RCM Apvd RELEASED 2012/05/11

00110S00003
Revision A

HIGH PERFORMANCE COMPUTING MODERNIZATION PROGRAM (HPCMP)

SLM SYSTEM DESIGN SPECIFICATION

Prepared for:
High Performance Computing Modernization Program Office
10501 Furnace Road, Suite 101
Lorton, VA 22079-2624

Contract # GS04T09BEC0003

Contractor: **General Atomics**
3550 General Atomics Court
San Diego, CA 92121-1122

GA PROJECT 39350/30332

GENERAL ATOMICS

REVISION HISTORY

Revision	Date	Identify the type of change documentation
A	11MAY12	Initial Release

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

Acronym	Description
ACSLS	Automated Cartridge System Library System
AES	Advanced Encryption Standard
AFRL	Air Force Research Laboratory
API	Application Programming Interface
ARL	Army Research Laboratory
ARSC	Arctic Region Super Computing
BOM	Bill of Materials
CA	Certificate Authority
CAC	Common Access Card
CCM	Counter with CBCMAC
COTS	Commercial off-the-shelf
CWFS	Center-Wide File System
DB	Database
DHCP	Dynamic Host Configuration Protocol
DICE	Data Intensive Computing Environment
DMAPI	Data Management Application Programming Interface
DNS	Domain Name Service
DoD	Department of Defense
DR	Disaster Recovery
DREN	Defense Research and Engineering Network
DSRC	DoD Supercomputing Resource Center
ECC	Elliptic Curve Cryptography
ELOM	Embedded Lights Out Management
ERDC	Engineering Research and Development Center
FC	Fibre Channel
FRA	Flash Recovery Area
FTP	File Transfer Protocol
GA	General Atomics
GID	Group Identifier
GUI	Graphical User Interface
HPC	High Performance Computing

Acronym	Description
HPCMP	High Performance Computing Modernization Program
HSM	Hierarchical Storage Management
HTTP	Hyper Text Transfer Protocol
HTTPS	Secure Hyper Text Transfer Protocol
ILM	Information Lifecycle Management
IPL	Initial Program Load
IPMI	Intelligent Platform Management Interface
IPSec	Internet Protocol Security
ITAR	International Traffic in Arms Regulation
KDC	Key Distribution Center
KMA	Key Management Appliance
KMS	Key Management System
KVM	Keyboard-Video-Mouse
LAN	Local Area Network
LCR	Logical Change Record
MAC	Mandatory Access Control
MCAT	Metadata Catalog
MHPCC	Maui High Performance Computing Center
NAC	National Agency Check
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
OCI	Oracle Client Interface
OEM	Oracle Enterprise Manager
OR	Open Research (sites such as ARSC)
PI	Principal Investigator
PID	Process Identifier
pIE	Portal to the Information Environment
PKI	Public Key Infrastructure
QFS	Quick File System
RAC	Real Application Cluster
RPC	Remote Process Call
S/AAA	Service/Agency Approval Authority

Acronym	Description
SAM	Storage Archive Manager
SCA	Sun Crypto Accelerator
SBU	Sensitive But Unclassified (sites such as AFRL, ARL, and ERDC)
SCP	Secure Copy
SFTP	Secure FTP
SLM	Storage Lifecycle Management
SNMP	Simple Network Management Protocol
SQL	Structure Query Language
SRB	Storage Resource Broker
SSH	Secure Shell
SSL	Secure Socket Layer
TCP	Transport Control Protocol
TGT	Ticket Granting Ticket
TLS	Transport Layer Secure
UDP	Universal Datagram Protocol
UID	User Identifier
VLAN	Virtual LAN
VPN	Virtual Private Network
VSN	Volume Serial Number
WORN	Write Once Read Never

1 INTRODUCTION

This SLM Design Specification (SDS) documents the design of a secure, fully integrated Storage Lifecycle Management (SLM) solution of commercial off-the-shelf (COTS) components. The SLM solution's main components include the Nirvana® Storage Resource Broker® (SRB®) Information Lifecycle Management (ILM) infrastructure with Oracle Database and Oracle's SAM-QFS Hierarchical Storage Management (HSM) system, which already operates across all the DoD Supercomputing Resource Centers (DSRCs). A high-level architectural diagram of this design is depicted in Figure 1-1. A more detailed description of the components follows in section 3.1 below. This document is aimed at describing what the architecture consists of, and how the various components interact with each other. Additionally, it provides the details necessary to integrate the various components into the HPCMP SLM solution, which include discussion topics on implementation and operation of the solution.

This architecture integrates SRB with the existing High Performance Computing (HPC), security, and HSM (SAM-QFS) environment and the Defense Research and Engineering Network (DREN). SRB consists of SRB Agents, SRB Clients and a Metadata Catalog (MCAT) residing on an Oracle Database at each of the DSRCs. SRB Agents and SRB Clients are implemented in various components of the HPC environment. The sum of all SRB Agents that authenticate to the same MCAT is referred to as an "SRB Federation". A single DSRC can be considered an SRB Federation. For example, a site with multiple SLM Servers and SRB Agents that all authenticate to the site's MCAT is considered an SRB Federation.

The data at rest on tape is protected through encryption and key management using Oracle's Key Management System (only the design is provided here, the integration details may be described in a subsequent document at a time when the Government requests its implementation).

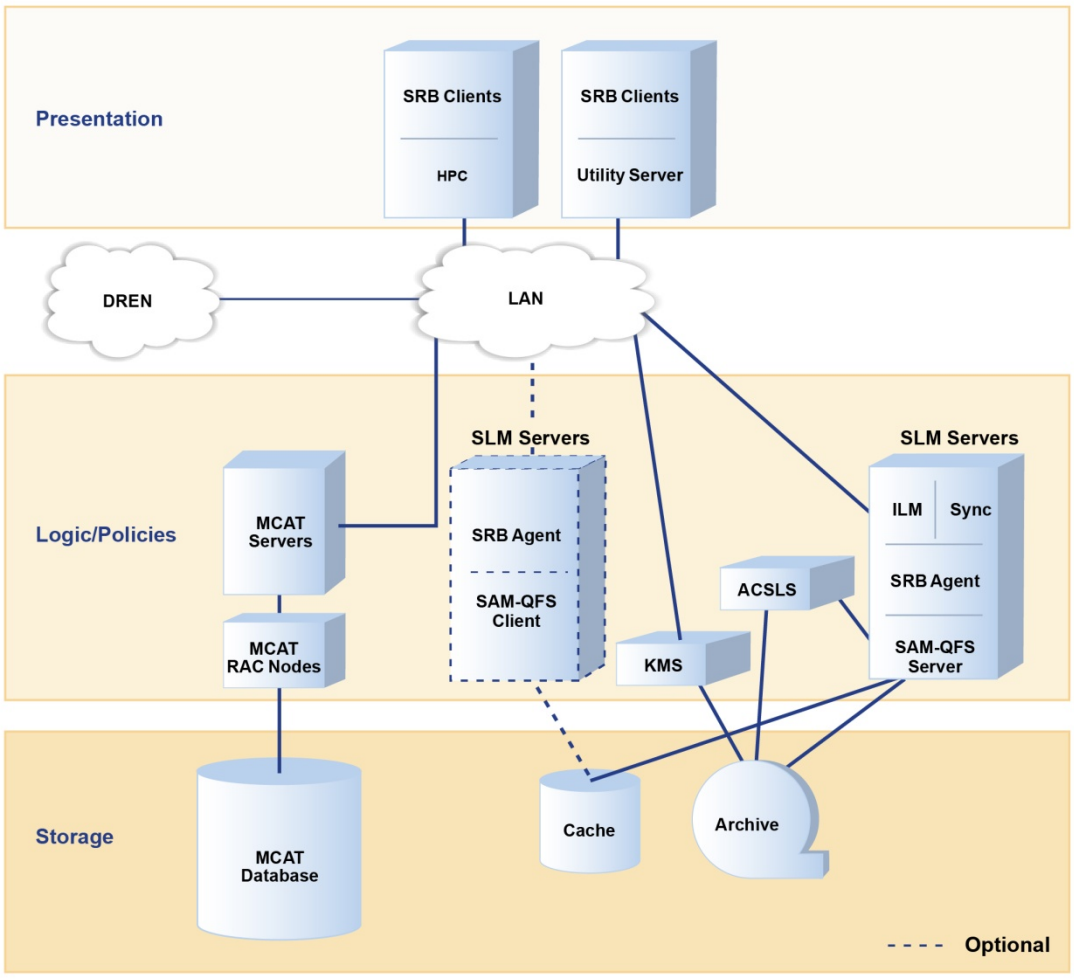


Figure 1-1. SLM Architecture

1.1 SLM Overview

The SLM solution integrates the ILM with the HSM components at each of the HPCMP centers. For the purpose of this document, the existing HSM Servers now become known as SLM Servers. Of these components, the SLM solution introduces a new hardware infrastructure to support the Metadata Catalog (MCAT) in a two-tier configuration.

In this design, the infrastructure includes 4 servers split into 2 tiers of two servers. The first tier is the MCAT Server and the 2nd is an MCAT RAC Node. The second tier is backed by a direct attached storage array with about 19 TB of raw storage (when fully populated with 300 GB drives).

The software infrastructure includes the Nirvana SRB Clients and Agents. SRB also relies heavily on an MCAT Database, and, in this instance, it is supported by Oracle RAC. While RAC and the MCAT Servers reside solely on the new hardware, the SRB Clients and Agents are installed in various existing systems throughout the HPCMP centers. SRB Clients are installed

on HPC systems and Utility Servers (or anywhere a user is likely to need access to an SRB controlled storage object). Agents reside where Data Objects are likely to be found, such as the existing HSM servers.

Two modes of operation are described in this document: Isolated Enclave Mode and Replicated Mode. In Isolated Enclave Mode, shown in Figure 1-2, the SLM solution is implemented entirely within a Center. Data and Metadata are not normally exchanged between centers unless the user explicitly connects to a remote site. Communication among SLM Clients, Agents, and Servers is through the Center's public LAN. Every site is its own SRB Federation.

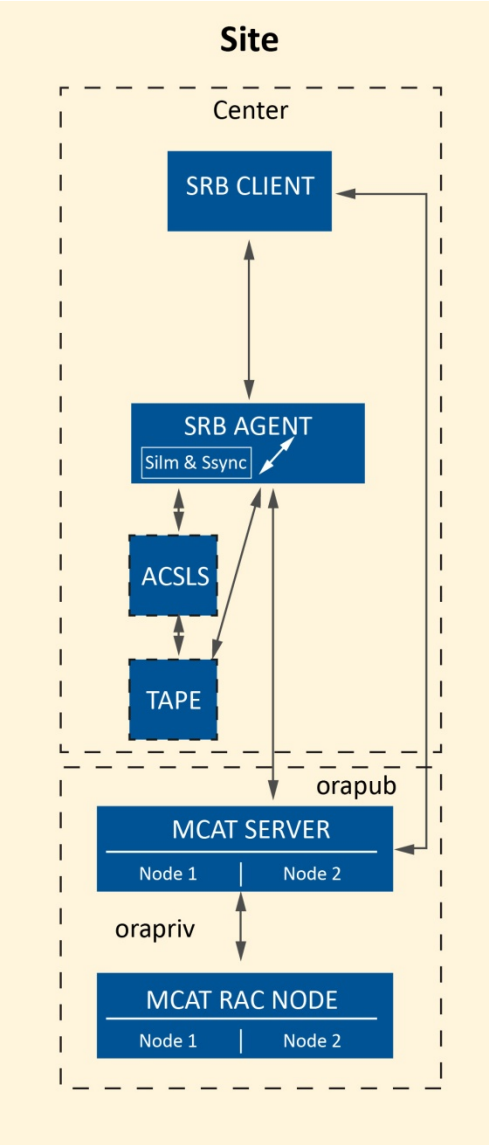


Figure 1-2. Isolated Enclave Mode

In Replicated Mode, communication across centers is necessary in order to support a Global HPCMP Namespace and to exchange data and metadata between sites. To achieve this, synchronization between MCAT Databases is necessary. This is accomplished via Oracle Streams. Only a single SRB Federation exists, which spans all sites. Figure 1-3 illustrates the various communication paths within and between sites in Replicated Mode.

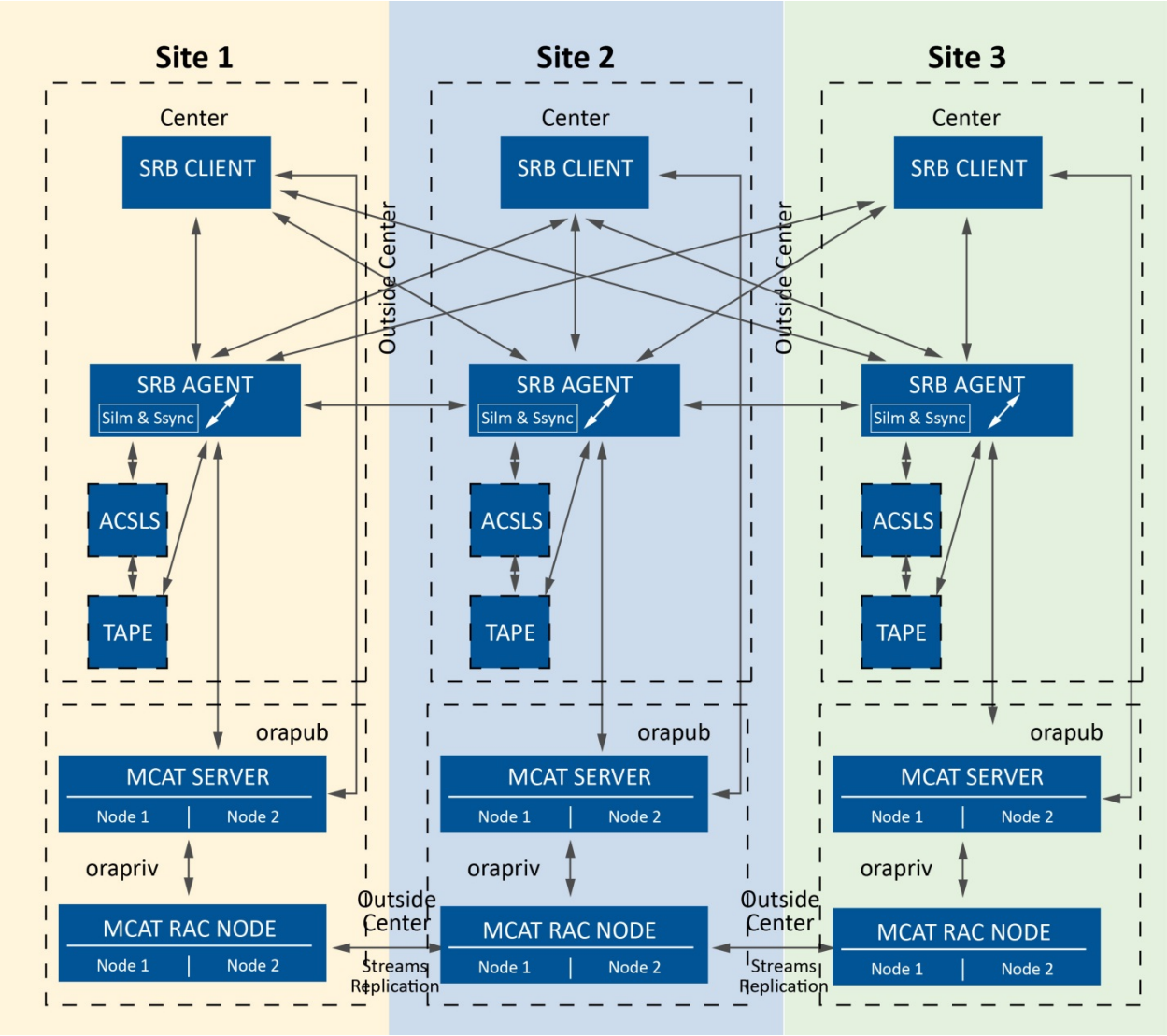


Figure 1-3. Replicated Mode.

The remainder of this document focuses on the design and implementation aspects of this solution to include:

- Global Namespace, Servers, and File Systems – the configuration of the global namespace and file system structure envisioned for the HPCMP data centers.
- Security – A thorough discussion of how SRB authenticates and interacts within its components and among data center systems.

- ILM Physical Configuration – configuration of the physical devices such as servers, networks and storage and how it integrates into an HPCMP center.
- Software/Applications – the installation and configuration of the various software components to include the SRB Clients, SRB Agents, and MCAT Database components.
- SRB-SAM QFS Interactions – discussion of file system activities, ILM policy management, and how SRB interacts with SAM-QFS for metadata synchronization.
- Administration – Administration tasks necessary for appropriately maintaining and monitoring the system, including Kerberos, failover and disaster recovery.
- Metadata Schemes – Schemes initially deployed in the solution to include Admin, Users, HSM, and an expansion on Dublin core.
- Data Management Policies – Archival, DR data management policies to include Release/ Purge and expiration of data objects.
- User Interfaces – brief discussion of the tools and interfaces available to the user.
- User Interaction – details such as metadata entry batch processing and file access.
- Cross-Site Metadata Replication – focused on the process and site interactions for effective metadata replication among the HPCMP data centers.
- Center-Wide File System (CWFS) – focusing on API requirements critical to SRB.
- Key Management System (KMS) – a discussion on a potential KMS implementation approach.

1.2 Objectives

SRB manages ILM metadata attributes to orchestrate data holdings at the HSM level in SAM-QFS. This provides minimal risk to the Program and efficient re-use of existing architecture. The highlights of this solution include:

- Existing files remain on same disk and tape media.
- SAM-QFS remains as the HSM archive system.
- COTS-based solution with ongoing development for commercial and government customers.
- Transparency to manage data while maintaining accessibility across sites.
- Integrate with and complement existing security infrastructure.
- Ease of transition to ILM system.
- Maintain business as usual while enhancing data management.
- Flexibility to manage data for decades to come.
- Web-enabled interface for access to the ILM system from remote locations.

The solution described herein aims at accomplishing the following objectives:

- Provide an open and transparent implementation by permitting files and directories to remain in native file system format with all metadata attributes residing in a standard Oracle database.
- In Replicated Mode, provide users a means to access and distribute files across multiple DSRC sites through the implementation of a Global Namespace spanning all sites.

- Provide a seamless way to collaborate and organize files across DSRCs into logical Collections, independent of physical storage location when the system is in Replicated Mode.
- Permit cross-domain access between sites in different security domains by implementing simple business logic to control access permissions and enforcing replication restrictions.
- Local files remain accessible in case of a wide area network failure as each site remains independent of the others.
- Minimize disruption to the environment and maximize reuse of existing HPCMP investment.
- Maintain and enforce two-factor authentication.
- Tightly enforce authorization so that no user can access or even see another user's files or directories unless explicitly permitted to do so.
- Eliminate Reduce WORN (Write Once Read Never) archives once and for allsymptoms by providing system- and user-level extended attributes for all data across all sites.
- Utilize the user account data from the portal to the Information Environment (pIE).
- Reduce the risk that failure of any single hardware component will interrupt work.
- Assure reliability of archive data through independent error detection and reporting.

2 GLOBAL NAMESPACE, SERVERS, AND FILE SYSTEMS

A Global Namespace is a heterogeneous, enterprise-wide abstraction of all file information, open to dynamic customization based on user-defined parameters. This becomes of particular importance as multiple network based file systems proliferate within an organization – the challenge becomes one of effective file management.

