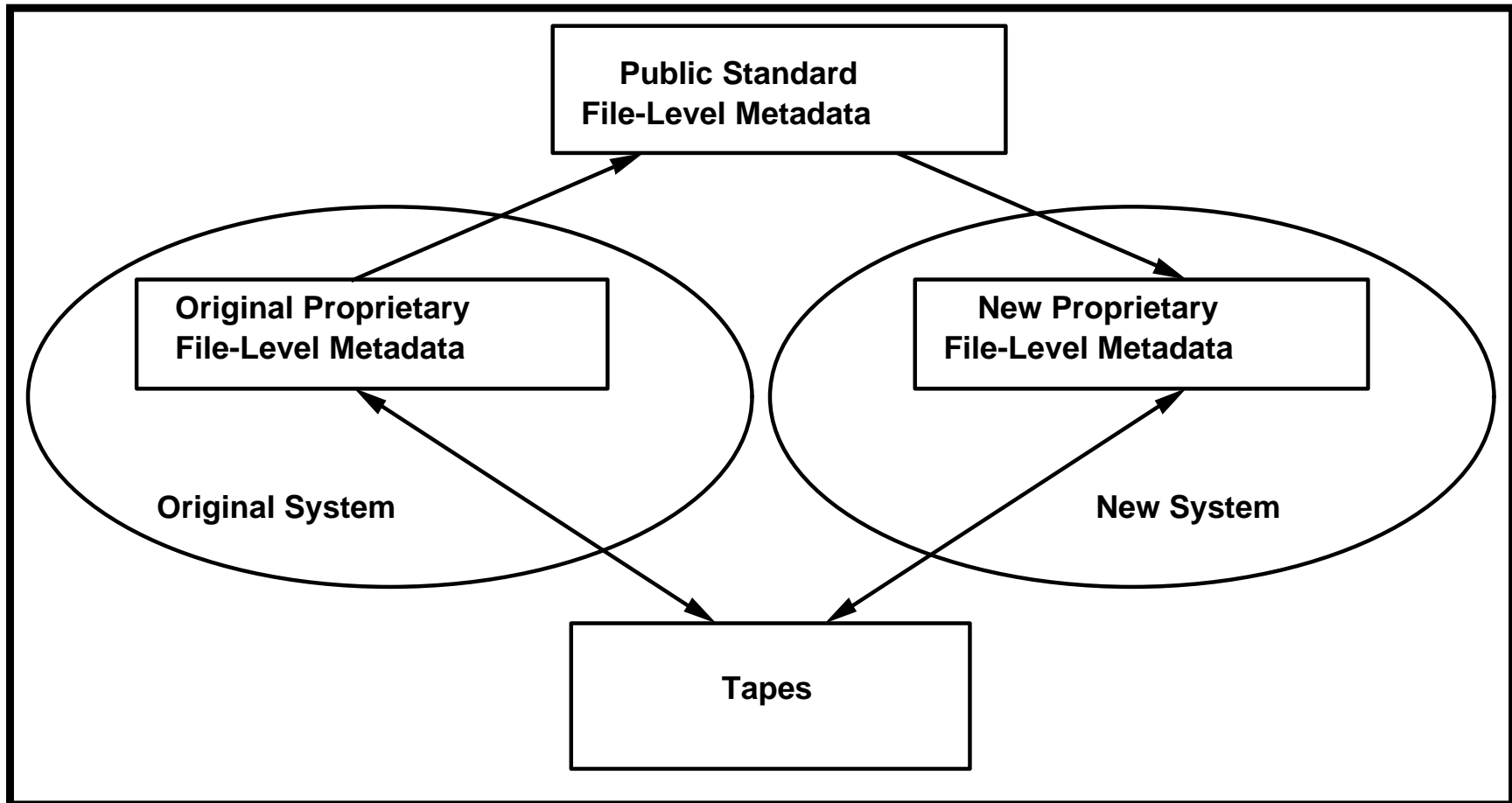
- **Meeting Schedule**
 - **November 5 - 7 in Cupertino**
 - **January 22 - 24 in Seattle**
 - **March 11 - 13 in Colorado**

FMP Progress

- **Purpose:** to develop a standard for the export of file-level metadata, so that tapes from one File Storage Management System (FSMS) may be imported to another FSMS without copying the tapes into a new format
- **Original Proposal** (briefed to THIC earlier) was to develop a tape format having an on-tape directory; this was deemed not feasible
- **Committee** has compiled a collection of data elements, and has defined a format for the export of the metadata
- **Current drafts** may be found at <http://www.ses-inc.com/~joel>

