

Open Source Systems Managed Storage

Stephen Cranage
StorageTek



STORAGETEK®

OpenSMS Principal Attributes

- ◆ Distributed TMS Namespace
- ◆ DMAPI Create/Modify/Data Fault Handling
- ◆ Policy Based Management of File Objects
- ◆ Rich Data Classification
- ◆ Filesystem Metadata Capture into RDBMS
- ◆ Distributed Configuration Management

Distributed TMS Namespace

- ◆ Hardware Independent
- ◆ Complete Abstraction for Removable Media, Including Device Allocation, Mount Request System, Low Level Device Control, Media Management and File Cataloging
- ◆ Provides a Namespace that is Flat, and Logically Divided into Volumesets
- ◆ File Objects in TMS have Enterprise Wide Scope
- ◆ TMS Clients are All User Level Code – Widely Ported Historically

Distributed TMS Namespace

- ◆ TMS Data Movers are FC and IP enabled
- ◆ Transports are any-to-any Shared SAN Resources
- ◆ Objects are Stored in the TMS Namespace as ANSI Standard HDR3 Label Tape Datasets, Metadata Model is Based on Available HDR3 Label Fields
- ◆ Completely Platform Independent, Application Neutral Data Representation in the TMS Namespace

Distributed TMS Namespace

- ◆ Data Sharing over the SAN Facilitated by Simplicity of Metadata Model
- ◆ File Updates, no Block Updates – Gno++
- ◆ Single User Access at Any Time
- ◆ Hide Access Limitations with Private Filesystems using Archive Policies and Handlers that Service Reads on Block Released Files (Data Faults)

DMAPI hsmd

- ◆ Based on XFS DMAPI
- ◆ Performance Neutral, Managed Regions Turned off on First Write I/O
- ◆ Create/Modify Events are Synthesized Seconds after the Event
- ◆ Result is a File Copy (Copy Policy) and/or SQL insert into RDBMS Server
- ◆ RDBMS work_q Drives Surrogate Policy Engines (Archive, Block Release, etc)

Policy Engines

- ◆ Copy Policy
 - Async File Replication to Federated Filesystem
 - File Rather Than Block Level
 - Enabler for Multi-tier Filesystems i.e., SSD Front Ending ATA & FC
 - Block Release with Data Fault Servicing from Peer Filesystem
 - Source Filesystem Requires DMAPI, Target Does Not

Policy Engines

- ◆ Archive Policy
 - Near Real Time Duplication of Files into TMS Namespace
 - Supports Block Release with TMS servicing Data Faults
 - TMS Containers Created with Policy Attributes
 - Archive Policy Directs Files into Appropriate Volumesets

Archive Policy Data Classification

- ◆ Select File Objects from work_q
- ◆ Perform REGEX on name/attributes
- ◆ Disaggregate File Objects Into vshandler Queues for Archiving into the Appropriate TMS Volumeset
- ◆ Model for Other Policy Engines That Can Act Directly on the File Objects to Set User Attributes, or Block Release the File, or Execute Some Other User Process

Meta Data RDBMS Integration

- ◆ File Create/Modify and Data Fault Events all cause a filehandle to be Inserted in a work_q
- ◆ On Create/Modify, We Also Insert the Metadata into a Metadata Table
- ◆ Unused Now, but Plan on Some Block-Release Candidate Selection and SRM Functionality Later On.

TMS Namespace Utilization

- ◆ Use Private Filesystems as a Means to Convey File Objects into a Global TMS Namespace
 - Distribute Metadata (inodes from dump/restore) to Create Empty File Objects in other Private, Local Filesystems
 - Data Fault Will Return the Object From TMS
 - Archive Policy will Update the TMS Object as New Generation on Local Modify
 - Block Release the Local File to Have a Subsequent Read Data Fault Return the Most Recent Version