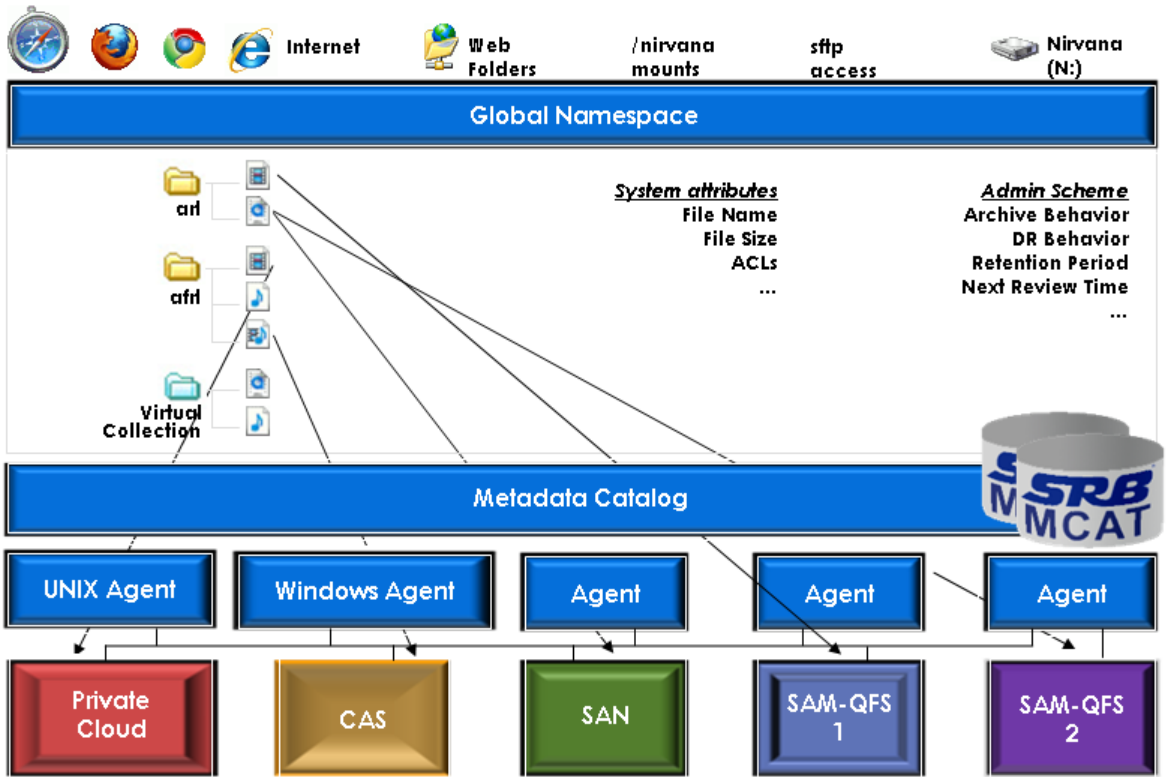


Figure 2-1. SRB Global Namespace

The SRB Global Namespace will be used for the $ARCHIVE file system. Each DSRC's MCAT namespace will initially host only its own file systems but the namespace hierarchy is designed to allow a future merging of all DSRC file systems into a single Global Namespace. The following Collection hierarchy is proposed for the HPCMP Global Namespace with each DSRC initially implementing (in Isolated Enclave Mode) only the portions that represent local file systems:

Table 2-1. Proposed Collection Hierarchy

Collection	Linked Collection	File System (from SAM Server)	Server
/			
/afrl			
/afrl/asc-ms1		/asc-ms1	arc3
/afrl/asc-ms2		/asc-ms2	arc4
/arl			
/arl/airforce		/archive/airforce	msas2
/arl/army		/archive/army	msas1
/arl/armya		/archive/armya	msas3
/arl/armyb		/archive/armyb	msas3
/arl/armyc		/archive/armyc	msas4
/arl/navy		/archive/navy	msas6
/arl/navya		/archive/navya	msas5

Collection	Linked Collection	File System (from SAM Server)	Server
/arl/navyb		/archive/navyb	msas5
/arl/navyc		/archive/navyc	msas5
/arl/others		/archive/others	msas6
/arl/quota		/home/quota	msas1
/ors			
/ors/archive		/sam-qfs/archive	wiseman
/erdc			
/erdc/erdc1		/erdc1	gold
/erdc/erdc2		/erdc2	gold
/mhpcc			
/mhpcc/samfs1		/usr/export/samfs1	drat
/mhpcc/samfs2		/usr/export/samfs2	drat
/navy			
/navy/b		/u/b	katrina
/navy/g		/u/g	katrina
/hpcmp		(cross-site Collections)	
/virtual		(virtual Collections)	
/archive			
USER HOME COLLECTION EXAMPLES			
/archive/tkendall			
/archive/tkendall/arl			
/archive/tkendall/arl/armya →	/arl/armya/tkendall	/archive/armya/tkendall	
/archive/tkendall/arl/armyb →	/arl/armyb/tkendall	/archive/armyb/tkendall	
/archive/tkendall/ors			
/archive/tkendall/ors/u2 →	/ors/u2/tkendall	/export/archive/u2/tkendall	
/archive/scheder			
/archive/scheder/arl			
/archive/scheder/arl/army →	/arl/army/scheder	/archive/army/scheder	
...			

The Global Namespace provides a method to logically organize all data into Collections. A single Collection may contain files from multiple sites or file systems and its naming does not necessarily specify its site affinity (i.e., only containing data from a single site).

The site-specific Collections (i.e., /afrl, /arl, /ors, /erdc, /mhpcc, /navy) are used to initially register all existing SAM-QFS file systems into the Global Namespaces at each center (Isolated Enclave Mode) or a single Global Namespace (Replicated Mode). To avoid naming conflicts, it is recommended that existing user archive directories (i.e., $ARCHIVE) are not spread across

multiple file systems so that they can easily be consolidated at the same hierarchy level underneath the site-specific Collections (e.g., /arl/armya/tkendall).

Users may use the site-specific Collection naming as a guideline as to where files are physically stored but are not prevented from storing files at sites that do not match the Collection name (e.g., /arl/armya/tkendall/myfileAtAFRL could physically reside at AFRL on the file system path /asc-ms1/tkendall/myfileAtAFRL).

The “logical” Collections that users would be assigned as Home Collections are all located under the /archive Collection. The “physical” Collections, which mirror the SAM file system structure are all located under the site-specific Collections (e.g., /arl or /ors). These “logical” and “physical” Collections and the linking between them are illustrated in Figure 2-2 below.

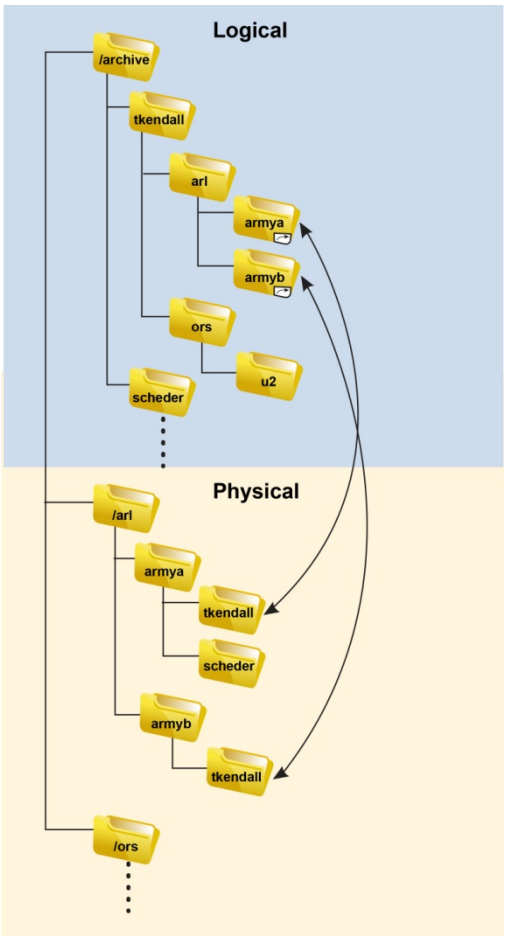


Figure 2-2. Logical to Physical SRB Collection Mapping

Figure 2-3 goes one step further and demonstrates how the physical SAM file system devices are mapped into this Collection organization.

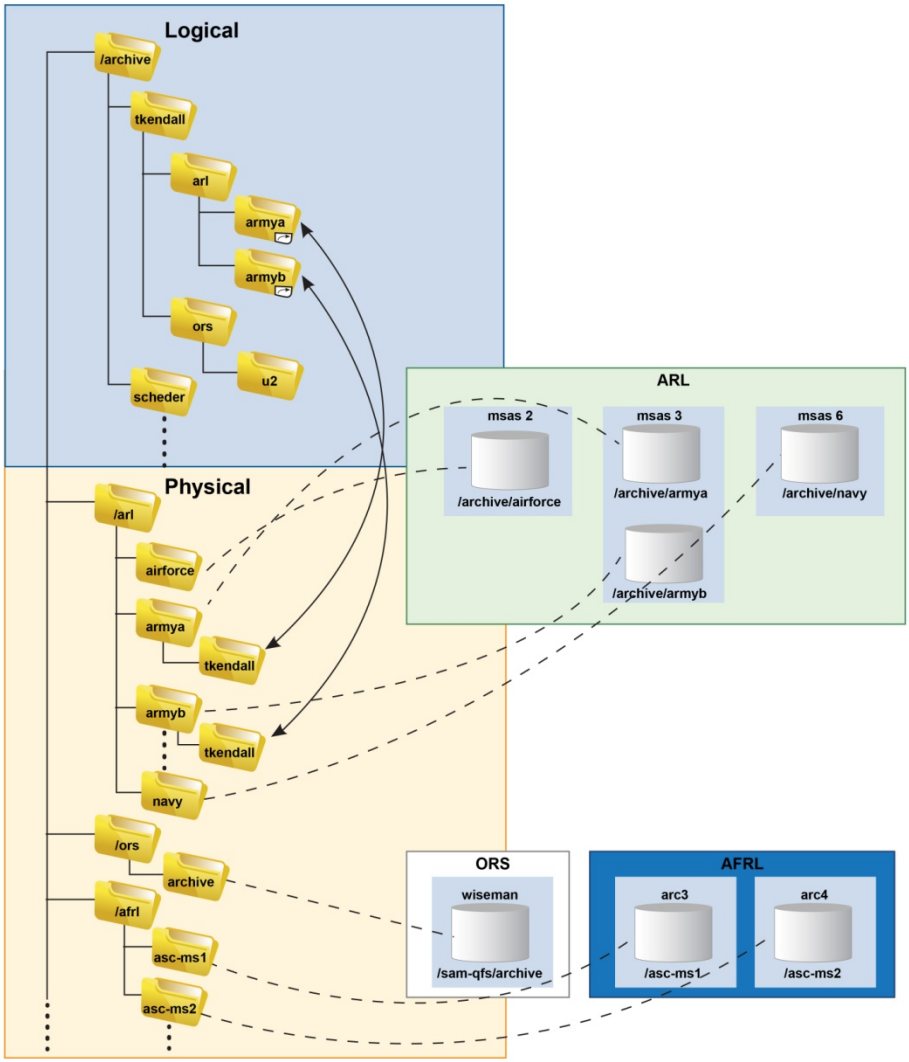


Figure 2-3. SRB Mapping of Collections to Physical Devices

However, the Global Namespace organization does not necessarily always have to match its underlying physical storage. For example, a /hpcmp Collection could be created, which may contain files from multiple sites. It could be organized by users and by projects (e.g., /hpcmp/archive/tkendall or /hpcmp/project/heatr).

Finally, there could be a space for Virtual Collections under /virtual, which would contain named database queries to the MCAT database (e.g., /virtual/expiring_files_user might contain all files that are due to expire within 30 days). The contents of these Virtual Collections are dynamically generated and are filtered based on the users' access permissions. The query for the expiring_files_user Virtual Collection could be:

```
((EXPRESSION.create_age > Admin.Retention_Period - Policy.Warning_Period_User) OR
(EXPRESSION.current_timestamp > Admin.Next_Review_Time – Policy.Warning_Period_User))
AND (Admin.Admin_Hold = 'no') AND (DATA_OBJECT.owner_id = EXPRESSION.current_user_id)
AND (DATA_OBJECT.data_type not like '*collection')
```

3 SECURITY

3.1 Architecture

The SLM architecture addresses the security of information pertaining to the SLM solution components. As described in Section 1, the SLM solution for HPCMP consists of the following components as shown in Figure 1-1.

MCAT Database: MCAT (Metadata Catalog) includes attributes required for the implementation of the Global Namespace and the mapping of a Data Object to storage resources. The MCAT Database also contains all the access permissions and attributes of SRB Data Objects. The MCAT Server ensures that only authorized SRB users have access to the stored data by referencing the data in the MCAT database.

MCAT RAC Nodes: The Oracle Real Application Cluster (RAC) server nodes that host the MCAT Database.

MCAT Servers: MCAT Servers handle requests for MCAT from SRB, and access MCAT RAC Nodes to process requests.

KMS Cluster: Oracle's Key Management System (KMS) manages keys for encrypted data stored on tape drives.

SAM-QFS: SAM-QFS provides a Hierarchical Storage Management System for the SLM solution.

SRB Agent: An SRB Agent provides a variety of data handling services. The SRB Agent listens to a specific IP Address and Port Number. The SAM-QFS Client and SAM-QFS Server include an SRB Agent to support Kerberized communications.

SRB Client: An SRB Client will be installed on an HPC system or Utility Server, and supports establishment of Kerberized sessions between the Client and other SRB components such as an SRB Agent or MCAT Server.

ACSLS: Automated Cartridge System Library System (ACSLS) software enables centralized and efficient sharing of tape library resources with any ACSLS-enabled application (e.g., SAM-QFS), allowing management of multiple libraries from a single point of control.

The security architecture can be broken down in four areas:

1. **Authentication** – With the goal of providing and supporting a strong two-factor authentication, the approach includes the use of passwords (something you know) and smart cards (something you have). Individual and entity authentication is accomplished currently using HPCMP Kerberos V5, PKI, CAC/SecurID, and PKINIT. This SLM solution utilizes this existing authentication framework. The architecture addresses authenticating of all entities that have access to data (see section 3.3).

2. **Data Confidentiality and Key Management** – This refers to providing user-selectable optional data privacy for data in motion and at rest. This includes data movement within a DSRC, as well as data movement across DSRCs using the SRB protocol. Encryption of sufficient strength is employed if the user selects the data privacy option upon establishing the SRB session. Once the Key Management System (KMS) is implemented, data at-rest will also be encrypted (see section 3.4).
3. **Authorization and Access Controls** – This area focuses on restricting and managing access to files and database objects including the requirements for HPCMP access control guidelines (see section 3.5).
4. **Data Integrity** – Data Integrity for data in motion and at rest is addressed by providing for a secured/encrypted message hash generation and verification (see section 3.6).

Compliance with 8500.1 and 8500.2 is specifically addressed as part of the Certification & Accreditation (C&A) Plan.

3.2 Session Initiation

3.2.1 Secure Shell (SSH)

SSH provides an encrypted pipe from the user workstation outside the HPCMP centers to the Login Node inside the centers as shown in Figure 3-1.

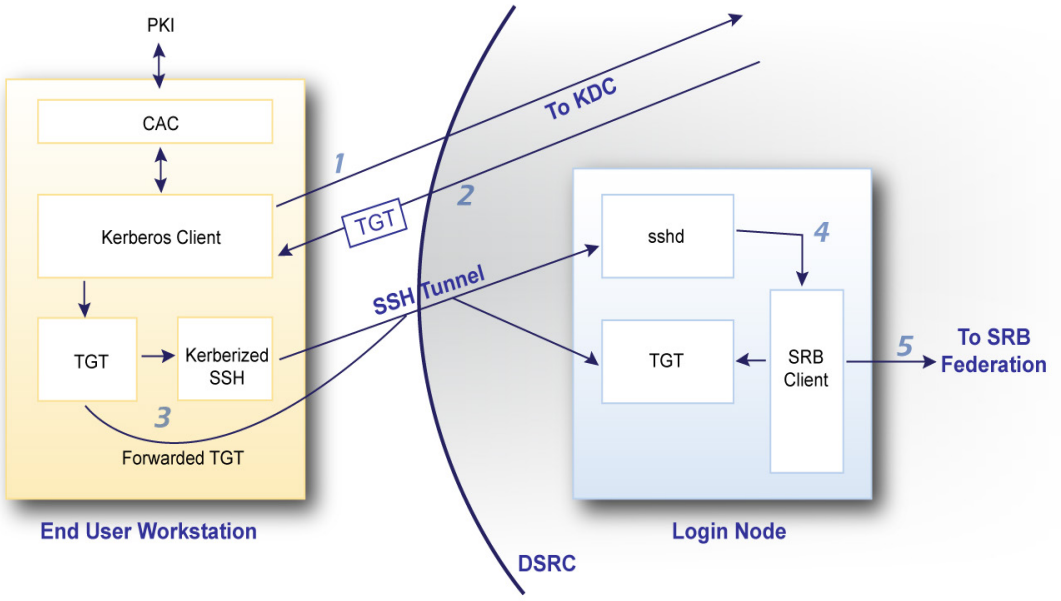


Figure 3-1. Initial Authentication

1. At the end-user workstation, the user provides the credentials (ID, Password, and/or smartcard [CAC with PKI digital certificate] or hToken) to obtain a Kerberos ticket granting ticket (TGT) from the Key Distribution Center (KDC).
2. The TGT is returned to the end-user workstation.

3. The TGT is then forwarded to the Login Node via Kerberized SSH.
4. At the Login Node, the SRB Client (Sinit) is launched by the user's login script.
5. The SRB Client then uses the TGT to establish the Kerberized sessions to the rest of the SRB Federation.

3.2.2 SRB Session Behavior

To run any command-line SRB Clients (i.e., Scommands or Acommands) it is necessary to first establish an SRB session. This is initiated using the Sinit command either launched as part of the login script or interactively by the user. The Sinit command merges the parameters given in the default session file (i.e., "$HOME/.srb/.srbUserSession.def") and the arguments given to Sinit. A user's session is then maintained in the user session file, which is stored in the same directory as the default session file but contains the PID of the shell as an extension (e.g., "$HOME/.srb/.srbUserSession.3056"). This also means that any new shell created within the user's session will need a new user session file. The creation of user session files can be automated using the login script or can be accomplished by calling Sinit explicitly – for example from inside a script.

In the case of HPCMP Kerberos authentication, the SRB session file is only valid in connection with an existing Kerberos credential cache (PIPE or persistent). SRB performs all its communications using a combination of the user session file and the user's Kerberos credential cache. An SRB session file alone – without the Kerberos credential cache – cannot be used to authenticate to SRB using Kerberos authentication.

The SRB session is terminated by calling the Sexit command, which deletes the current shell's user session file. A call to "Sexit –a" deletes the default session file and all user session files. It does not have an effect on the Kerberos credential cache.

When a shell is exited without first calling Sexit, the user session file remains within the user's session directory - only accessible by the user. The user's session file is only usable by another shell with the exact same PID as the previous shell. Even then, the user session file is unusable without the user's Kerberos credential cache, which is always needed in connection with the user session file.

In the (relatively unlikely) event that another shell or application with the same PID as a previously unclosed user session (Sexit was forgotten) is executed by the same user or a user sharing the same user home directory, it would use the existing SRB session file. However, because of Kerberos authentication, the shell/application would actually receive an authentication error unless the Kerberos credential cache for the logged-on user is still present and valid and matches with the SRB session file's user name information.

3.3 Authentication

Both users and system components are authenticated. Table 2-1 shows the authentication approach deployed in various pair-wise interactions. Detailed flows and descriptions are provided in this section.

Table 3-1. Authentication Schemes for Pair-Wise Interactions

Pair-Wise Interactions	Authentication Approach	Comments
End users to SRB Clients	Existing: Smart Cards: CAC/SecurID, PKINIT HPCMP Kerberos V5	SRB Client resides on HPC systems and Utility Servers.
Admin access to SAM-QFS Servers	HPCMP Kerberos V5	Administrator accesses the SAM-QFS server as an unprivileged user then becomes root according to site-specific procedures.
Admin access to MCAT Servers	HPCMP Kerberos V5	Administrator accesses the MCAT Servers as an unprivileged user then becomes root according to site-specific procedures.
SRB Clients to MCAT Servers SRB Clients to SRB Agents SRB Agents to MCAT Servers SRB Agents to SRB Agents	HPCMP Kerberos V5	SRB Agents reside on SLM Servers and utilize the system's Kerberos keytab for authentication; second Kerberos realm for long-running batch processes will provide long-running Kerberos tickets to certain users.
SRB Agents to SAM-QFS	SRB Agent runs as a process on SAM-QFS server; can specify the Root as the owner; and is authenticated by the operating system.	SRB Agents reside on SLM Servers
SAM-QFS to ACSLS	Host-based authentication	ACSLS-internal user names assigned to SAM-QFS clients; SAM-QFS clients are mapped by IP addresses on the ACSLS servers.
MCAT Server to MCAT RAC Node	Database authentication	Explained later in this section
Authentication of KMS components	One-time shared Secrets, KMS-Digital Certificates	Explained later in this section
Oracle to Oracle Authentication	Database authentication utilizing account/password embedded in private database links	Explained later in this section
Cross-realm	HPCMP Kerberos V5 using either TGTs from multiple realms or from cross-realm trust.	