- **A significant unresolved issue is how to represent striped files**

Metadata Scheme



Outline of Current Position

- **Four types of data elements--those**
 - **Having to do with the export itself**
 - **Having to do with the tapes**
 - **Having to do with the file segments**
 - **Having to do with the directory and file structure**

- **Export divided into those four sections, with the data elements having to do with the export itself at the first, and the others afterwards in any order**

- **Format of the export**

- **Character records - unicode**
- **Named records, with the name the first field**
- **Fields separated by the SYMBOL FOR NULL, 2400(hex)**
- **Records separated by the SYMBOL FOR RECORD SEPARATOR 241E(hex)**

Characteristics of the Proposed Record Format

- **A collection of English names is defined for record names and field identifiers, but the remainder of the record may be in any language**

- **16-bit character set allows for different languages**

- **The standard may be extended by using different record names**

- **Some records may be optional--since the records are named, the order may be somewhat flexible**

Data Elements Having to do with the Export Itself

Identifier	Description of the Field	Required?
EXPORT_VERSION	Name and version of the export standard	Yes
EXPORT_HARDWARE_SYSTEM	Computer system which wrote the export	Yes
EXPORT_OS	Operating system and version which wrote the export	Yes
EXPORT_TIME	Date and time of the export: mm/dd/yyyy/ hh:mm:ss	Yes
EXPORT_FSMS	Vendor name and version of FSMS	If it exists

The version must be first.

Data Elements Having to do with a Tape in the Export Set

Identifier	Description of the Field	Required?
TAPE_UUID	UUID of the tape	Yes
TAPE_ID	External label - Visual Serial Number	Yes
TAPE_INTERNAL_LABEL_TYPE	Standard label format name	If it exists
TAPE_INTERNAL_LABEL	Tape name on internal label	If it exists
TAPE_RECORDING_FORMAT	Subfields: MFR for the tape manufacturer, MODEL, for the tape model, and TYPE for the tape type.	To the extent available
TAPE_HW_COMPRESSION	Hardware compression algorithm name	If it exists
TAPE_DRIVE_TYPE	Subfields: MFR for the drive manufacturer, MAKE for the make, and MODEL for the model	Yes

Data Elements Having to do with a Tape in the Export Set (Continued)

Identifier	Description of the Field	Required?
TAPE_PARTITIONS	The number of partitions on the tape	If applicable
TAPE_PARTITION_STRUCTURE	Subfields: NAME for the partition name, NUMBER for the partition number, 0, 1, 2..., and FREE_SPACE for the approximate free space in bytes remaining in the partition	If applicable
TAPE_FREE_SPACE	Approximate free space in bytes if the tape is not partitioned	If applicable
TAPE_BLOCK_SIZE	Block size, in bytes, if constant throughout the tape (except possibly for ending block)	If applicable
TAPE_NO_BLOCKS	Number of blocks between file marks, if constant throughout the tape	If applicable
TAPE_FAMILY	Name of the tape family	If supported

TAPE_BILLING_ID	Billing id or charge number for this tape	If supported
------------------------	--	---------------------

The UUID and Id must be first.

Data Elements Having to do with a File Segment

Identifier	Description of the Field	Required?
FILE_SEG_UUID	UUID of the File	Yes
FILE_SEG_ID	Full path name of the file	Yes
FILE_SEG_PARTITION	Partition number where the file is located: 0, 1, 2,...	If applicable
FILE_SEG_PARTITION_NAME	Name of the partition where the file is located. This is for implementations where the partition represents a logical volume.	If applicable
FILE_SEG_NO	Segment number of the file: 0, 1, 2,...	If applicable
FILE_SEG_TAPE_UUID	Value of TAPE_UUID in Tape data elements	Yes

FILE_SEG_TAPE_ID	Value of TAPE_ID in Tape data elements	Yes
-------------------------	---	------------

Some Data Elements having to do with Directory and File Structure

Identifier	Description of the Field	Required?
DIR_DIR_UUID	UUID of the parent directory	Yes
DIR_DIR	Full path name of parent directory, with list of child directories	Yes
DIR_DIR_ACL	Access list on the parent directory	Optional
DIR_FILE	Full path name of parent directory, with list of files in it	Yes
DIR_FILE_ACL	File name - one in the list above, with access list	Optional
DIR_FILE_FAMILY	File name followed by family name	If supported
DIR_SYMBOLIC_LINK	File name followed by name of link	Optional
DIR_HARD_LINK	File name followed by name of link	Optional
DIR_BILLING_ID	Billing id associated with the parent directory	If supported