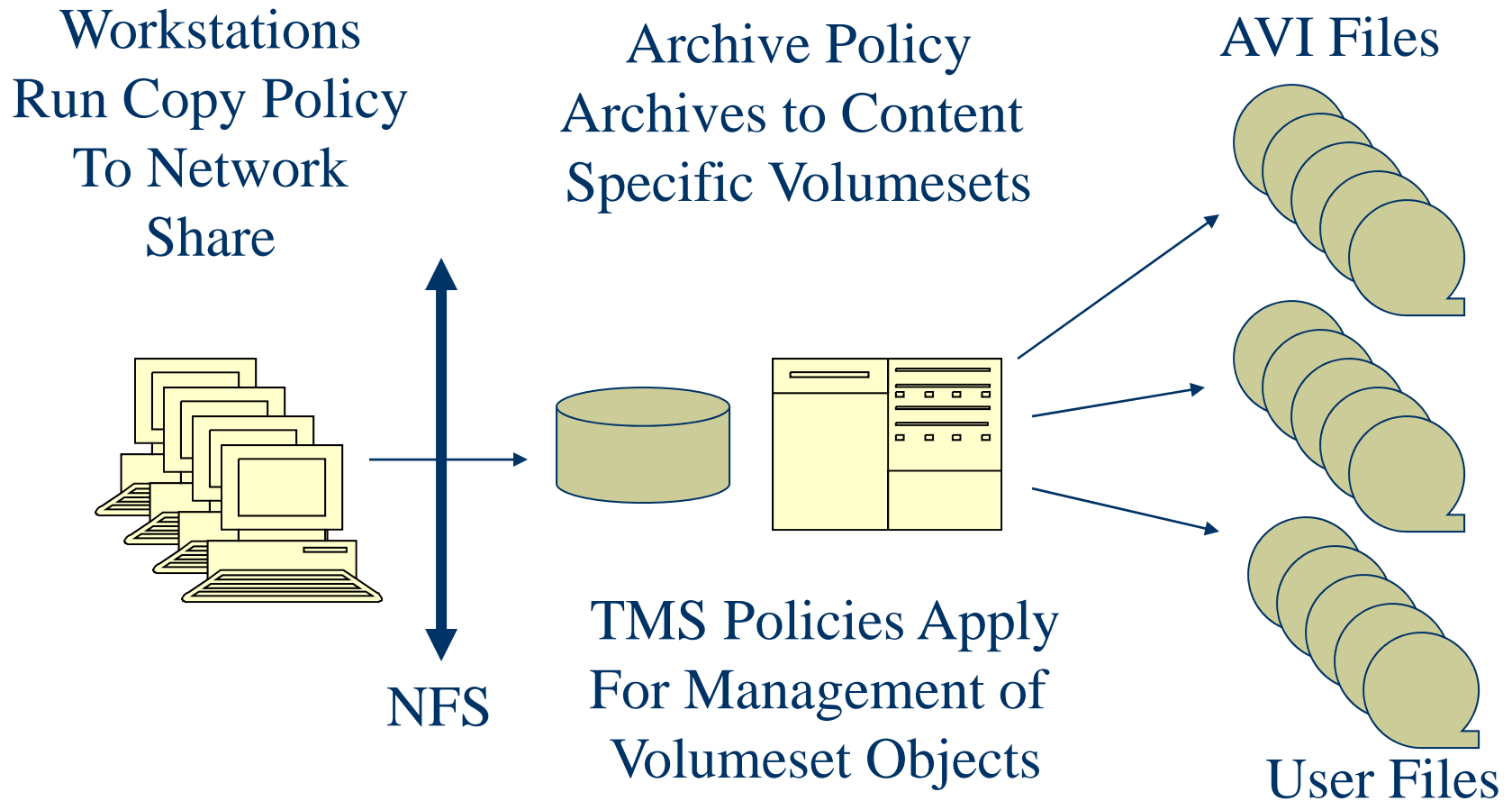
TMS Namespace Utilization

- ◆ Access the Volumeset from Command Line or API
- ◆ Perform Stream I/O from the Command Line or API