Kerberos keytabs are always used when there is not an interactive person at either end of the communication channel.

3.3.1 Kerberos Support

SRB is integrated with Kerberos v5 authentication and has been tested and demonstrated to work within the HPCMP environment. The HPCMP Kerberos implementation will be exploited as follows:

1. Uses HPCMP Kerberos V5 as the underlying Single Sign-on solution. The solution includes a new library that utilizes the HPCMP Kerberos library without requiring any modifications to the HPCMP Kerberos source code. The new library makes function calls into the HPCMP Kerberos library.
2. The realms in use include local realms (e.g., ARSC.EDU) and global realms (e.g., HPCMP.HPC.MIL and STORAGE.HPC.MIL with an extended ticket lifetime for long running batch processes). The STORAGE.HPC.MIL realm can be used by SRB made available by the work load manager (e.g. PBS Pro) for authenticating long-running batch processes. System level cron scripts can utilize Kerberos keytabs for authentication.
3. Uses HPCMP Kerberos V5 for authentication of users as well as system entities. SRB maintains a list of users and locations (i.e. servers) with their Kerberos principal and realm names in the MCAT. Hence, authentication can be performed with the user's existing Kerberos credential cache without the need for additional authentication. Keytabs are used to support server authentication.
4. SRB users registered in the MCAT are associated with Kerberos principal names and realms. Those associations are established and maintained by a central user management authority.
5. The Kerberos principal name from the credential cache can easily be mapped back to a SRB User and hence, the HPCMP login user is authenticated in SRB. This process has been tested and demonstrated in the DICE test bed using AFRL's Kerberos realm.
6. Uses the Kerberos features that utilize the symmetric keys derived during Kerberos-authentication exchange to encrypt and protect the integrity of the traffic between the authenticated entities.
7. If the Kerberos ticket expires, users must re-authenticate using kinit followed by Sinit.

Figure 3-2 depicts the various ways in which Kerberos sessions are deployed in the proposed solution. By definition, these sessions are strongly authenticated and the traffic may be encrypted whether within a DSRC or across multiple DSRCs.

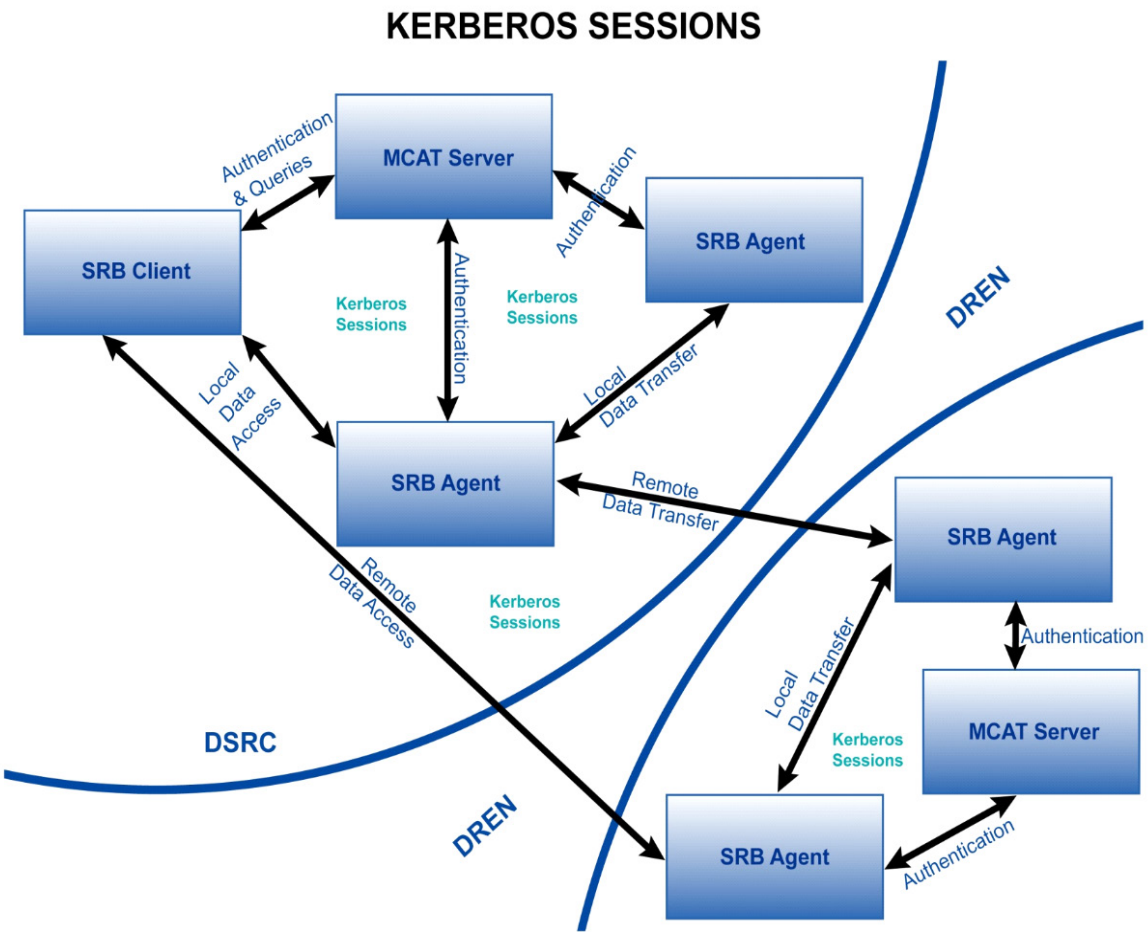


Figure 3-2. Kerberos Sessions within SRB Entities

The figure depicts the various Kerberos sessions used to secure authentication and data privacy. Within a DSRC, there is a Kerberos session between every SRB Client, SRB Agent and the MCAT Servers. This allows the authentication process to perform user operations such as Queries. SRB Agents and SRB Clients communicate with each other across DSRCs. This allows data transfers among remote DSRCs.

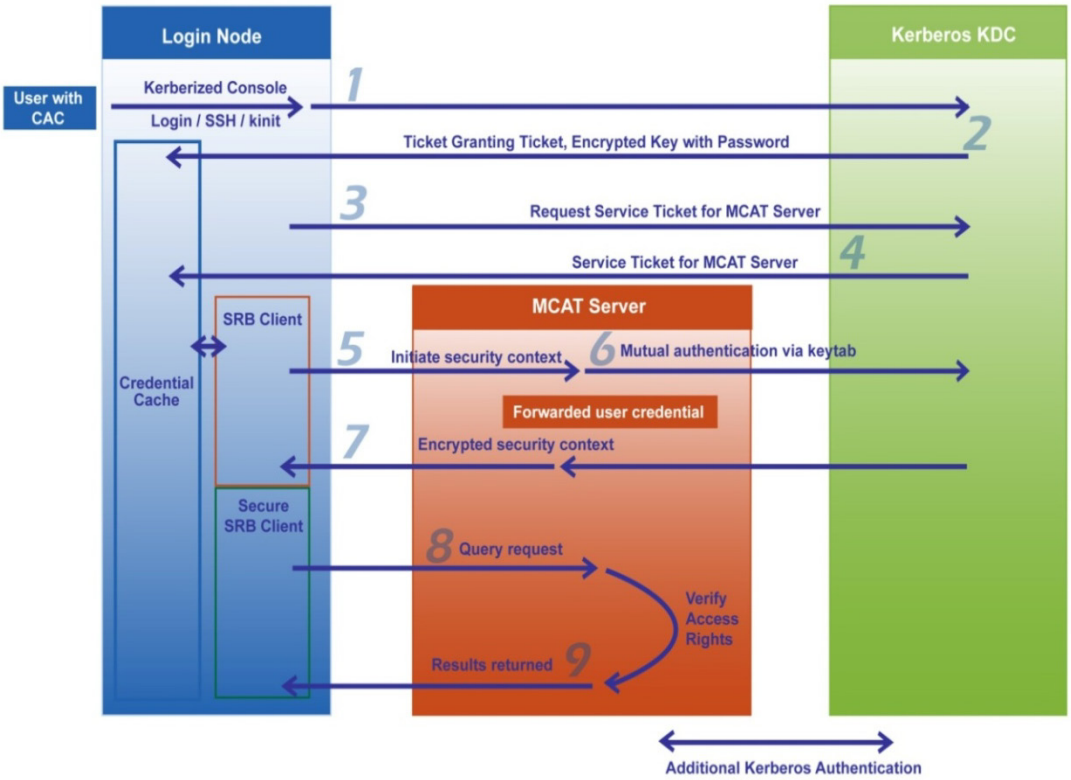


Figure 3-3. Kerberos Authentication Exchanges to Execute a Query Command

Each Kerberos session is secured during the authentication process, and later during the data exchanges within the session as shown in Figure 3-3. An example flow to execute a user query is outlined below.

1. To begin the process, the user logs into the workstation and requests and obtains Kerberos credentials from the Kerberos Key Distribution Center (KDC) via kinit (or krb5.exe in Windows). Then the user logs into the Login Node via a Kerberized SSH session whereby the credentials are forwarded to the Login Node. In the case where the ticket on the Login node has expired, a kinit will need to be executed causing a request to the KDC for a TGT.
2. The Kerberos KDC server sends the TGT to the Login Node, where it is stored in the Credential Cache. The TGT is sent encrypted using the user's password as the key. The TGT lifetime determines the period for which the TGT can be used to obtain Server Tickets. If the TGT lifetime is exceeded, another login (step 1) will have to be performed.
3. Using the TGT, the SRB Client (on the Login Node) requests a Service Ticket with an SRB service principal in order to create a session with the MCAT Server.
4. The Kerberos KDC Server sends the Service Ticket for the MCAT Server back to the SRB Client.

5. The SRB Client establishes the Security Context with the MCAT Server using the Service Ticket from KDC.
6. The MCAT Server establishes mutual authentication with KDC using its keytab file.
7. The user credentials are forwarded to the MCAT Server. The Security Context is established between the SRB Client and the MCAT Server. This means all subsequent communication of user data can optionally be encrypted.
8. The secure SRB Client sends an encrypted (user's) query request to the MCAT Server to verify the access rights for the user and to process the query.
9. Upon authorization, the MCAT Server sends the encrypted results back to the SRB Client.

Figure 3-4 below shows the exchange between an SRB Client and an SRB Agent, which in-turn authenticates with the MCAT Server. The process shown in this diagram also illustrates the exchange between the SRB Agents if SRB Client on the Login Node is substituted by SRB Agent on SLM Server, for example. Note that the SRB Client utilizes this approach for data transfer purposes.

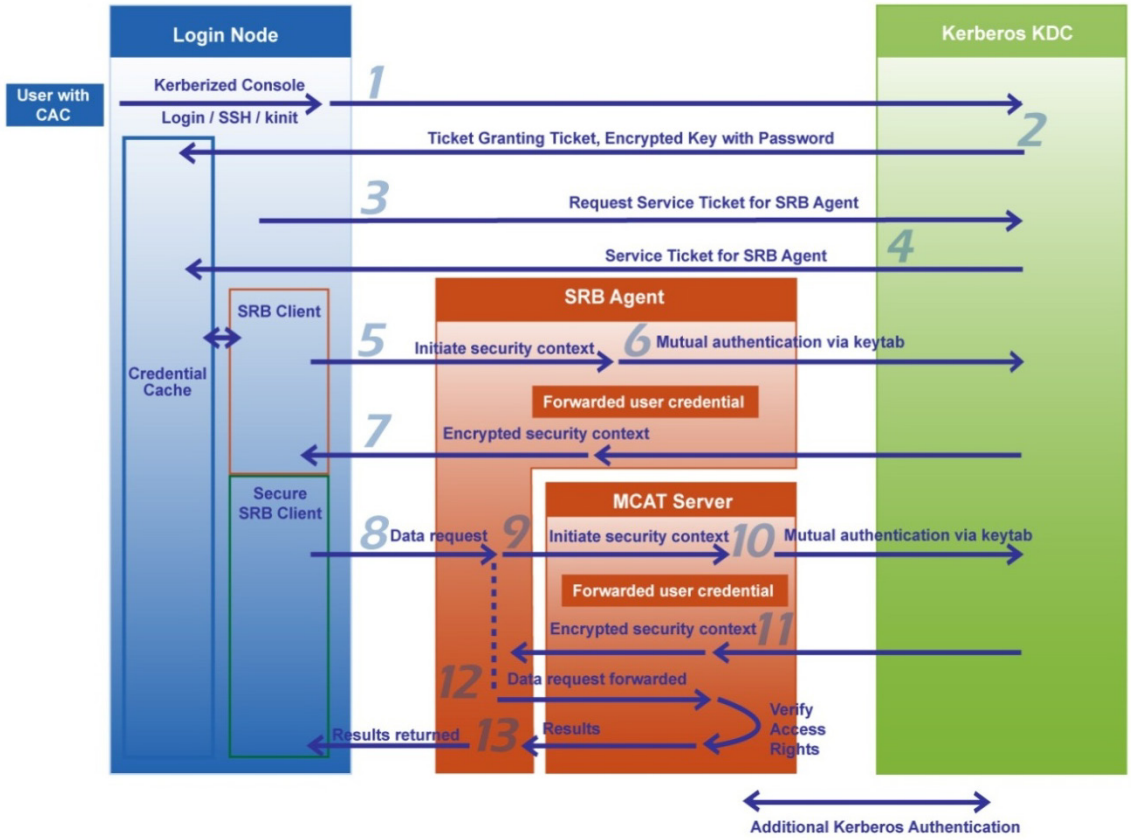


Figure 3-4. Kerberos Authentication Exchanges through SRB Agent for a Data Transfer

1. To begin the process, the user logs into the workstation and requests and obtains Kerberos credentials from the Kerberos KDC via kinit (or krb5.exe in Windows). Then the user logs into the Login Node via a Kerberized SSH session whereby the credentials

are forwarded to the Login Node. In the case where the ticket on the Login node has expired, a kinit will need to be executed causing a request to the KDC for a TGT.

2. The Kerberos KDC server sends the TGT to the Login Node, where it is stored in the Credential Cache. The TGT is sent encrypted using the user's password as the key. The TGT lifetime determines the period for which the TGT can be used to obtain Server Tickets. If the TGT lifetime is exceeded, another login (step 1) will have to be performed.
3. Using the TGT, the SRB Client (on the Login Node) requests a Service Ticket in order to create a session with the SRB Agent.
4. The Kerberos KDC Server sends the Service Ticket for the SRB Agent back to the SRB Client.
5. The SRB Client establishes the Security Context with the SRB Agent using the Service Ticket from KDC.
6. The SRB Agent establishes mutual authentication with KDC using its keytab file.
7. The user credentials are forwarded to the SRB Agent. The Security Context is established between the SRB Client and the SRB Agent. This means all subsequent communication of user data can optionally be encrypted.
8. The Secure SRB Client sends an encrypted data request to the SRB Agent to verify the access rights for the user and to process the data request.
9. The SRB Agent initiates the Security Context with the MCAT Server using its keytab file.
10. The MCAT Server establishes mutual authentication with KDC using its keytab file.
11. The user credentials are forwarded to the MCAT Server. The Security Context is established between the SRB Agent and the MCAT Server. This means all subsequent communication can optionally be encrypted.
12. The data request is forwarded from the SRB Agent to the MCAT Server.
13. The MCAT Server authorizes the data request and the SRB Agent sends the data back to the SRB Client.

3.3.2 Kerberos Algorithm and Key sizes

Kerberos can be configured to support the symmetric AES algorithm at 256 bit key sizes. Kerberos v5 also supports MD5 and SHA1 hashing algorithms. For ECC (Elliptic Curve Cryptography) support on Kerberos, RFC 5349 was issued on September 2008. SRB will support ECC and the requisite key sizes as ECC support becomes available.

Keys need to be renewed at frequent intervals to resist offline brute force attacks. Keys for digital certificates comply with the issuing Certificate Authority (CA) policies. Per current HPCMP security policy, the keys in the keytab file (where the server keeps the secret it has shared with the KDC) are updated every 3 months, but can be updated more frequently.

3.3.3 SAM-QFS Server Access

The long term goal is not to allow direct access to SAM-QFS systems unless they are System Administrators. However, users could be granted direct access to meet an operational need (such as at the NAVY DSRC). System Administrators log in as regular users, using the Kerberos login along with appropriate smart cards. Once logged in, users obtain root access in the site-preferred manner.

3.3.4 Authenticating KMS Entities

The Oracle Key Management System (KMS) 2.0 uses authentication mechanisms that are internal to the system. It does not yet integrate with Kerberos or Smart Cards. There are three types of entities in a generic KMS environment.

- One or more (up to 20) Key Management Appliances (KMAs). The KMAs implement a distributed database of keys and, as such, manage the details of key life cycle, the delivery of encryption keys to encrypting tape drives, and the replication of keys among appliances. They also implement the key management security policies.
- Tape drives (generically referred to as “encryption agents” in the KMS architecture) which use encryption keys to encrypt and decrypt user data. Currently, the only encryption agents implemented for KMS 2.0 are tape drives. Therefore, for clarity in this document, the term tape drive will be used as a synonym for the more generic KMS term “encryption agent.” Most statements made below about a tape drive in a KMS 2.0 context applies generically to any entity that implements KMS 2.0 encryption agent logic.
- Users of the KMS Manager management GUI. The KMS Manager is a Java-based application that serves as an administrative client to the KMA. It can be used to configure, control, and monitor the KMA. Depending on their assigned roles, users can perform different operations.

A KMS cluster begins its life when an administrator deploys the first KMA. Additional KMAs or tape drives are then added incrementally to the cluster via a secure enrollment process, details of which are given below. Administrative users, regardless of role, access the KMS cluster for administrative purposes via the KMS Manager software. Their authentication is via a password database in the KMS. Details of initial enrollment and subsequent authentication processes are as follows.

3.3.5 MCAT Server to Oracle Authentication

The communication path between the MCAT Servers and Oracle goes from the MCAT Server process, through the Oracle database driver, through the Oracle Client Interface (OCI) API – which represents an Oracle client – into the Oracle server.

The recommended authentication mechanism for MCAT Server access to Oracle is database authentication. Authentication is established across the Oracle public network from the OCI

libraries to the MCAT RAC Nodes. The database password should be changed on a regular basis.

`$SRB_HOME/mcat.config` on the MCAT Server(s):

```
ORACLE       DEBUG_LEVEL             "0"
ORACLE       LOG_FILE                "oracle.log"
ORACLE       LOG_FILE_SIZE           "10000000"
ORACLE       LOG_FILE_GROUP          "srb"
ORACLE       LOG_FILE_MODE           "0640"
ORACLE       OPERATION_ALERT_TIME    "120"
ORACLE       DB_VERSION              "ORACLE"
ORACLE       DB_LIBRARY              ""
ORACLE       DB_NAME                 "mcatdb"
ORACLE       DB_AUTH_SCHEME          "DATABASE"
ORACLE       DB_USER                 "srb"
ORACLE       DB_AUTH                 "[kLAD8Zz/ja2hxB+t/6K+A0qZ3VkR4eL9B4aDVZ2yRoY=]"
ORACLE       DB_SCHEMA               "srb"
```

3.3.6 Oracle to Oracle Authentication

The recommended database authentication mechanism for Oracle to Oracle access (utilized by cross site synchronization - Oracle Streams - processes) is the Oracle private database link in which Oracle credentials - account and password - are embedded in the link definition and leveraged by Oracle to authenticate access. It is recommended that access to the Oracle listener, required to support remote database access, be restricted.

3.3.7 SRB External Authentication

Since a second Kerberos realm for long-running batch processes is planned, the need to use SRB External Authentication is unnecessary for this solution and will therefore be disabled. However, the following paragraphs will try to provide some understanding of SRB External Authentication.

If External Authentication is enabled, SRB will trust the underlying operating system of an SRB Agent with the local system user authentication. SRB extracts the UID, GID, user name, and group name of the SRB Client process owner and tries to find a match within its MCAT database using SRB user names and aliases. Once a match is found, the user is authenticated into SRB.

By default, External Authentication is disabled. External Authentication is a configuration option for each individual SRB Location.

In order for SRB External Authentication to function securely, all primary group names or GIDs need to be unique and consistent across all sites. UIDs may be inconsistent. This is required because the SRB user names and aliases need to be globally unique.