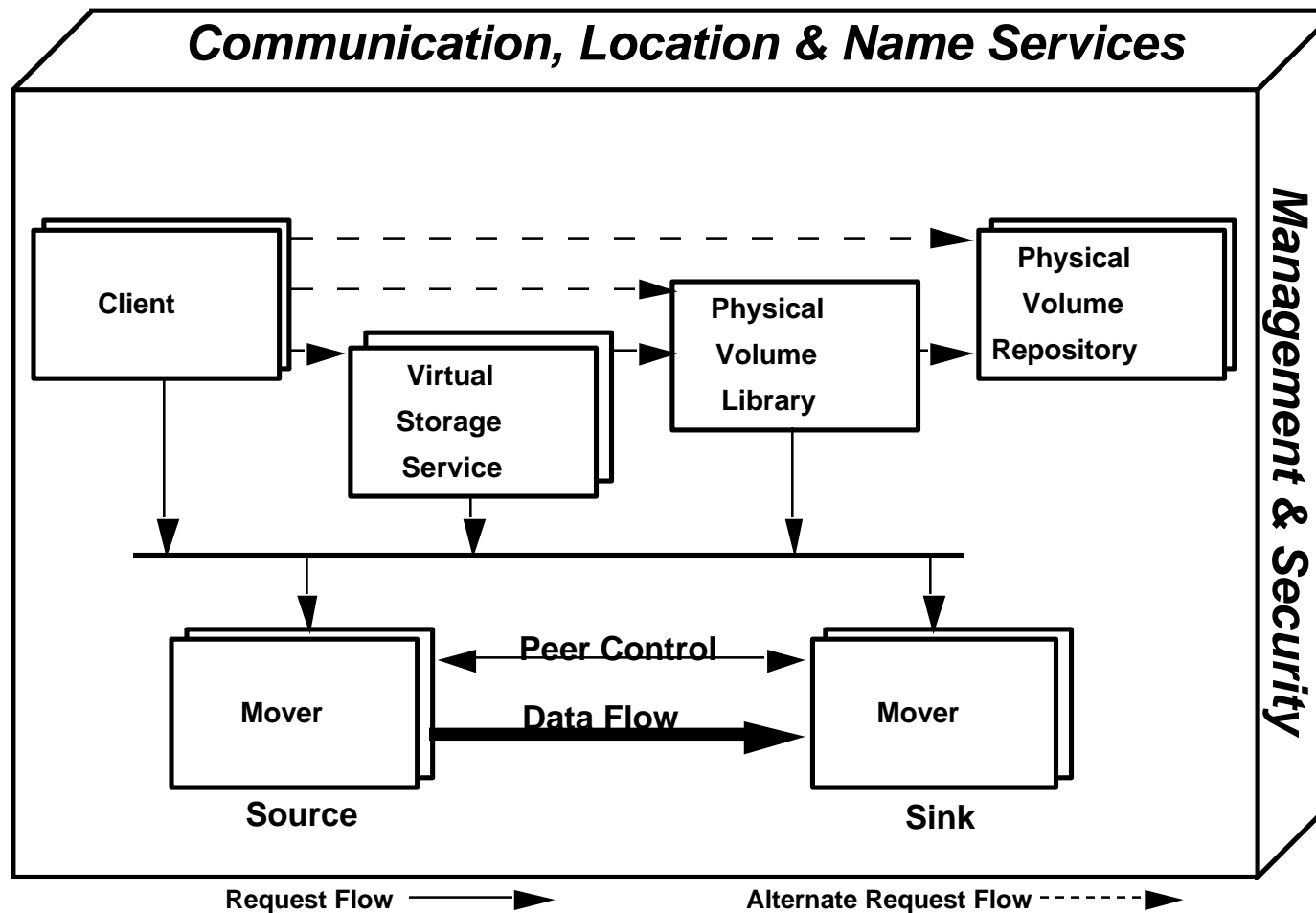
DIR_FILE_ BILLING_ID	Billing id associated with the file	If supported
---------------------------------	--	---------------------