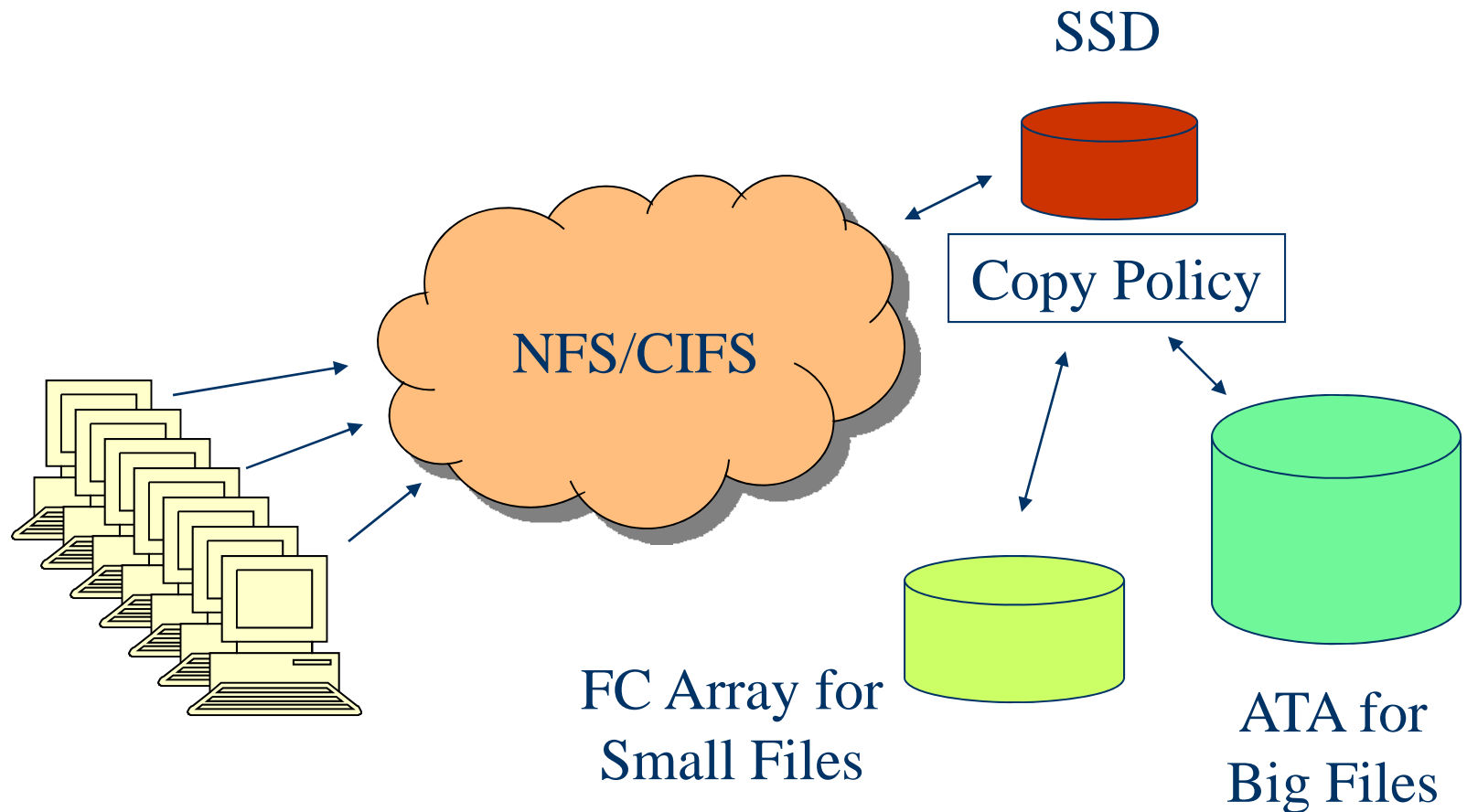
OR

- ◆ Use GUI to Browse a Volumeset's Files
- ◆ Execute a Stored Procedure Against the File Object

Example Topologies



Example Topologies



Distributed Configuration Management

- ◆ OpenSMS is a Distributed Storage Management Toolkit
- ◆ Server Based Conf Files Added Complexity, Counter to Our Goals
- ◆ Needed and XMLish Way to Define and Distribute Rules and Policies

Distributed Configuration Management

- ◆ A GUI Builds a Set Of Nested name/value pairs to define policies, Volumeset Attributes, and all Other Configuration Variables
- ◆ SMS daemons Request Their Configuration Variables At Startup from RDBMS Server
- ◆ XML Conf Files on a Web Server Logical Next Step

Next Steps

- ◆ “Dataless” Dump Files done, Need to create an Integrated Data Protection Environment
- ◆ More Policy Engines – FTP Policy, etc
- ◆ Web Server for TMS Namespace
- ◆ File “rollback”

URLs

<http://opentms.sourceforge.net>

<http://openhsm.sourceforge.net>