3.4 Data Confidentiality and Key Management

There will be underlying transfer of data resulting from SLM operations. Sensitive and classified data should be protected against unauthorized access or exposure. The breakdown of the SLM

components in this solution is as shown in Figure 3-5. The communication between these components is described in the sections below.

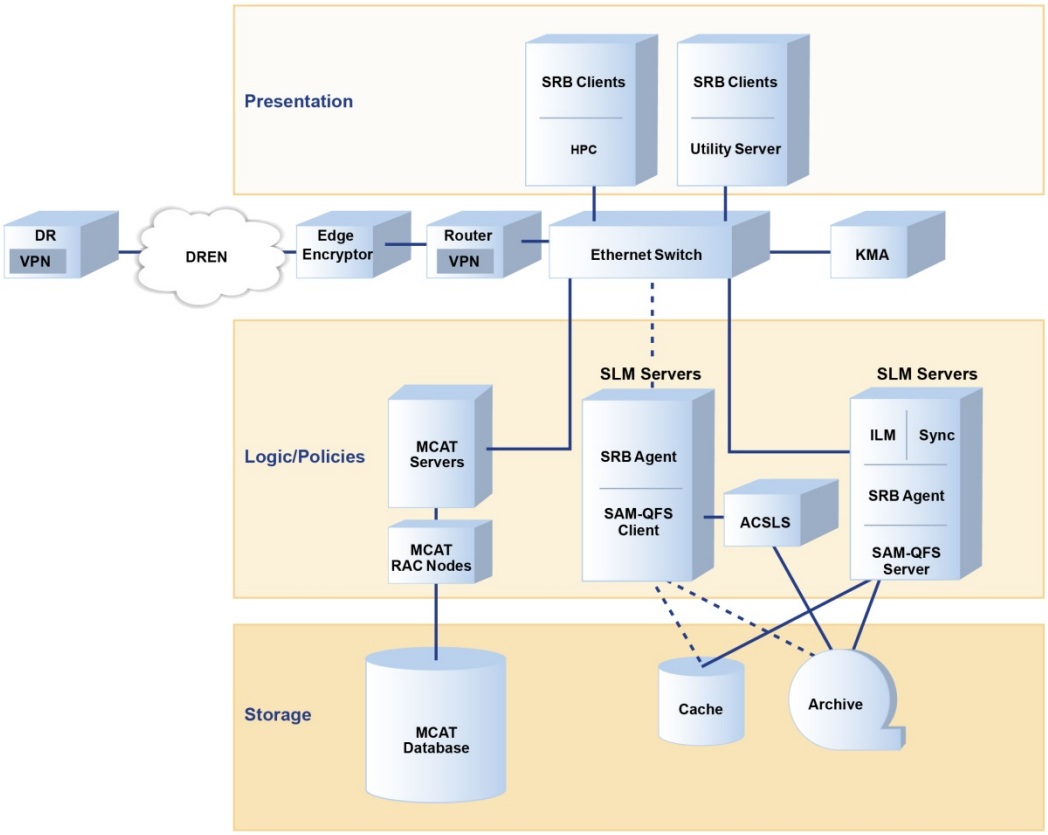


Figure 3-5. Interactions Among SLM Components

The following scenarios are encountered when moving data within and across the DSRC environments. Encryption can be optionally employed as outlined in the table below.

Table 3-2. Data Confidentiality Scenarios for Data in Motion and at Rest

Data-in-Motion	Data Confidentiality Mechanism
Metadata Access	User-SRB Client: Kerberized SSH SRB Client-MCAT: Kerberos
SRB Client to SRB Agent SRB Agent to SRB Agent	Kerberos
Oracle to Oracle	SSL or VPN/IPSec over DREN
DSRC to DSRC	Kerberos, VPN/IPSec over DREN
DSRC to DR Site	VPN/IPSec over DREN
Data-at-Rest	**Data Confidentiality Mechanism**
User Data – Tape drives	Encrypting Tape Drives, KMS

3.4.1 Securing Data in Motion

The design goal is to ensure that all data that moves using the Kerberos-supported session can be optionally encrypted by the method chosen by the user or administrator. SRB supports plain text and two Kerberos data transfer schemes: PLAIN_TEXT, KERBEROS_INTEGRITY and KERBEROS_ SECURE. Both Kerberos schemes will perform cryptographic integrity checking. Additionally, KERBEROS_SECURE will also encrypt the communication channel. If the TGT expires during a long running operation, the KERBEROS_INTEGRITY and KERBEROS_ SECURE options will terminate the session. Hence, for long running, non-sensitive operations, it is recommended that the PLAIN_TEXT option be selected.

During the Kerberos authentication exchanges between the client and the server, symmetric keys are established for session data encryption. These keys are used to afford symmetric encryption of session data. This feature clearly ensures that symmetric encryption is provided for data while in motion.

In the diagrams that follow (Figure 3-6 thru Figure 3-8) the user workstation represents the system in use by the user (could be a Windows laptop, UNIX workstation, etc), and the Utility Node represents either an HPC system or a Utility Server.

3.4.1.1 Metadata Access

This type of data flow occurs during queries and database updates, which do not involve file system access.

- Steps 1 and 8: The traffic between the end user and the Login Node (SRB Client) is secured by Kerberized SSH.
- Steps 2 and 7: The traffic between the SRB Client and MCAT Server can optionally be secured by Kerberos.
- Steps 3 and 6: The SRB Server and DB driver are within the same machine and operate in the same address space.
- Steps 4 and 5: The DB driver and MCAT DB operate via Oracle OCI library calls traveling across the Oracle public network.

The following diagram illustrates this flow:

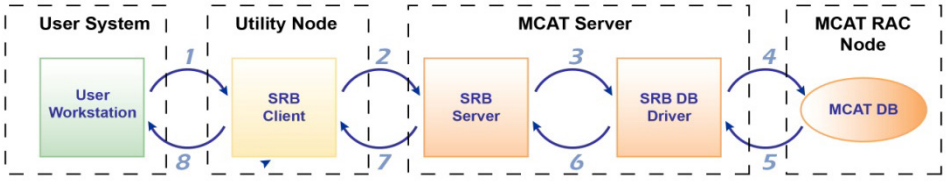


Figure 3-6. Metadata Access Data Flow (Post Authentication)

3.4.1.2 Data Access between SRB Client and SRB Agent

As soon as file system access is initiated, at least one SRB Agent gets involved in the data flow. The following figure illustrates this type of flow:

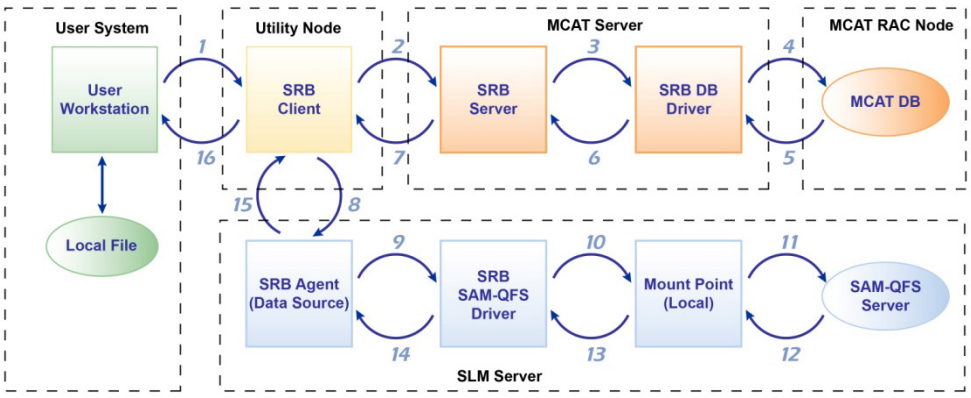


Figure 3-7. Data Flow through SRB Agent (Post Authentication)

In Figure 3-7, the command to transfer data, such as Sput, is communicated by the user at the workstation. This discussion assumes that the various Kerberos authentications have taken place and the secure data transfer channels have been established if desired.

1. Steps 1 and 16: The transfer of a file between the User System and the Utility node is via SCP or SFTP using the Kerberized SSH tunnel. The request for Sput/Sget/Scat and the final response are via Kerberized SSH.
2. Steps 2 and 7: Communication is via a Kerberos or plain text session.
3. Steps 3 and 6: The SRB Server and DB driver are within the same machine and operate in the same address space.
4. Steps 4 and 5: The DB driver and MCAT DB operate via Oracle OCI library calls traveling across the Oracle public network.
5. Steps 8 and 15: Communication is via a Kerberos or plain text session separate from 2, 7.
6. Steps 9 thru 14: Communication takes place within the same machine.

3.4.1.3 Data Access between Two SRB Agents

The following data flow occurs when data is transferred within the system not involving a file from the User Workstation but involving two SRB Agents. Examples of these commands include Sreplicate, Scp, and Smv.

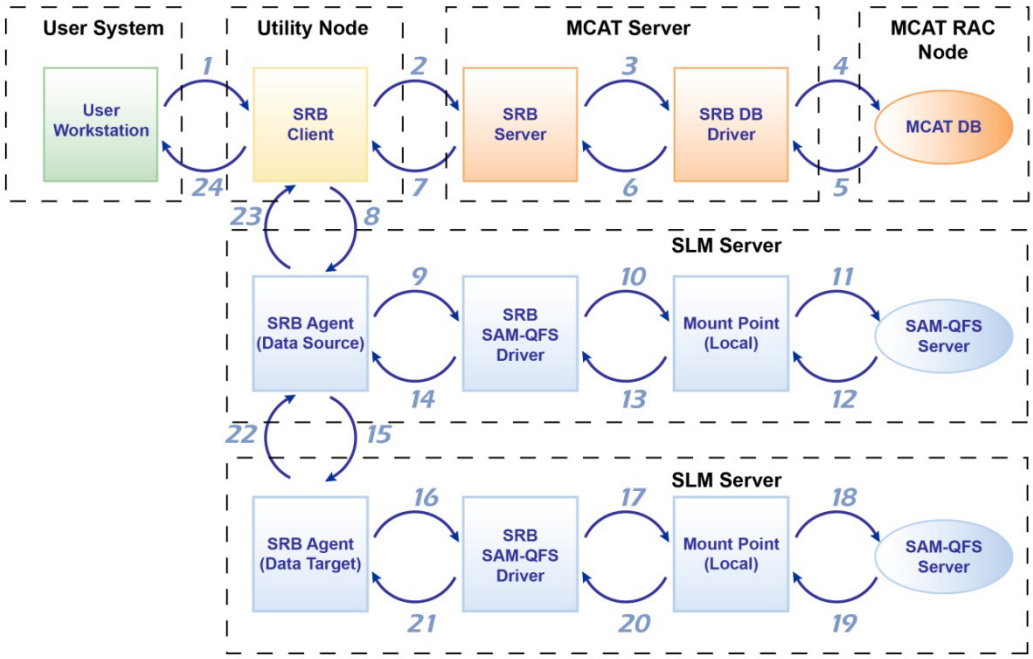


Figure 3-8. Data Flow between Two SRB Agents (Post Authentication)

In Figure 3-8, the process for accessing data through an SRB Agent is repeated.

7. Steps 1 thru 14: Same as above with the exception that there is no file transfer in Step 1. However the file is sourced from this SLM Server.
8. Step 15: The file is transferred to another SLM Server via the respective SRB Agents using Kerberos or plain text protocol.
9. Steps 16 thru 21: Communication takes place within the same machine.
10. Steps 22 and 23: The success or error messages are communicated back to the SRB Client over Kerberos or plain text.
11. Step 24: The result is communicated to the user over Kerberized SSH.

3.4.1.4 Oracle to Oracle

The communication between the Oracle MCAT databases from site to site can be protected using either VPN/IPSec or SSL. Either mechanism could be applied between the database servers. The usage of Oracle Advanced Security is not planned. The preferred method is to utilize an external encryption appliance.

3.4.1.5 DSRC to DSRC

There may be data traffic from or between DSRCs. The data traffic from one DSRC to another is an SRB Kerberized session between an SRB Client and an SRB Agent, or an SRB Agent and an SRB Agent as outlined in Figure 3-7 and Figure 3-8. Additionally, the traffic between the DSRC boundaries (edges) is protected by VPN/IPSec over DREN.

3.4.1.6 DSRC to DR Site

For data traffic from the DSRCs to the DR site is protected by VPN/IPSec over DREN. Since the data move takes place between two Oracle Solaris systems, it can optionally use the built-in IPSec in Solaris. It is believed that IPSec would work transparently with SAM-QFS but it could be a task under the *Engineering Studies and Analysis Services* portion of the solicitation to qualify SAM-QFS for the usage of IPSec to secure shared QFS metadata and disk archiving.

3.4.2 Securing Data at Rest

The only data for which encryption at rest is practical is data resident on SAM archive tapes. The SLM solution recommends hardware-encrypting T10000 tape drives and Oracle's KMS to meet the requirement for encrypted data at rest for removable media (tape). The Oracle StorageTek T10000 tape drives support the AES (Advanced Encryption Standard) algorithm at 256 bits. Data at rest on disk cache will not be encrypted.

3.4.2.1 KMS Architecture

The KMS is architected as a cluster consisting of one to 20 KMA, either entirely located at single site or spread across multiple sites. The KMAs work together over an IP network to generate, distribute and manage keys for encrypting tape drives that are clients of the cluster. A KMA will generally service key requests for local tape drives, however, when required, they can service requests from drives at any site in the cluster.

3.4.2.2 Single-Site Architecture

To ensure availability of encrypted data, the KMS is designed not to put a key into use until a copy of the key has been made. Where the cluster consists of a single KMA, copies can only be made via the KMS manual process for backing up keys. When multiple KMAs are joined together into a KMS cluster, secure, inter-KMA replication processes ensure that keys exist in multiple locations before being put into use. For automated operations, therefore, two KMAs per cluster are required

Performance is not a major driver for having multiple KMAs in a cluster. Two KMAs are expected to provide excellent performance in even the most demanding environments (as measured by tape loads per second).

It is required that all tape drives in a cluster be able to route IP traffic to every KMA that may be required to provide it keys. Similarly, the management GUI workstations must be able to route IP traffic to all KMAs via which they will manage the cluster. A simple implementation of KMS networking places all three KMS elements on a common "management" network and this architecture is shown in Figure 3-9. Dedicated network hardware is the most secure way to implement such a management network. In the absence of dedicated hardware, isolating the KMS traffic onto a VLAN is a second option. While KMS traffic is strongly encrypted, the least desirable option places the elements of the KMS cluster on a non-private network.

Note that because the KMS Manager GUI never directly accesses tape drives, it is acceptable to further tighten security in KMS environments by isolating these two elements via network architecture or security rules. The current general recommendation, then, is that HPCMP sites employ a dual-KMA cluster. Sites wanting greater availability should configure a third KMA. Where implementations support physical separation of KMAs within a site, overall availability can be enhanced. Network latency from KMA to tape drive should be considered to ensure optimal performance.

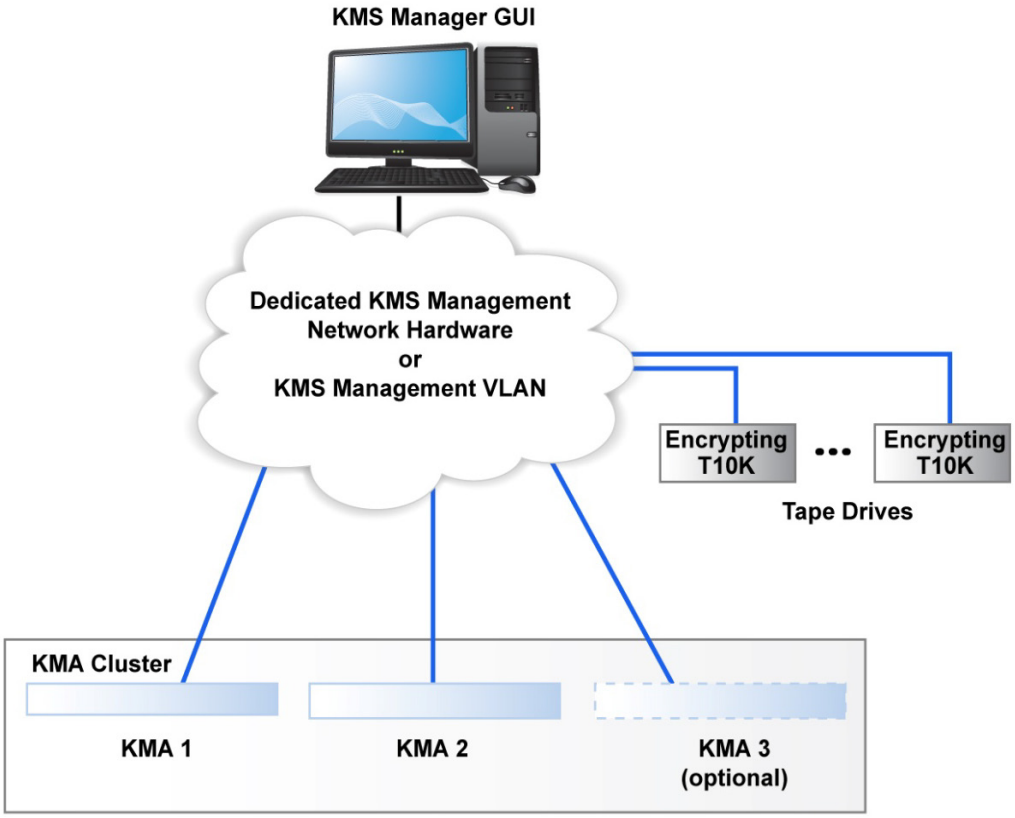


Figure 3-9. Recommended Single-Site KMS Architecture

3.4.2.3 Multi-Site Architecture

Automated protection against loss of all KMAs at a site can be achieved by establishing a multi-site KMS cluster. In such a cluster, keys are automatically and transparently replicated to remote site KMA's. However, unless media is being sent to remote sites, multi-site KMS clusters will only be useful if all of a site's KMAs fail or are incapacitated, but the encrypted media is still available.

Should two or more sites choose to implement a multi-site cluster, the same order of preference exists for containing KMS traffic. The order of preference is: dedicated network hardware, followed by VLANs.

A second alternative is to maintain single site clusters but establish a second site as a key sharing partner. Key sharing allows keys and any associated tape media to be securely exchanged between partners. It is based on purpose-specific, key-transfer public/private key pairs that are created and managed inside each site's KMS. Any sending site is loaded with the public key of the receiving site and uses that public key to encrypt a key bundle for export. The receiving KMS can then use its private key-transfer key to unlock the imported bundle from the sending site. A multi-site architecture is depicted in Figure 3-10.

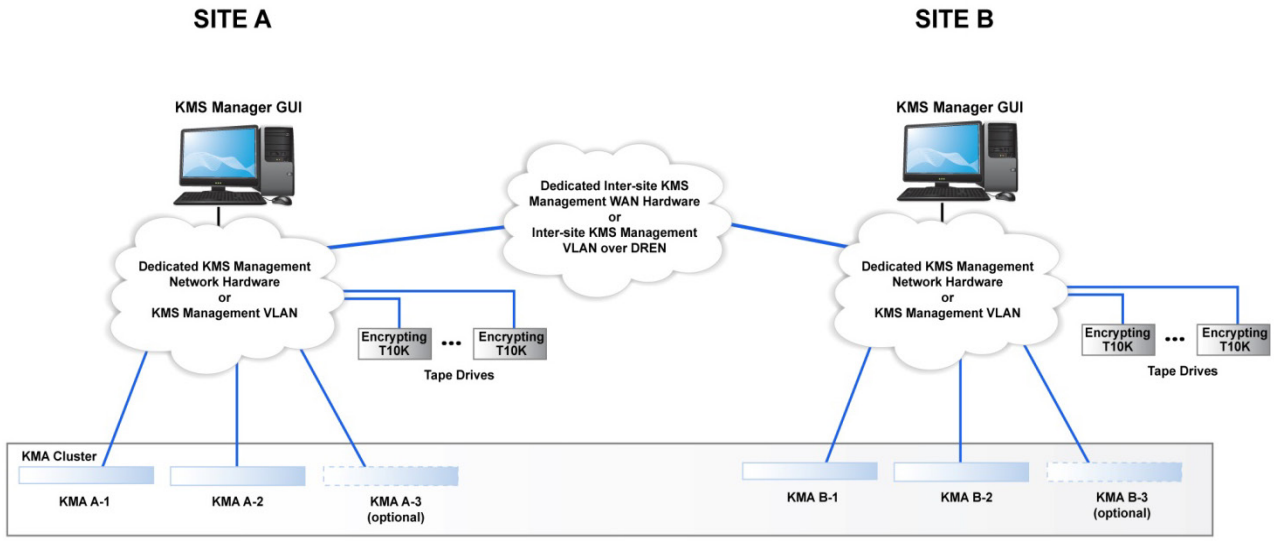


Figure 3-10. Multi-Site KMS Architecture

3.4.2.4 Key Management

The encryption keys are managed through the Oracle StorageTek KMS 2.0. The significant characteristics of the KMS with regards to keys are:

- Never stored as clear text or on the same media as the data it encrypted.
- Encrypted in the KMS.
- Encrypted for transport from the KMS to another storage device.
- Encrypted when copied to backup devices.