Background of IEEE Standards Work

- **The IEEE Storage Systems Standards Working Group has developed a *Reference Model for Open Storage Systems Interconnection* in September 1994**
- **Work since then has concentrated on the development of Application Programming Interfaces for the various modules defined by the model**
 - **The Physical Volume Repository**
 - **The Physical Volume Library**
 - **The Mover**
 - **The Virtual Storage Server**
- **A decision was made that the client-server context of the standard was to be provided by DCE**

- **There are, however, no DCE-specifics in the function calls of the proposed API**

The Reference Model



The Reference Model: Key Definitions

- ***Store***: an addressable storage space, either physical or virtual
 - Physical attributes defined by the media type
 - Virtual attributes defined by the client request

- ***Device***: A set of media access points for data access and mount points for physical access.

- ***Physical Volume***: The recording medium accessible without intervening load operations.

- ***Cartridge:*** A set of physical volumes or cartridges.

The Reference Model: Modules

- ***Physical Volume Repository (PVR)***: It sees cartridges and drive mount points, and its major operation is to *mount* cartridges.
- ***Physical Volume Library (PVL)***: It makes the volume to cartridge mapping and causes the PVR to mount cartridges. Its major operation is to *mount* physical volumes.
- ***Virtual Storage Server (VSS)***: It creates virtual stores and performs the store to volume mapping. Its major operation is to create and manage virtual stores.

- **Mover (MVR):** It manages data transfer and is designed in particular to manage high-speed data transfer. Its major operation is to *load* media to media access points and to perform data transfer

Why the PVR Layer Exists

- **Minimize the effort required to incorporate into a storage system a new device or system for loading media**
 - **No software changes to a properly designed PVL**

- **Allow sharing of a robotic device between non-cooperating clients**

Design Principles

- **One interface for both management and service functions**
 - **Management is not an add-on, or an application customized for a particular site**
 - **No more arguments about whether a call is for management or service**
- **Operating System Neutral (but a C-language API)**
- **One interface for both human and robotic implementations**

- **PVR makes accessible to the client sufficient information to allow it to be used intelligently--no other source of information is needed**

PVR Definition in the API

- **One or more storage locations organized into partition objects**
 - **Storage locations for exactly one cartridge type**
 - **One location with one disaster protection attribute**
 - **May be defined as a hierarchy**
- **Zero or more cartridge objects**
- **One or more media location domains (MLDs) for grouping cartridges of the same type**
- **Zero or more drive objects**
- **Zero or more device location domains (DLDs) for grouping drives of the same type or dependent PVRs**

- **Zero or more area objects for staging prior to mounts**

PVR Definition in the API (Continued)

- **Zero or more port objects for ejecting cartridges from or injecting cartridges into the PVR**
- **One or more transfer mechanisms, either mechanical or human, operating in the same context, and capable of moving cartridges between their storage locations, drives, and ports**
- **One or more controllers which command the transfer mechanism within a single context**
- **Additionally, there is a task object. Tasks execute asynchronously from the calling thread, and may be monitored or canceled by the client.**
- **A partition may contain one or more MLDs which may contain PVR cartridges.**

- **MLDs and DLDs are effectively paired, depending on whether or not a cartridge from an MLD may be mounted into a drive in a DLD**

Basic PVR Operations

- **Mounting cartridges on drives and dismounting cartridges from drives**
- **Injecting and ejecting cartridges through ports.**
- **Management directives: to allow object attributes to be accessed and modified.**
- **Note that PVR does not access the drive data path and has no knowledge of the contents of cartridges.**

Non-Cooperating Clients

- **The PVR subcommittee of the SSSWG took it as a requirement that the PVR be able to be shared by non-cooperating clients.**

- **Access Control Lists on the MLDs and DLDs accomplish this by effectively partitioning the PVR**

- **Note that in a full implementation of the Reference Model, the PVR has essentially only two types of clients: the Physical Volume Library and System Management**

Status

- **Standards documents are provided at <http://www.ses-inc.com/~joelw/>**
 - **Microsoft Word 6 and HTML versions, with links for the PVR API**
 - **FMP documents available in HTML, and will be available in Word 6 as well**
 - **FTP capability for both versions**

- **Volunteers are needed to review**

- **Next projects**
 - **PVL API**
 - **FMP Study Group is expected to begin to write a standard**

- **Check <http://www.arl.mil/IEEE/ssswg.html> (the SSSWG Web Site) for the details about the next SSSWG meetings.**