3.4.2.4.1 Encryption Key Sizes

The Sun KMS system provides strong, end-to-end encryption designed to meet government standards. The “key” transmission and tape encryption use AES algorithms using 256-bit key sizes. The AES is a strong block cipher encryption algorithm used in CCM (Counter with CBCMAC) mode to protect user data, enabling data privacy as well as reliable and efficient authentication.

3.4.2.4.2 Key Renewals

For KMS, key life cycles are based on the NIST 800-57 guidelines to which the KMS conforms. New keys have a lifecycle that play out according to the encryption period (the period during

which a key can be used to encrypt) and cryptoperiod (the period during which a key can be used to decrypt data) designated by the policy associated with the key. Both periods are configurable to be as short as one minute or may span multiple years. With short encryption periods relative to the time it takes to fully fill a tape with data, a tape may have multiple associated keys. Alternately, with longer encryption periods defined, data on a cartridge may be secured by a single key.

NIST guidelines specifically allow keys past their cryptoperiod to continue to process data. Once past the cryptoperiod, these keys are, however, eligible to be destroyed. The NIST guidelines do not provide any basis for destroying keys based on time.

To achieve the effect of renewing keys for data at rest, a rewrite of encrypted data would be required. In a SAM context, this would mean a rearchive operation on all encrypted media. Since the keys are used to encrypt all the data, a renewal of the key will required re-encryption of the tape. This could be accomplished at time of tape technology refresh.

3.4.2.4.3 Key Recovery

The KMAs in the KMS cluster serve to back up each other to allow recovery of lost keys. As long as the cluster lives, the keys will live. Secure exports of keys outside the cluster are possible as an extra measure of protection. If all copies of a key are lost, the data it encrypts is unrecoverable.

3.4.2.4.4 Key Storage on Disk

In the KMS architecture, the Sun Crypto Accelerator (SCA 6000), a FIPS-140 Level 3 component (see next section), wraps and manages all key material. Keys are never stored or sent in the clear.

3.4.2.4.5 Key Destruction

Security policies may require the destruction of keys that are no longer in active use. The KMS can support this requirement with its ability to report keys that have been used for a particular Volume Serial Number (VSN). Because the KMS-internal identifier for a VSN changes upon each relabeling of the media (as happens with a SAM recycle operation), administrator reports allow distinguishing between keys used before the last relabeling (and therefore no longer in use) and keys used after the last relabeling. This distinction is useful to ensure that the KMS administrators delete only keys that are no longer in active use.

3.4.3 Crypto Accelerator

The SCA 6000 PCI Express crypto accelerator adapter has two potential roles in HPC systems.

First, where KMS is deployed, the implementation of a KMA transparently employs the SCA 6000 for encryption operations.

Additionally, the SCA 6000 may have a role where non-Kerberized software entities, such as the SAM remote disk clients and servers, currently communicate over HPCMP data networks. The Solaris IPsec facility offers one means of injecting encryption and (host-based) authentication into these communications. While IPsec operations can be entirely performed in software in Solaris, cryptographic computations can divert a significant portion of CPU cycles from the primary workload. The SCA 6000 adapter integrates via kernel driver with Solaris and allows offloading of IPsec cryptographic processing.

Note that qualification of the above SAM-QFS components with IPsec and the SCA 6000 has not yet been performed and would require an *Engineering Studies and Analysis Services* effort.

3.4.4 Securing Metadata in Store

There are three sets of metadata containing sensitive information stored in the MCAT Database.

1. The audit tables (user, resource, data audits).
2. The access control tables (user, resource, data, configuration, and metadata attribute ACLs).
3. User Metadata that are deemed sensitive.

This information is restricted to a single database user (i.e. “srb”) that the MCAT Server uses to access the MCAT Database. Additionally, the SRB user can only add new records, and cannot modify or delete records in the audit tables. It is recommended that Oracle table-level auditing be enabled for the audit and access control tables.

None of the data in the MCAT tables are encrypted by default. However, columns in Oracle can be encrypted selectively using the approach given on the following website: http://www.oracle.com/technology/oramag/oracle/05-sep/o55security.html.

The downside to column encryption and decryption is that additional CPU cycles are needed. Also, if the encrypted columns are used in LIKE-queries, the database would perform a full table scan on those encrypted column values, which can be very expensive for large tables.

Conversely, the Oracle Streams is not affected by encrypted columns because it relies on SQL transactions logs that are sent to each site before column encryption.

3.5 Authorization and Access Control

3.5.1 Authorization Overview

Authorization and access to the SLM solution is through the establishment of an access control mechanism consisting of the following:

1. There are two locations where access permissions are stored:
    a. The MCAT holds Access Control Lists (ACLs), ownership, curatorship, group, and inheritance metadata about all data. This information is initially synchronized from the underlying file systems.
    b. The file system holds UNIX access permissions such as mode, owner, and group

owner. SLM continues to apply this information to the file system upon file ingest.

2. The access control scheme applies to both individual end users and encompasses system components, such as ILM or SRB Sync Daemons. Daemons authenticate and obey the access restrictions of the user under which they run.
3. Conversely, no one without specific authorization is allowed access. For example, if user A has "write" privileges to user B's files, then user A can modify user B's files. On the other hand, if user A has no rights (i.e., no ownership, no curatorship, no entries in user B's ACLs) on user B's files, user A would not be able to see user B's files, and could never modify them.
4. Access to modify Access Control Lists, ownership, or inheritance is restricted to:
    a. Owner of the respective Object
    b. Curator of a collection tree
    c. Super user
    d. System group all_data_all
    e. Anyone designated by the above
5. Access to modify curatorship is restricted to:
    a. Curator of a collection tree
    b. Super user
6. By default, a user has no privileges. All privileges are established by adding users, groups, or domains to the ACL of an SRB object.
7. Access is classified in the categories listed below. Only one can be chosen per ACL entry:
    a. Null – this is the default when no ACL entry is present and hence no access is granted.
    b. Deny – access is explicitly denied in the ACL.
    c. Execute – not used (reserved for future use).
    d. Read – read access.
    e. Write – read and write access.
    f. All – read, write, delete, change ACL, and other modifications.

If a user is granted access to an object through multiple ACL entries, (e.g. group, domain) then the least restrictive category takes precedence. For example, if the user admin@nirvana has "read" access to an object but the nirvana domain has "write" access the result will be that the admin@nirvana user will also have "write" access. The "deny" constraint always takes precedence. For example, the user admin@nirvana will be denied access to an object if either the individual user (admin@nirvana), or the entire domain (nirvana), or a group that admin@nirvana belongs to, is denied access.

3.5.2 User Home Collection Access

User Home Collections are structured as shown in section 1.2. Access to User Home Collections is assigned in the following manner:

- The /archive/<userName> Collection is owned by super@root but the user has "read" permissions. Users are not permitted to write into the Collection at this level because all files would otherwise be physically written into the root of the SAM file system.
- The /archive/<userName>/<siteName> Collections are owned by super@root but the user has "read" permissions via inheritance from the /archive/<userName> Collection.

- The /archive/<userName>/<siteName>/<fsLabel> links are owned by super@root but the user has “all” permissions to the links via the “all” permissions on the original Collection (i.e., /<siteName>/<fsLabel>/<userName>).
- The /<siteName>/<fsLabel>/<userName> Collections are owned by the user who also has “all” permissions. These “all” permissions are inherited by all lower-level Data Objects and Collections.
- ACLs can also be managed at the Project Collection level. This permits users to have different access controls for their Home and their Project Collections.

3.5.3 Mandatory Access Control (MAC)

The system prevents unintentional data migration from high to low to avoid unintended exposure of sensitive information. There are four scenarios under which this can theoretically occur:

1. Data is physically moved or replicated to an Open Research (non-Sensitive) system via SRB.
2. Access permissions to sensitive ORS files and directories are given to uncleared users, groups or domains.
3. Sensitive metadata is unintentionally exposed to unauthorized entities.
4. A sensitive SRB object is physically moved or copied to a resource not cleared to hold sensitive information.

SRB 2010, and later, feature a Mandatory Access Control (MAC) mechanism. This mechanism is based on assigning access flags to users during account setup. The flags are based on the users’ security classification, such as ITAR for U.S. citizens or Sensitive but Unclassified (SBU) when users complete their National Agency Check (NAC).

Correspondingly, each SRB Object has a mandatory access flag that specifies the minimum level of security classification required by a user for accessing the object. If a user creates new Data Objects or Collections, the default classification of the user will be assigned to those objects.

Finally, the SRB Physical Resource (i.e. file system) must have a security classification that is at least as high as the classification of the objects being created on it. For example, the ORS SAM-QFS file system would not have the SBU flag set and, therefore, would never be able to accept any SBU objects.

A user may consciously declassify objects by downgrading their security classification flag. Through this mechanism, it will be possible for non-ORS users to share files with ORS users.

On the other hand, ORS users who do not generally have the SBU security classification will not be able to access SBU files from *any* site.

Benefits of this approach include an HPCMP-wide mandatory access control of all objects and users. Also, if a user mistakenly moves an object to a less secure enclave, the users with a lower security classification still cannot access the object. Similarly, if the ownership of the

object is changed to a user with a lower security classification, the new owner will have no access.

3.6 Data Integrity

Data Integrity provides for protection of data from unauthorized data alterations such as data stream modifications. A *hash* (a one-way function that computes similar to a “checksum” and helps verify unauthorized data modification) is computed at a time when the data is created. The hash can be checked at a later time to ensure that no modifications have taken place. Ways to protect the hash include encrypting it with the originator’s Private Key. Data Integrity applies to both data in motion and data at rest.

3.6.1 Data Integrity for Data in Motion

For data-in-motion, Kerberos V5 offers data integrity protection. RFC 3961 (3) outlines the algorithms and provisions for Kerberos V5 to deploy encryption as well as data integrity.

So, besides using Kerberos for cryptographic authentication, SRB also uses it to ensure the integrity of all data and control channel communications. SRB supports two Kerberos data transfer schemes, which are user selectable: KERBEROS_INTEGRITY and KERBEROS_ SECURE. Both schemes will perform cryptographic integrity checking. In addition to that, KERBEROS_SECURE will also encrypt the communication channel at a performance cost.

3.6.2 Data Integrity for Data at Rest

For data at rest, there is currently no support provided. For data integrity for stored data, an application may be developed.

It is possible to design and implement an Integrity Daemon in a future design effort (through an *Engineering Studies and Analysis Services* effort), which would periodically stage and calculate hashes and checksums of files. The type of algorithm would be customizable – MD5, SHA1, CRC32. The hashes and checksums will be stored in the MCAT Database and associated with each file. This will provide for an end-user comparison with utilities such as md5sum or sha1sum.

SAM-QFS can be configured to generate a checksum when an archive copy is created and verify that checksum when that archive copy is retrieved at a later date. SAM-QFS offers an additional option to use checksums to restage and verify an archive copy immediately after creation. However, this option is not recommended as it imposes significant additional overhead.

3.7 Secure Migration to the SRB Environment

Migration to the SRB environment is achieved while maintaining the following goals outlined below. A key goal is to ensure that security of the data is not compromised. As described, the Sync Daemon is used for addition of users and registration of files into the MCAT database during the migration process.

- Preserve file and directory ownership.
- Maintain user accounts.
- Preserve ability for users to hold accounts at each site.
- Preserve current file locations.
- Support movement to a global Kerberos realm, while allowing other local Kerberos realms to persist (i.e. ARSC.EDU).
- Use secured, controlled and automated process to register files into the MCAT database and associate them with authorized users.

For a discussion on SRB and local file system interaction please refer to section 6.1.

3.7.1 SRB User Management

The implemented SRB solution includes the following characteristics for managing SRB users at multiple sites with different UNIX user accounts at each site:

1.0 User accounts with different authentication schemes – for SLM only the Kerberos authentication scheme is implemented.
2.0 User domain management with users being able to be part of more than one domain – for example, user tkendall can be part of both the ARL.HPC.MIL and HPCMP.HPC.MIL domains.
3.0 User group management with users being part of many groups. Groups span domains and sites.
4.0 User alias management with the ability to give users different names for different domains – for example, kendall@ARL.HPC.MIL is an alias to tkendall@HPCMP.HPC.MIL.
5.0 Single unique user ID, which ties user with multiple site identities back to a single individual – for example, kendall@ARL.HPC.MIL and tkendall@HPCMP.HPC.MIL tie back to the SRB user ID 1000.
6.0 Metadata, such as telephone number and e-mail, etc., is stored for each user.
7.0 User access control lists are implemented so that groups of users can be managed by different administrators.
8.0 Mandatory Access Controls have been implemented through the use of User, Resource, and data classification (see section 3.5).

SRB manages user, group, domain, and alias objects. They are all maintained solely in the MCAT database. Figure 3-11 illustrates one possible relationship between those objects.

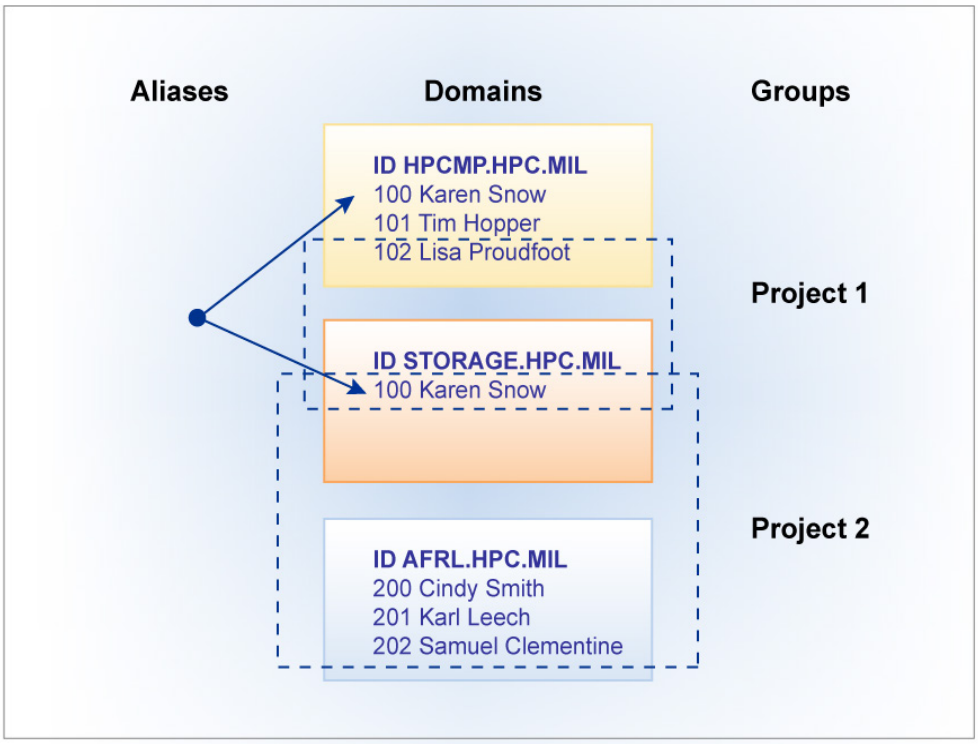


Figure 3-11. SRB Federation

SRB Users represent individual persons. Each SRB User belongs to one and only one primary SRB Domain. However, SRB Users can belong to multiple SRB Domains using aliases. In the diagram above, Karen Snow belongs to HPCMP.HPC.MIL as her primary domain but has an alias in the STORAGE.HPC.MIL domain as well. This allows her to log onto SRB using either HPCMP.HPC.MIL or STORAGE.HPC.MIL as her SRB Domain name (i.e., Sinit –user "Karen Snow@HPCMP.HPC.MIL"; or Sinit –user "Karen R. Snow@STORAGE.HPC.MIL"). The alias mechanism further allows her to authenticate via any of two Kerberos realms but still be identified as the same individual despite her SRB User name being different between the realms.

SRB Domains represent security domains such as Kerberos realms, Microsoft Active Directory domains, or Internet domains.

SRB Groups represent common roles that tie multiple users together – projects, roles of responsibilities, restricted access groups such as ITAR, or organizational divisions. They do not typically map directly to UNIX groups.

3.7.2 UID/GID to SRB User ID Mapping

Common UIDs/GIDs across all sites are not going to be a requirement. The following is an example of the approach for mapping local UIDs to global user names in SRB:

1. Primary user stubs are created in MCAT containing the global Kerberos User Principal. For example, 'scherrsj@HPCMP.HPC.MIL' might be created with SRB user ID 222.

2. User Aliases are associated with the primary user stubs for all the local user names. So if a global user 'scherrsj@HPCMP.HPC.MIL' has accounts at AFRL and ARL, then two user aliases would be created in MCAT 'scherrsj@afrl' and maybe 'scherr@arl'. Notice how the user names between aliases may vary among sites and between the global and local names.
3. The Sync Daemon is configured to run at a particular site - say AFRL.
4. The Sync Daemon starts and reads-in the entire site's user names and SRB user IDs building a lookup table - including the alias 'scherrsj@afrl' with SRB user ID 222.
5. As the Sync Daemon gets ready to register a file into SRB it determines that the file is owned by UID 111.
6. Using a system API, the Sync Daemon determines that UID 111 translates into the local user 'scherrsj'.
7. The Sync Daemon goes to its lookup table and determines that local user 'scherrsj' maps to SRB user ID 222.
8. The Sync Daemon issues the call to register the file in MCAT with the SRB user ID 222 as the owner.
9. Taking the same approach, the files owned by root at a particular site (say AFRL) would be mapped to an SRB user (or alias) called ‘root@afrl’. The SRB user ID is determined using the same lookup table and the file is registered with that SRB user ID as the owner.
10. The ‘root@afrl’ alias could be associated with a single “real” user – probably an administrative user at AFRL. Alternatively, there could be a “real” SRB user called ‘root@afrl’.
11. There is also the possibility that a local UID cannot be mapped to an SRB user ID. This has two potential causes: a) the UID has no user name entry in /etc/passwd or b) the user name determined through /etc/passwd does not have a corresponding primary or alias entry in SRB. If this happens, the Sync Daemon will report an error message in its log file and the file registration will fail.

Refer to section 6.1.3 for a discussion on GID mapping. Local group owner names are stored as an attribute in the MCAT database and can be reapplied to the file upon retrieval from SRB.

3.8 Ports and Protocols

The following tables provide the ports and protocols used by each of the SLM solution components.

3.8.1 Oracle RAC

Table 3-3. Oracle RAC Default Ports and Protocols

Application	Protocol	Port #	Scope	Comments
ORACLE DB Service	TCP	1521	orapub, Center	Configurable using tns ora files; utilized by SRB MCAT Server.
ORACLE Streams	TCP	1521	orapub, Outside Center	See NOTES below
OEM	TCP	1158	SLM unit	Configurable with command: “emca -reconfig ports”

NOTES: For Oracle Streams replication, database links need to be setup such as

```
CREATE PRIVATE DATABASE LINK &DB_2..&DB_2_DOMAIN USING
'(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=&DB_2_SCAN_HOST)(PORT=1521))
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=&DB_2)))';
```

The port and protocol can be configured in those links. The default is TCP and 1521.

3.8.2 SRB Federation

Table 3-4. SRB Client/Server Communication Paths

From	To	Protocol	Port #	Scope	Comments
SRB Client	MCAT Server	SRB/TCP	5625	Center	Configurable in MCAT
SRB Agent	MCAT Server	SRB/TCP	5625	Center	Configurable in MCAT
SRB Client	SRB Agent	SRB/TCP	5625	Outside Center	Configurable in MCAT (for performance reasons, this connection is direct).
SRB Agent	SRB Agent	SRB/TCP	5625	Outside Center	Configurable in MCAT
MCAT Server	SRB Agent	SRB/TCP	5625	Outside Center	Configurable in MCAT

Table 3-5. SRB Drivers Communication Paths

FROM	TO	Protocol	Port	Scope	Comments
MCAT Server SRB DB Driver	MCAT RAC Node MCAT DB	OCI/TCP	1521	orapub	Configured using mcat.config file.
SAMFS Driver	SAM Server	---	---	System	See SAM-QFS ports Table 3-11 below

Table 3-6. Other SRB Communication Paths

From	To	Protocol	Port	Scope	Comments
Client/Agent/MCAT	DNS Server	UDP	53	Center	host name to IP address resolutions
Client/Agent/MCAT	KDC	TCP	88/750	Center	outside center for cross realm authentication

The following figures illustrate the scope (private, public, within system, center, or outside center) of the communication between SRB Clients, Agents, and MCAT Servers in the various system modes.

Figure 3-12 shows that no cross-center communication exists in Isolated Enclave Mode. All communication takes place within the Ora Private, Ora Public, and Center networks.

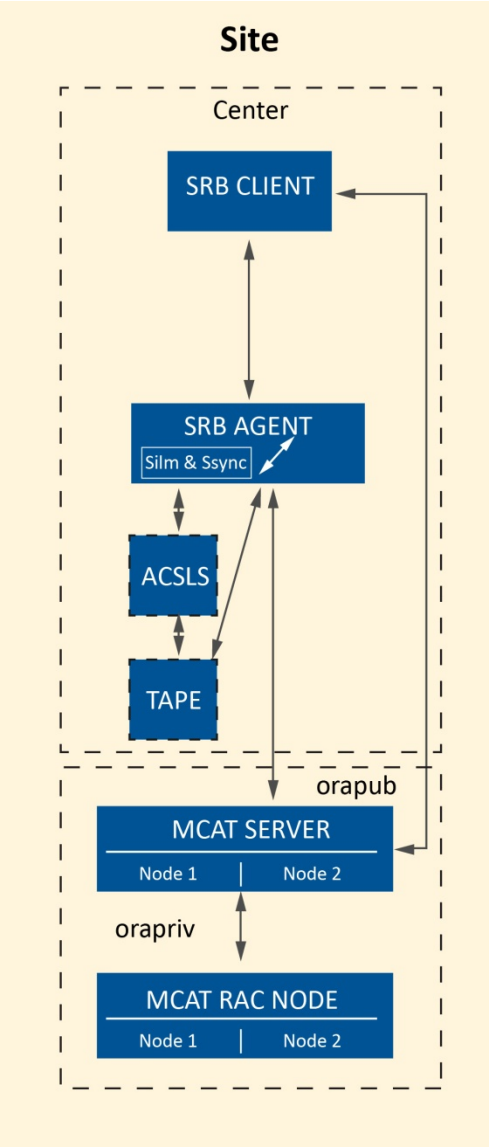


Figure 3-12. Isolated Enclave Mode (no Cross-Center Communication)

Figure 3-13 shows cross-center communication exists in Isolated Enclave Mode if SRB Clients are manually pointed towards a remote site's MCAT Server. This may be the case if users need to query or exchange files with remote sites. In those cases communication would be established outside the center.

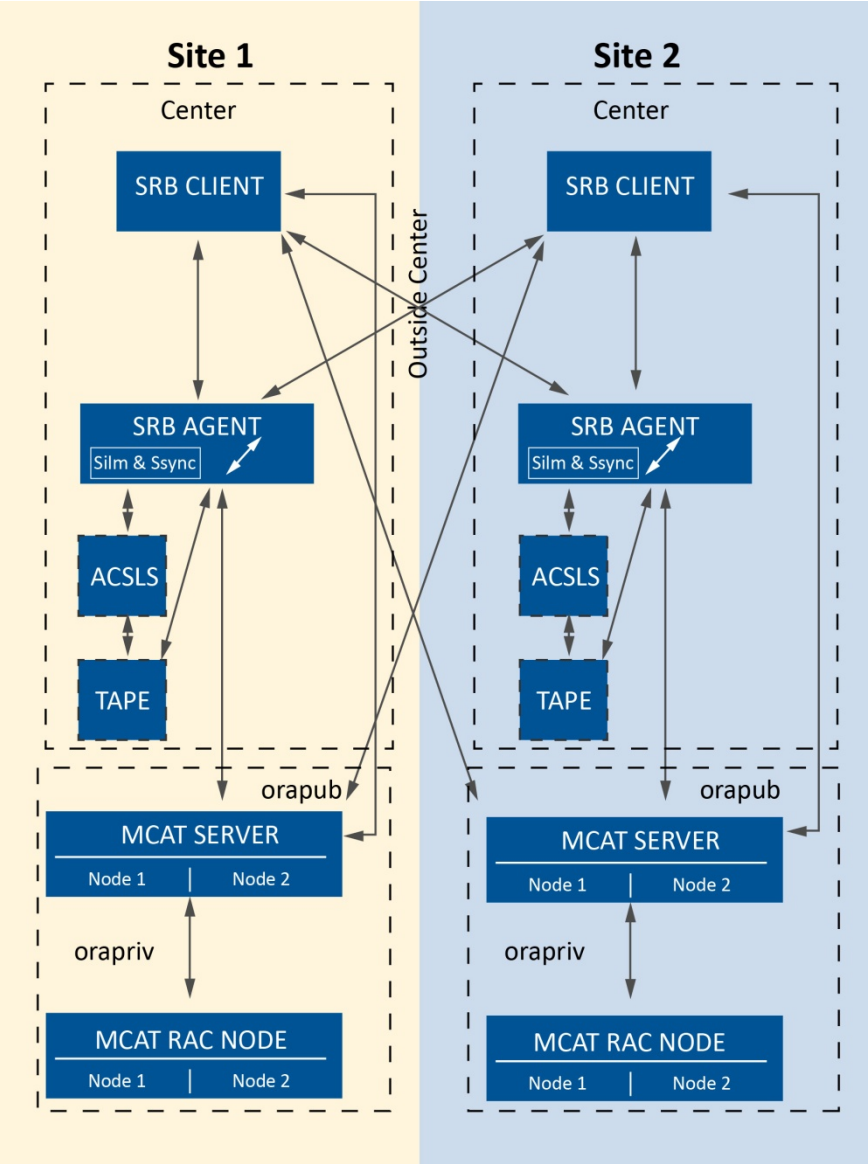


Figure 3-13. Isolated Enclave Mode (with Cross-Center Communication)

Figure 3-14 shows cross-center communication exists in Replicated Mode during normal operations. SRB Clients can communicate with SRB Agents locally or with any remote site for the purpose of file transfers. SRB Agents can communicate amongst themselves for the same purpose. MCAT RAC Nodes communicate via Oracle Streams for metadata synchronization.

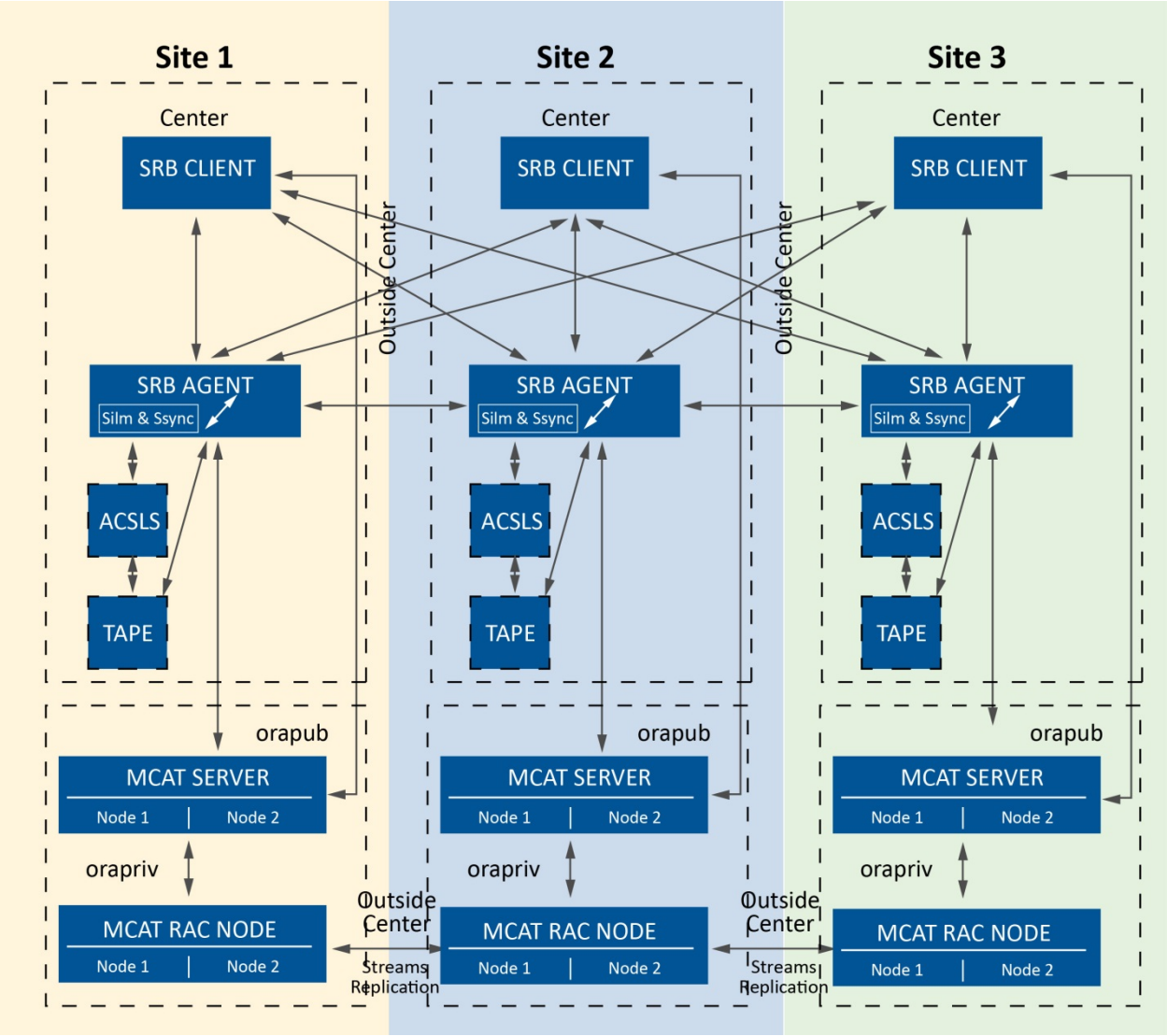


Figure 3-14. Replicated Mode (with Cross-Center Communication)

Figure 3-15 shows cross-center communication exists in Replicated Mode in a failover scenario. It acts similar to Replicated Mode under normal operations but the communication between SRB Clients and MCAT Servers at the failed site is supplanted by a connection to remote MCAT Servers.

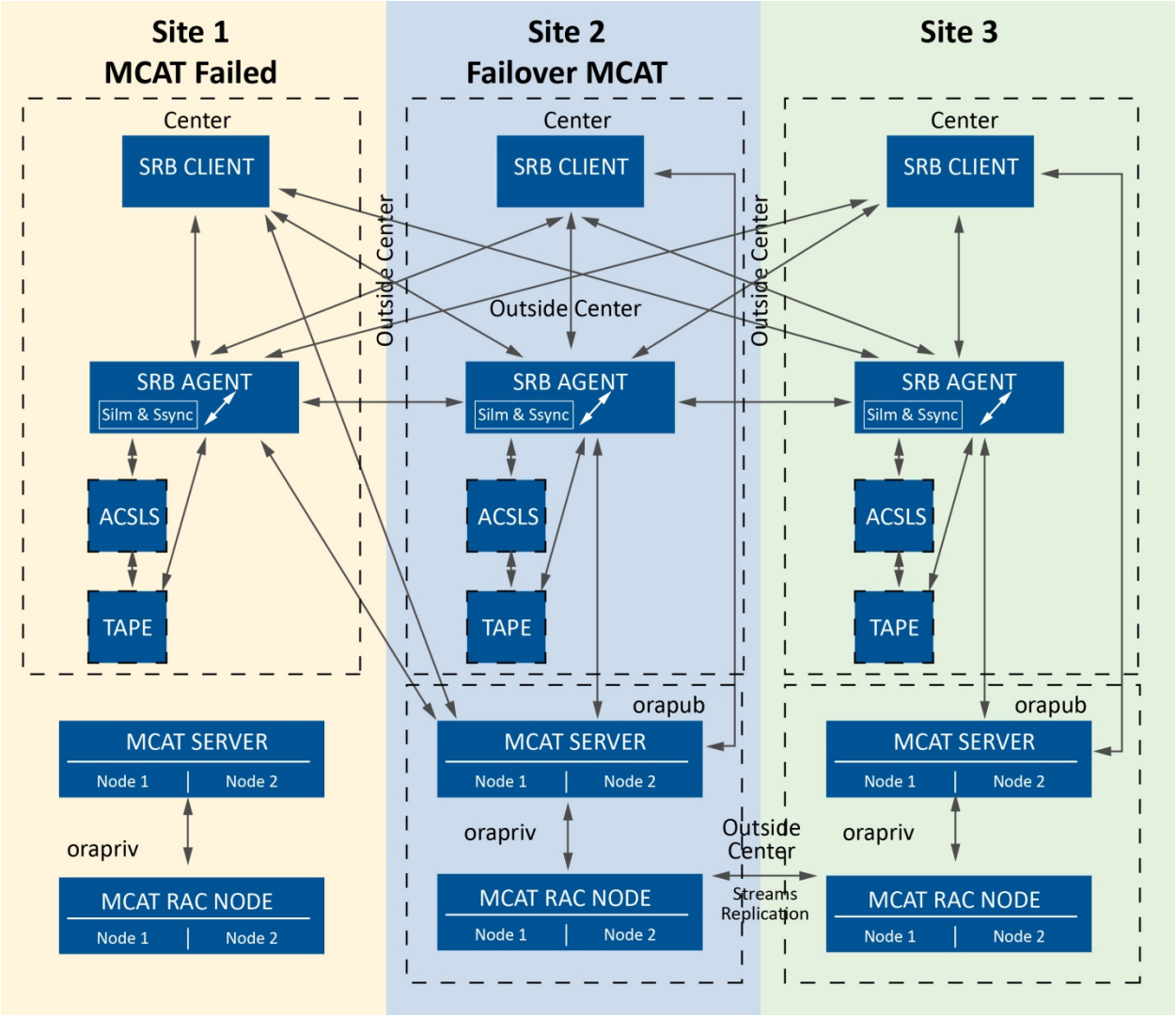


Figure 3-15. Failed-Over Replicated Mode (with Cross-Center Communication)

3.8.3 Key Management System Ports and Protocols

The KMS ports and protocols pertain to the following entities:

- A cluster of 1 to 20 KMAs.
- Attached tape drives.
- Administrative users who access the KMS cluster using the KMS Manager software GUI to connect to a KMA.
- KMA Embedded Lights Out Management (ELOM) service processor.

3.8.3.1 KMA Network Ports and Protocols

The management interface of a KMS refers to the main network port of the KMA over which normal KMS activities are performed. The ports in Table 2-7 are used by the KMAs on their management network interface in a KMS cluster:

Table 3-7. KMA Ports and Protocols

Port	Protocol	Direction	Description
22	SSH over TCP	Listening	Disabled by default and in normal operations. Enabled when Technical Support access is explicitly enabled.
53	DNS over TCP/UDP	Connecting	Not required but sites may choose to configure KMAs to use DNS.
68	DHCP over UDP	Connecting	Not required but sites may choose to configure KMAs to use DHCP.
123	NTP over TCP/UDP	Listening	Recommended but not required. Time synchronization between KMAs must be maintained, whether automatically via NTP or manually.
161	SNMP over UDP	Connecting	Not required. When at least one SNMP Managers is defined to the KMS cluster, the KMAs will send SNMP Informs to the IP address of that SNMP Manager(s). The version of SNMP is selectable. Currently, versions 2 and 3 are supported.
3331	HTTP over TCP	Connecting / Listening	KMS CA Service. A KMA connects to this port only when it is being joined to cluster. Once a part of the cluster, the KMA listens on this port enabling entities to be joined to the cluster via this KMA. In listening mode, the KMS CA Service provides the root certificate to the client (another KMA, tape drive or KMS Manager software user) to be used for encryption by the client.
3332	HTTPS (TLS) over TCP	Connecting / Listening	KMS Certificate Service. A KMA connects to this port only when it is being joined to cluster. Once a part of the cluster, the KMA listens on this port to facilitate clients (another KMA, tape drive or KMS Manager software user) being joined to the cluster via this KMA. In listening mode, the KMS Certificate Service performs the passphrase authentication on the client. Upon success, it generates and delivers a client certificate and key pair to be used for future secure communication by the client.
3333	HTTPS (TLS) over TCP	Listening	KMS Management Service. The KMS Management Service provides services for KMS Manager GUI requests.
3334	HTTPS (TLS) over TCP	Listening	KMS Agent Service. The KMS Agent Service provides services for key requests from tape drives.
3335	HTTPS (TLS) over TCP	Connecting / Listening	KMS Discovery Service. The KMS Discovery Service provides reachable KMAs in a cluster information to a client. This information may be different for each KMA.
3336	HTTPS (TLS) over TCP	Connecting / Listening	KMS Replication Service. The KMS Replication Service operates from KMA-to-KMA to keep all KMAs in sync.

3.8.3.2 KMA ELOM (Console) Ports and Protocols

All KMAs have an ELOM port that may be optionally used for network-based graphical console access. A console session is required for initial KMA configuration, and in particular, to initialize the appliance via the Quick Start program. The ELOM port may be used for this purpose or alternately, console access may be achieved without the ELOM port via keyboard and monitor directly attached to the appliance.

Normal operations *do not* require the KMA ELOM port to remain connected to any system or network. Should the ELOM port remain on a network, the following ports and protocols are used.

Table 3-8. KMA ELOM ports and Protocols

Port	Protocol	Direction	Description
22	SSH over TCP	Listening	Allows command line administration if needed.
53	DNS over TCP/UDP	Connecting	Not required. Only used if configured.
68	DHCP over UDP	Connecting	Not required. Only used if configured.
69	TFTP over UDP	Connecting	ELOM firmware updates.
80 or 443	HTTP or HTTPS over TCP	Listening	Web server interface for monitoring and control application. Launch point for Remote Console.
161	SNMP over UDP	Listening/ Connecting	If SNMP monitoring for ELOM port is desired.
623	IPMI over UDP	Listening	IPMI
8890, 9000, 9001, 9002, 9003	TCP	Listening	Remote Console Java Webstart application. (Remote KVM)

3.8.3.3 Tape Drive Ports and Protocols

Tape drives connect to KMAs via the following ports in a KMS environment.

Table 3-9. Tape drive ports and Protocols

Port	Protocol	Direction	Description
3331	HTTP over TCP	Connecting	KMS CA Service. The tape drive connects to this port when it is being enrolled into the cluster. See Table 3-7 for more detail.
3332	HTTPS (TLS) over TCP	Connecting	KMS Certificate Service. The tape drive connects to this port when it is being enrolled into the cluster. See Table 3-7 for more detail.
3334	HTTPS (TLS) over TCP	Connecting	KMS Agent Service. The tape drive connects to this port to get key requests serviced.
3335	HTTPS (TLS) over TCP	Connecting	KMS Discovery Service. The tape drive connects to this port to get a list of all reachable KMAs.

3.8.3.4 KMS Manager Workstation Ports and Protocols

The workstation that runs the KMS manager connects via the following ports.

Table 3-10. KMS Manager Workstation Ports and Protocols

Port	Protocol	Direction	Description

Port	Protocol	Direction	Description
3331	HTTP over TCP	Connecting	KMS CA Service. The KMS Manager GUI connects to this port when a user logs in. See Table 3-7for more detail.
3332	HTTPS (TLS) over TCP	Connecting	KMS Certificate Service. The KMS Manager GUI connects to this port when a user logs in. See Table 3-7for more detail.
3333	HTTPS (TLS) over TCP	Connecting	KMS Management Service. The KMS Manager GUI connects to this port on a KMA to manage the KMS cluster.
3335	HTTPS (TLS) over TCP	Connecting	KMS Discovery Service. The KMS Manager GUI connects to this port to determine all reachable KMAs in a cluster.

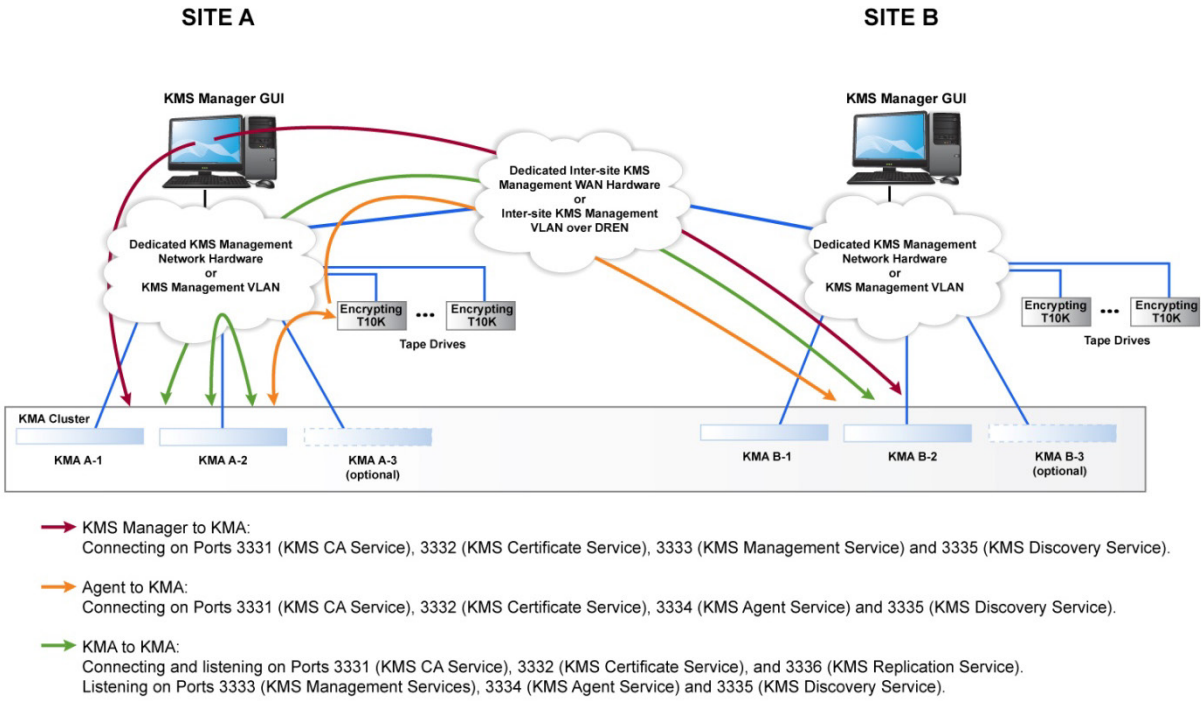


Figure 3-16. 2-Site KMS Replication Cluster Example

3.8.4 SAM-QFS Ports

Port usage in SAM-QFS will remain as currently configured at the DSRCs and is shown in the following table for completeness.

Table 3-11. SAM-QFS Ports

Function	Port	Protocol	Entities	Description
SAM port manager daemon	7105 / TCP	SAM custom	SAM server, Various SAM client software	The SAM port manager listens on the specified port and acts as a “clearing-house” daemon for all SAM-related services, whether those services are entirely local or network-enabled. Its supporting role for other TCP/IP connectivity is described in the entries below.

Function	Port	Protocol	Entities	Description
QFS metadata services	7105 / TCP One free TCP port on server as selected by Solaris bind() called with 0 for port argument. One free TCP port on client as selected by Solaris bind() called with 0 for port argument.	QFS custom	QFS metadata server, QFS metadata clients	The QFS metadata server coordinates shared file system operations in a share QFS environment. The QFS metadata server program acquires a port on startup and registers it with the SAM port manager. QFS metadata clients first connect to the well known port of the SAM port manager on the QFS server where they are then redirected to connect to the actual QFS metadata port.
SAM-Remote	7105 / TCP Chosen from 5000-5999 / TCP (First attempt to bind is at 5000 + SAM-internal server ordinal. Successive ports are tried, incrementing by 10 each time, until an available one is found.)	SAM custom	SAM server running SAM-Remote, SAM-Remote client	A SAM-Remote client system can configure a logical tape drive locally that is actually backed by a remote tape drive on the separate SAM-Remote server. Delivery of data to the remote tape drive happens via the SAM-Remote protocol. Upon startup, the SAM-Remote server acquires its port and registers it with the SAM port manager daemon. When initiating a server connection, the SAM-Remote client contacts the port manager at the well known port 7105. The port manager redirects the client to the actual SAM-Remote server port that was registered and the client reconnects. HPCMP: The SLM architecture does not depend on or use this service directly, and it is *not* expected that any site employs this service in their existing SAM-QFS implementations. Remote archiving of data in this environment is understood to happen via sam-rftd, documented below.
SAM-rftd	7105 / TCP One free TCP port on server as selected by Solaris bind() called with 0 for port argument. One free TCP port on client as selected by Solaris bind() called with 0 for port argument.	SAM custom	SAM server running SAM-rftd, SAM-rftd client	A SAM-rftd server offers a file system as a target for archiving by a SAM-rftd client on a separate system. Transfer of data between client and server is via the SAM-rftd protocol. Similar to SAM-Remote, the SAM-rftd server registers the port it acquires on startup with the SAM port manager. SAM-rftd clients first connect to the SAM port manager and are then redirected to the actual SAM-rftd server port where it reconnects. HPCMP notes: This service is in widespread use in the customer environment as this is the means by which data is sent to the DR site.
SAM API RPC service	111 / TCP (aka RPC) 5012 / TCP	RPC + SAM custom	SAM server, SAM API clients	This API allows remote systems that are not themselves SAM servers to initiate SAM operations remotely. When enabled in the SAM configuration file, the SAM API RPC service registers with rpcbind and attempts to bind to port 5012 / TCP. HPCMP notes: The SLM architecture does not depend on or use this service. It is documented in this table for completeness.

Function	Port	Protocol	Entities	Description
(NOT REQUIRED) SAMDB server port	3306 / TCP (MySQL) A different port can be configured in the MySQL server configuration and samdb.conf config files.	MySQL	SAM server using SAMDB database, MySQL server hosting SAMDB database	With the introduction of SAM 5.0, a sideband "SAMDB" MySQL database can be established and updated nearly synchronously with SAM metadata, allowing for enhanced query capability of SAM metadata. HPCMP notes: For the initial release of SAM-QFS 5.0, enabling the SAM log daemon that SLM functionality will use also requires that a "SAMDB" MySQL database be specified in the SAM configuration. Until these two features are decoupled in a future release of SAM, sites not wanting to use the SAMDB feature may specify a nonexistent MySQL instance, in which case the periodic attempts to connect to the MySQL instance will be tried but will fail.

3.8.5 ACSLS Server and SL8500

The following HPCMP-supplied table summarizes the port numbers used for ACSLS and the SL8500 at each of the DSRCs:

Table 3-12. ACSLS and SL8500 Tape Library Ports

DSRC	ACSLS Ports	SL8500 Port
AFRL	50002, 50003, 50006, 50007, 50008, 50010	50001
ARL	50041-50048	50001
ORS	50019, 50020	50001
ERDC	40173, 40203, 50010, 50020	50001
MHPCC	50014	50001
Navy Env 1 (Katrina)	50014, 50016, 50018, 50019	49640
Navy Env 3	50004	50001

3.9 Open Source Software

Depending on configuration, there are a number of open source packages used within SRB and SAM-QFS. The following table lists those packages and their roles within this solution.

Table 3-13. Open Source Software Used in this Solution

Open Source Package	Usage within Nirvana SRB
libcurl	Certain data transfer protocols at the SRB driver level and for SAM-QFS.
libical	Scheduling.
libiconv	Character set conversion.
libz	Compression and decompression.
MIT Kerberos	Optional authentication and secure data transfer protocol; for the HPCMP project MIT Kerberos will not be used; HPCMP Kerberos will be used instead.
OpenSSH	Certain SSH-based data transfer protocols (sftp/scp) at the SRB driver level.
OpenSSL	License management and encrypted data transfer for SRB and SAM-QFS.

4 ILM PHYSICAL CONFIGURATION

The SLM solution introduces two new software components to the existing HPCMP center HSM environment. These components are the Nirvana SRB ILM software and a Metadata Catalog supported by Oracle Real Application Cluster (RAC). To support Oracle RAC and SRB, a new

set of four servers with accompanying storage will be implemented at each Center. Two systems will be dedicated to the database function, while the other two are used for the SRB function. This configuration is depicted in Figure 4-1.

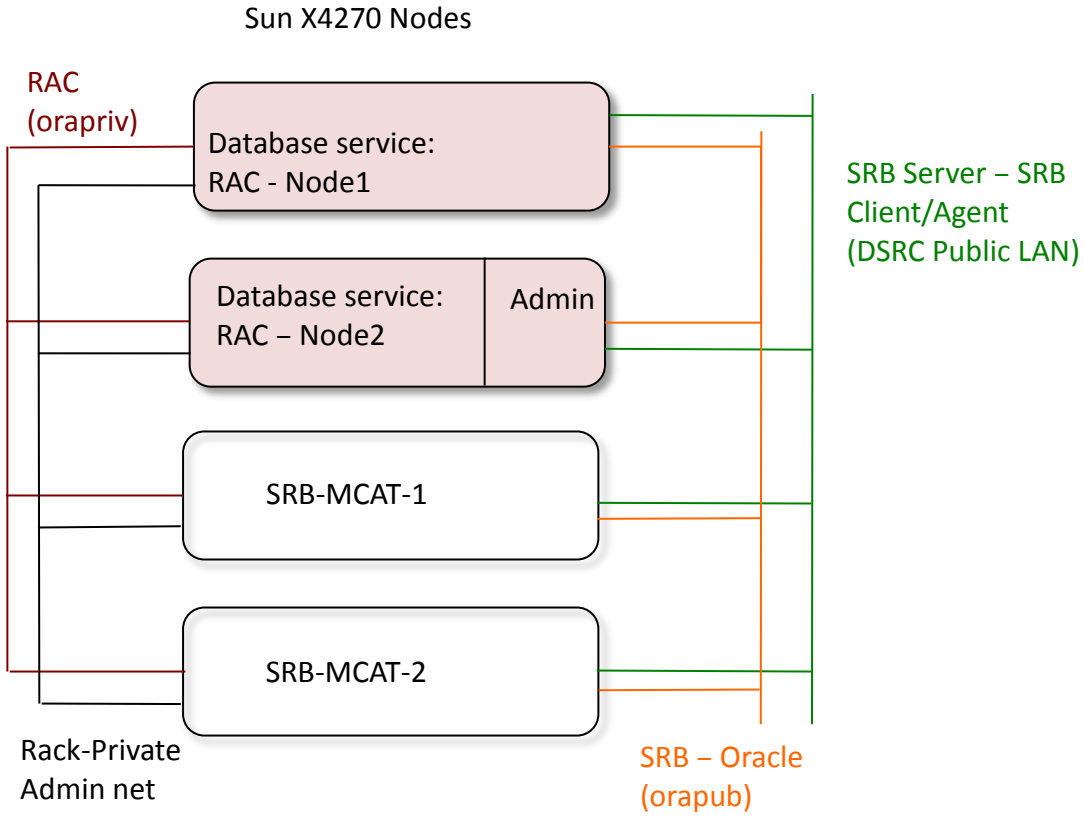


Figure 4-1. MCAT/SRB Server Configuration.

The RAC servers provide the needed I/O performance and scalability needed to support an HPC environment. Two Oracle Sun Fire X4270 servers are used in Oracle cluster configuration. As the MCAT will require ample storage to support metadata storage from all centers, an Oracle Sun Storage 6180 storage array with three CS200 expansion trays is provided. In its final configuration, the storage array includes 64 300GB hard disks to provide about 19TB of raw capacity that is direct attached to the RAC servers. A high level depiction of these systems with notional network connections is shown in Figure 4-2, below.

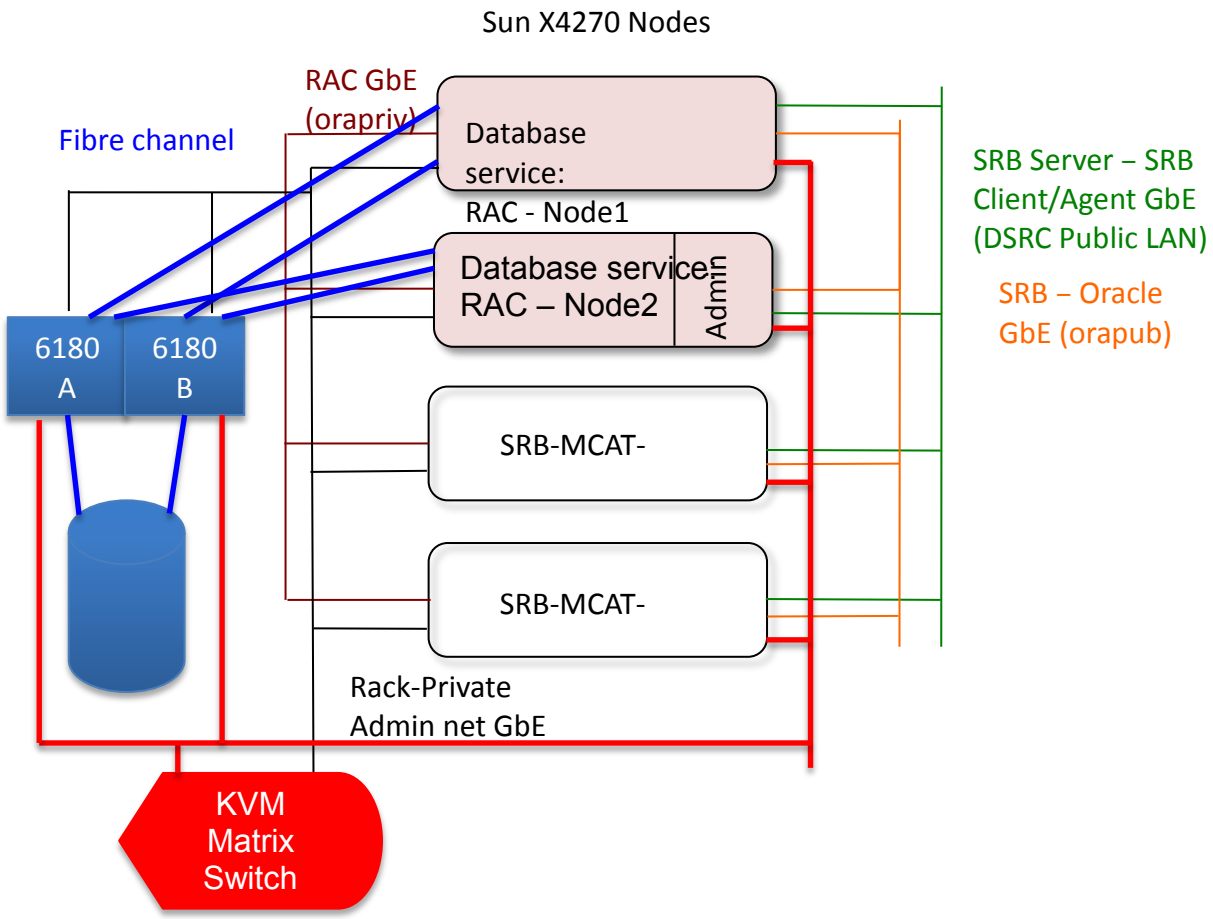


Figure 4-2. High-level interconnectivity of X4270 servers and ST 6180 Disk Array

The sections that follow will provide additional details on this architecture as follows:

- Hardware Physical Characteristics
  - MCAT Server Physical Configuration
  - Storage Array Physical Configuration
- Storage Network Connections
- Storage Configuration
- Network Physical Configuration
- Space, Power and Cooling
- Backup Power Requirements

4.1 Hardware Physical Characteristics

A complete Bill of Materials (BOM) of the architecture to be deployed at each Center is included in the appendix. This section summarizes the physical characteristics of the MCAT Servers (X4270) and the MCAT/RAC Storage Array (6180). For complete details on these systems, it is recommended that the systems reference material is consulted.

4.1.1 MCAT Server (X4270) Physical Configuration

Memory. Each X4270 server is configured with 48GB of memory initially as delivered. Upon install at each Center, the configuration should differ among sites primarily based on the number of concurrent users but also take into consideration the number of files at each DSRC. The following table illustrates the recommended memory configuration per Center.

Table 4-1. Per-Center MCAT Server Memory Configuration.

Site	Users	Concurrent Users	User %	Memory [GB]	Files
AFRL	1595	160	35%	48	39,350,629
ARL	666	67	15%	48	111,418,250
ORS	300	30	7%	48	44,348,522
ERDC	666	67	15%	48	24,463,622
MHPCC	221	22	5%	48	18,813,209
NAVO	1052	105	23%	48	93,125,193
Total	**4500**	**451**	**100%**	**288**	**331,519,425**

Storage. As purchased, the X4270 servers are initially configured with two 300GB disk drives and an internal RAID-capable SAS controller. Four additional drives were subsequently added to each system. The six drives are configured into two three-way mirrors. One of the mirrors is for the OS while the other mirror set is for applications and logging. One of the three disk drives in each mirror set will normally be offline as a DR copy, and it will be periodically brought online, synched, and configured back offline. Six additional drives will be added in the test environment systems.

Ethernet Network. The MCAT / RAC X4270 servers are configured with four onboard GbE ports. An additional quad GbE PCIe card adds network hardware redundancy. The fourth node intended for admin use has an additional quad GbE PCIe card for connectivity to DSRC backup resources.

Fibre Channel. Two dual-port FC HBA's enable storage data path redundancy. The SRB-MCAT Servers have these installed but they are not used.

Crypto card. All X4270 servers have a Crypto Accelerator 6000 installed in slot 5. The card is not used in the initial configuration but is available for future use.

Adapter placement in the X4270's is proposed as shown in Table 4-2. A short-hand “Adapter ID” is also assigned for future reference.

Table 4-2. X4270 Adapter Placement.

PCIe Slot	Adapter	Adapter ID
0	Sun StorageTek 8 port internal SAS RAID, Host Bus Adapter with RAID 0, 1, 1E, 10, 5, 5EE, 50, 6, and 60 support, 256 MB of onboard memory with 72 hour battery backed write, cache.	Internal SAS
1	Sun StorageTek PCI-E Enterprise 8Gbps Fibre Channel host,bus adapter, Dual Port, Emulex, includes standard and low profile brackets. (Not used on SRB-MCAT servers.)	HBA 1
2	Sun StorageTek PCI-E Enterprise 8Gbps Fibre Channel host, bus adapter, Dual Port, Emulex, includes standard and low profile brackets. (Not used on SRB-MCAT servers.)	HBA 2
3	Sun x4 PCI Express Quad Gigabit Ethernet,UTP low profile adapter, low profile bracket on board, standard bracket included.	PCIe GbE 1
4	Admin node only: Sun x4 PCI Express Quad Gigabit Ethernet,UTP low profile adapter, low profile bracket on board, standard bracket included.	PCIe GbE 2
5	Crypto Accelerator 6000 PCIE NIC	Crypto 1

The following diagram depicts a rear view of the X4270 MCAT server and illustrates the placement of the various adapters as identified above. The Crypto Accelerator is not shown. In the admin node, the extra Quad GbE goes in slot 4, above FC HBA 1 in the drawing

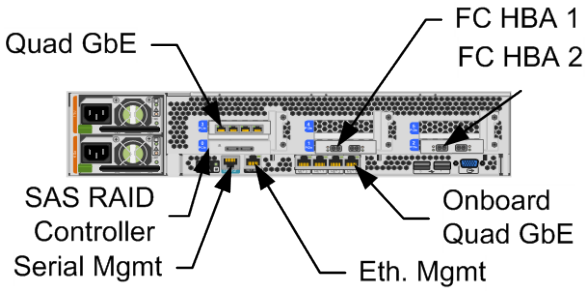


Figure 4-3. View of the X4270 Rear Panel.

4.1.2 MCAT/RAC Storage Array (6180) Physical Configuration

Each site will deploy an Oracle Sun Storage 6180 as the MCAT / RAC shared data store. The 6180 contains two Fibre Channel (FC) controllers (A and B) that provide host and expansion FC ports as well as Network Management ports.

Network Management. Each controller provides two network management ports to allow for out-of-band management of each controller.

FC Host Ports. Each controller provides four 8-Gb/s FC Host ports that will be connected to the X4270 hosts as will be shown later in this document.

FC Expansion Ports. Two 4-Gb/s FC ports are provided to support the addition of storage expansion trays. These ports will be connected to the CSM200 expansion trays

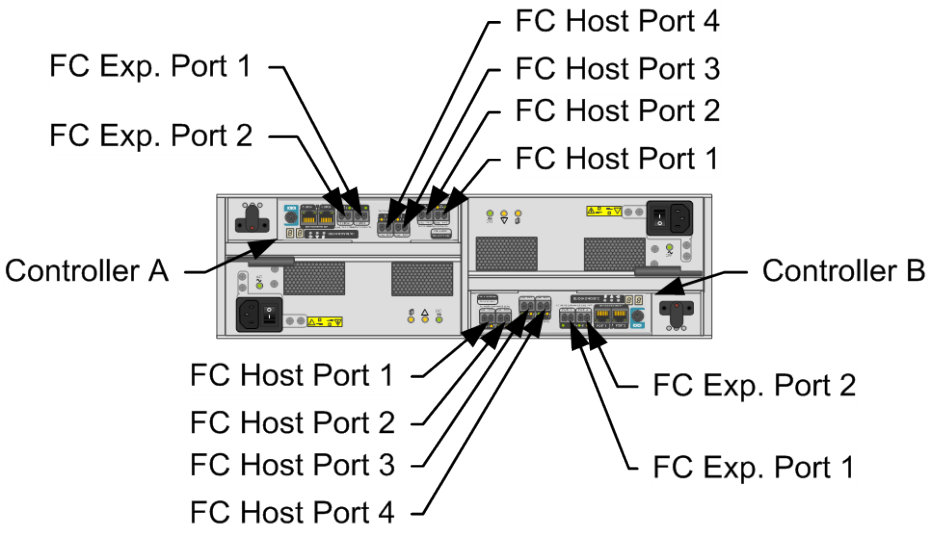


Figure 4-4. 6180 Rear Panel Showing Fibre Channel Ports.

4.1.3 CSM200 Drive Modules (Expansion Trays)

Also included in the ILM Hardware configuration are three CSM200 Drive Modules. Each Drive module also contains two 4 Gb/s FC ports per controller and will be used to connect to either another drive module, or to the 6180 storage

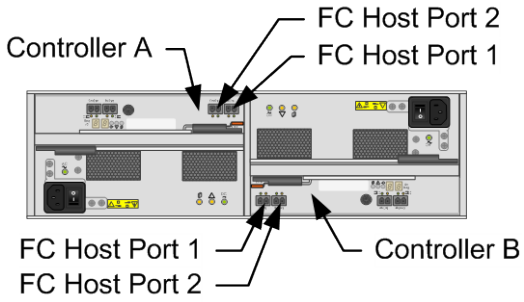


Figure 4-5. CSM200 Drive Module Rear Panel.

4.1.4 "Controller x Array" Configuration

The physical configuration of the provided solution is three expansion trays attached to the 6180 storage array controller. In this configuration, the 6180 controller is itself considered a tray (tray 0). The standard naming convention is (controller number) x (number of trays). The drive attachment for the solution configuration is to be per Oracle product documentation recommendations for a "1x4" array configuration. Complete details of this configuration may be found at http://docs.oracle.com/cd/E19373-01/821-0135-11/821-0135-11.pdf (or latest updated thereof).

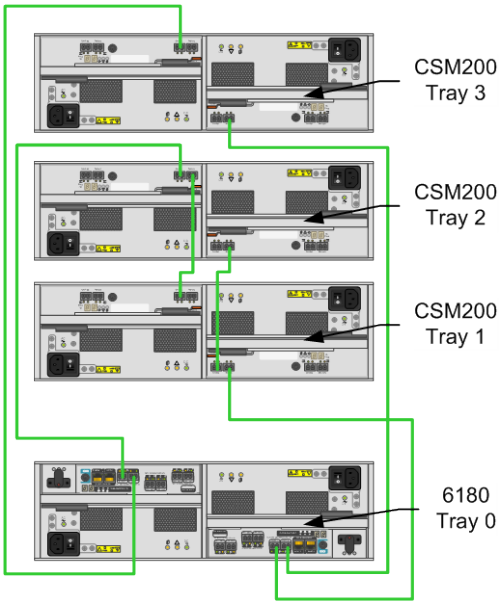


Figure 4-6. "1x4" Array Configuration.

4.2 Direct Attached Storage

It is possible to directly attach storage so that no SAN ports will be consumed within the existing infrastructure and no additional SAN FC switches will be required. As originally designed, all four servers were attached to the 6180 but the revised 2-tier configuration only requires nodes 3 and 4 be connected to the 6180. Fiber for nodes 1 and 2 should be disconnected at the HBAs but left in place in the rack for possible future use. The following table shows the connections between the initiator node (MCAT/RAC server) ports and the target ports on the MCAT 6180 Storage Array.

Table 4-3. FC Initiator Port to Target Port Mapping.

Initiator Node	SAN-Visible Target Port(s)
MCAT/RAC Server 3, HBA 1, Port 1	MCAT 6180, Controller A, Channel 1
MCAT/RAC Server 4, HBA 1, Port 1	MCAT 6180, Controller A, Channel 2
MCAT/RAC Server 3, HBA 2, Port 1	MCAT 6180, Controller B, Channel 1
MCAT/RAC Server 4, HBA 2, Port 1	MCAT 6180, Controller B, Channel 2

The following diagram depicts the configuration defined in the table above.

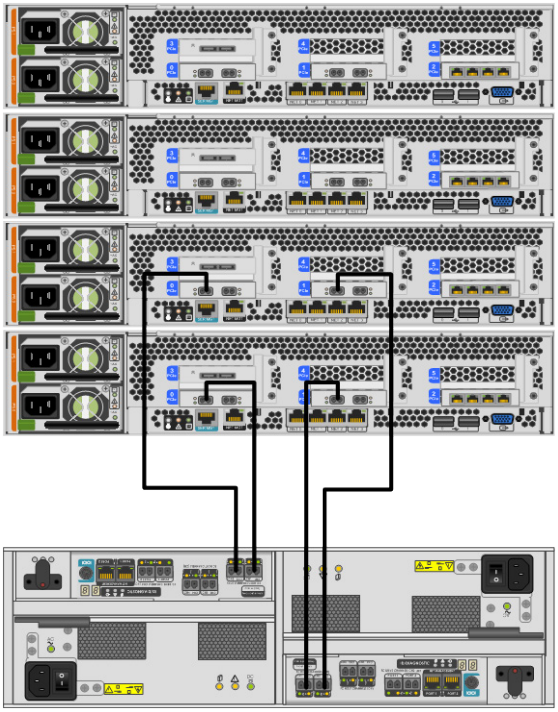


Figure 4-7. MCAT Server FC Port Connectivity for Direct Attached.

4.3 Storage Configuration

4.3.1 Array Parameters

The following parameters are set at the array level. Initial recommendations are given with the expectation that tuning may be required.

Table 4-4. Array Parameter Recommendations.

Attribute	Value
Default Host Type	Solaris (with Traffic Manager)
Cache Block Size	8KB
Default Cache Start %	default
Default Cache Stop %	default
Disk scrubbing:	enable

4.3.2 Storage Domain Configuration

Both of the MCAT RAC Nodes will normally access the same LUNs, so a single storage domain (association of hosts to LUNs) will be configured in CAM for the MCAT storage. In a failure situation, it is possible any of the four nodes could be reconfigured for MCAT RAC Node service, so the 6180 will be configured ready for all four nodes. Until needed in such a failure/recovery situation, the fibers will be unplugged on servers 1 and 2.

The following host group, host and initiator naming is suggested.

Table 4-5. Suggested Host/Initiator Naming.

Entity	Contains
Host Group “MCAT_SERVERS”	Hosts: “mcat1”, “mcat2”, “mcat3”, “mcat4”
Host “mcat1”	Initiators: “mcat1_hba1_port0”, “mcat1_hba1_port1”, “mcat1_hba2_port0”, “mcat1_hba2_port1”,
Host “mcat2”	Initiators: “mcat2_hba1_port0”, “mcat2_hba1_port1”, “mcat2_hba2_port0”, “mcat2_hba2_port1”,
Host “mcat3”	Initiators: “mcat3_hba1_port0”, “mcat3_hba1_port1”, “mcat3_hba2_port0”, “mcat3_hba2_port1”,
Host “mcat4”	Initiators: “mcat4_hba1_port0”, “mcat4_hba1_port1”, “mcat4_hba2_port0”, “mcat4_hba2_port1”,
Initiators “mcat<n>_hba1_port0 ”	WWN for HBA in server <n>, PCIe slot 1, Port 0
Initiators “mcat<n>_hba1_port1 ”	WWN for HBA in server <n>, PCIe slot 1, Port 1
Initiators “mcat<n>_hba2_port0 ”	WWN for HBA in server <n>, PCIe slot 2, Port 0
Initiators “mcat<n>_hba1_port1 ”	WWN for HBA in server <n>, PCIe slot 2, Port 1

To achieve the desired LUN mapping, all LUNs would be mapped to the host group “MCAT Servers”.

4.3.3 6180 RAID Set and Volume Configuration

The recommended RAID set layout across the storage trays is depicted in Table 4-6. A pictorial illustrating the table is also shown in Figure 4-8, below.

Table 4-6. Disk Array RAID Set Layout.

Virtual Disk	RAID Type	# of Disks	Virtual Disk Size	Segment Size	Volume Use	Disks Used	Owning Controller
VD-1	5 (3+1)	4	837 GB	16KB	Datafile	t85d05, t0d04, t1d03, t2d02	A
VD-2	5 (3+1)	4	837 GB	16KB	Datafile	t85d07, t0d06, t1d05, t2d04	B
VD-3	5 (3+1)	4	837 GB	16KB	Datafile	t85d09, t0d08, t1d07, t2d06	A
VD-4	5 (3+1)	4	837 GB	16KB	Datafile	t85d11, t0d10, t1d09, t2d08	B

Virtual Disk	RAID Type	# of Disks	Virtual Disk Size	Segment Size	Volume Use	Disks Used	Owning Controller
VD-5	5 (3+1)	4	837 GB	16 KB	Datafile	t85d13, t0d12, t1d11, t2d10	A
VD-6	5 (3+1)	4	837 GB	16 KB	Datafile	t85d15, t0d14, t1d13, t2d12	B
VD-7	5 (3+1)	4	837 GB	16 KB	Fast Recovery	t85d04, t0d03, t1d02, t2d01	A
VD-8	5 (3+1)	4	837 GB	16 KB	Fast Recovery	t85d06, t0d05, t1d04, t2d03	B
VD-9	5 (3+1)	4	837 GB	16 KB	Fast Recovery	t85d08, t0d07, t1d06, t2d05	A
VD-10	5 (3+1)	4	837 GB	16 KB	Fast Recovery	t85d10, t0d09, t1d08, t2d07	B
VD-11	5 (3+1)	4	837 GB	16 KB	Fast Recovery	t85d12, t0d11, t1d10, t2d09	A
VD-12	5 (3+1)	4	837 GB	16 KB	Fast Recovery	t85d14, t0d13, t1d12, t2d11	B
VD-13	5 (3+1)	4	837 GB	16 KB	Fast Recovery	t85d16, t0d15, t1d14, t2d13	A
VD-14	5 (3+1)	4	837 GB	16 KB	Fast Recovery	t85d03, t0d16, t1d15, t2d14	B
VD-15	1 (1+1)	2	279 GB	512 KB	Vote/OCR (FG 0)	t2d15 ,t1d16	A
VD-16	1 (1+1)	2	279 GB	512 KB	Vote/OCR (FG 1)	t0d01 ,t85d02	A
VD-17	1 (1+1)	2	279 GB	512 KB	Vote/OCR (FG 1)	t1d01, t0d02	B
Hot Spare		2				t85d01,t2d16	

Hot Spare Configuration. The above allocation leaves two drives for use as hot spares.

Tray	Disk Slots 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	VD-7 R5 3+1	VD-1 R5 3+1	VD-8 R5 3+1	VD-2 R5 3+1	VD-9 R5 3+1	VD-3 R5 3+1	VD-10 R5 3+1	VD-4 R5 3+1	VD-11 R5 3+1	VD-5 R5 3+1	VD-12 R5 3+1	VD-6 R5 3+1	VD-13 R5 3+1	VD-14 R5 3+1	VD-15 R1 1+1	HS
1	VD-17 R1 1+1	VD-7 R5 3+1	VD-1 R5 3+1	VD-8 R5 3+1	VD-2 R5 3+1	VD-9 R5 3+1	VD-3 R5 3+1	VD-10 R5 3+1	VD-4 R5 3+1	VD-11 R5 3+1	VD-5 R5 3+1	VD-12 R5 3+1	VD-6 R5 3+1	VD-13 R5 3+1	VD-14 R5 3+1	VD-15 R1 1+1
0	VD-16 R1 1+1	VD-17 R1 1+1	VD-7 R5 3+1	VD-1 R5 3+1	VD-8 R5 3+1	VD-2 R5 3+1	VD-9 R5 3+1	VD-3 R5 3+1	VD-10 R5 3+1	VD-4 R5 3+1	VD-11 R5 3+1	VD-5 R5 3+1	VD-12 R5 3+1	VD-6 R5 3+1	VD-13 R5 3+1	VD-14 R5 3+1
85	HS	VD-16 R1 1+1	VD-14 R5 3+1	VD-7 R5 3+1	VD-1 R5 3+1	VD-8 R5 3+1	VD-2 R5 3+1	VD-9 R5 3+1	VD-3 R5 3+1	VD-10 R5 3+1	VD-4 R5 3+1	VD-11 R5 3+1	VD-5 R5 3+1	VD-12 R5 3+1	VD-6 R5 3+1	VD-13 R5 3+1

Figure 4-8. Disk Array RAID Set Layout.

4.3.4 Volume Configuration (LUN)

Using the RAID sets as described in the previous section, the following table illustrates the LUN or volume configuration.

Table 4-7. Volume Configuration and LUN Mapping.

LUN #	LUN NAME	LUN Capacity (GB)	FORMAT VOLNAME
0	LUN-DAT01-0	837	DAT01-0
1	LUN-DAT02-1	837	DAT02-1
2	LUN-DAT03-2	837	DAT03-2
3	LUN-DAT04-3	837	DAT04-3
4	LUN-DAT05-4	837	DAT05-4
5	LUN-DAT06-5	837	DAT06-5
6	LUN-FLA01-6	837	FLA01-6
7	LUN-FLA02-7	837	FLA02-7
8	LUN-FLA03-8	837	FLA03-8
9	LUN-FLA04-9	837	FLA04-9
10	LUN-FLA05-10	837	FLA05-10
11	LUN-FLA06-11	837	FLA06-11
12	LUN-FLA07-12	837	FLA07-12
13	LUN-FLA08-13	837	FLA08-13
14	LUN-VOT01-14	20	VOT01-14
15	LUN-VOT02-15	20	VOT02-15
16	LUN-VOT03-16	20	VOT03-16

For the above volumes, if available, enable the read-ahead and write cache parameters. As all of the above volumes should be visible to all nodes in the RAC cluster, they may all be mapped to the default storage domain (in CAM terminology) as shown in Table 4-7.

Since the 6180 arrays are proposed to be managed out-of-band via the existing CAM servers, LUN 31 of the in-band management access volume must be deleted in the default mapping.

4.4 Network Physical Configuration

As stated above and as shown in Figure 4-2, the ILM solution will be integrated into each HPC Center's existing LAN environment. It is critical that each center have sufficient Ethernet ports to support all the connections emanating from the new hardware components. Two 10GbE connections are required to connect the SLM rack Cisco 4900M to the DSRC LAN.

The table below describes the expected traffic patterns among the networks to, from and within the SLM architecture. This table is subject to change based on hardware availability and site policy.

Table 4-8. Expected SLM Architecture Network Traffic Patterns.

Traffic	Hardware Entity 1	Hardware Entity 2	Physical Network(s)
SRB Client to MCAT Server	HPC nodes, Utility Server	MCAT Server	DSRC Internal Network
SRB Client to remote MCAT Server	HPC nodes, Utility Server	Remote MCAT Server	DSRC Internal Network and DREN
SRB Agent to MCAT Server	SAM-QFS servers	MCAT Server	DSRC Internal Network
SRB Client to SRB Agent	HPC nodes, Utility servers	SAM-QFS servers	DSRC Internal Network
SRB Client to remote SRB Agent	HPC nodes, Utility servers	Remote SAM-QFS servers	DSRC Internal Network and DREN
SRB Agent to SRB Agent	SAM-QFS servers	SAM-QFS servers	DSRC Internal Network
MCAT to SRB Agent	MCAT Server	SAM-QFS servers	DSRC Internal Network
Oracle RAC Interconnect	MCAT RAC Node	MCAT RAC Node	RAC Cluster Interconnect (orapriv) VLAN 300
MCAT to Database	MCAT Server	MCAT RAC Node	Oracle Public Network (orapub) VLAN 100
Oracle Streams Replication (Deferred until Replicated Mode implemented)	Local MCAT RAC Nodes	Remote Center MCAT RAC Nodes	Oracle Public Network (orapub)
OEM Agents	Local MCAT RAC Node	Local MCAT RAC Node	Oracle Public Network (orapub)
OEM GUI	Local MCAT RAC Node	Admin workstation	DSRC Internal Network via ssh tunnel
Server console	Existing console management workstation(s)	Server ELOM	Console Network
CAM Management	Existing CAM server	6180 Management Ports	Console Network

Based on these recommendations and the expected traffic patterns in identified in Table 4-8, the following table summarizes the physical networks and port count needed to support the architecture.

Table 4-9. Recommended Networks and Port Count.

Physical Network	Virtual LAN ID	Server Ports (dual-redundant) configured with IPMP failover		MTU	Description
Network Hardware / Connections External To SLM Unit		External Label	OS Name		
Public, Dual, cross-connected Cisco 4900M / Dual 10GE optical uplink to DSRC public	100	- NET0 - PCIe slot3 port 0	- igb0 - e1000g0	Jumbo 9000	"orapub" database service to MCAT nodes within SLM Unit & DREN WAN path to SLM Support Team
	200	- NET1 - PCIe slot3 port 1	- igb1 - e1000g1	Jumbo 9000	Meta-data Catalog (MCAT) service to SRB agents & clients in DSRC internal environment of HPC login nodes, transfer queue nodes, and archive servers
	TBD	- PCIe slot4 - port 0-3	- TBD	TBD	site-local option for admin nodes (e.g. database backup to SAM file system)
Private, Dual, cross-connected Cisco 3750X / Optional site-specific 1GE Cu private net connect.	300	- NET2 - PCIe slot3 port 2	- igb2 - e1000g2	Jumbo 9000	"orapriv" Oracle Grid interconnect between database nodes only within SLM unit
	400	- NET3 - PCIe slot3 port 3	- igb3 - e1000g3	Existing 1500	Device management within SLM unit: KVM, X4270 server service processors, RAID array controllers, PDUs

Where two interfaces have been allocated for a given network (DSRC Internal Network, the Console Network, and the Oracle Public Network), Solaris IP Multipathing or Link Aggregation will be implemented to provide resilience to interface failures. Two interfaces are allocated to the RAC Cluster Interconnect which manages its own load-balancing and redundancy for failover.

The table below details exactly which MCAT Node (X4270) ports attach to which network.

Table 4-10. MCAT Node (X4270) Network to Port Mapping.

Port or Adapter ID	Port Identifier	Network
Onboard GbE	0	Oracle Public Network
Onboard GbE	1	DSRC Internal Network
Onboard GbE	2	Oracle Private Network
Onboard GbE	3	Console Network
PCIe GbE	0	Oracle Public Network
PCIe GbE	1	DSRC Internal Network
PCIe GbE	2	Oracle Private Network
PCIe GbE	3	Console Network
ELOM	NET MGT	Console Network

As stated above, it is recommended that each site's Console Network also be connected to the storage array (6180) for management via Common Array Manager. The proposed connectivity is as detailed in the table below.

Table 4-11. Storage Array (6180) Network to Port Mapping.

Port or Adapter ID	Port Identifier	Network Connected
Controller A	NET MGT 1	Console Network
Controller A	NET MGT 2	Not attached
Controller B	NET MGT 1	Console Network
Controller B	NET MGT 2	Not attached

The figure below summarizes the port physical port connections described in this section.

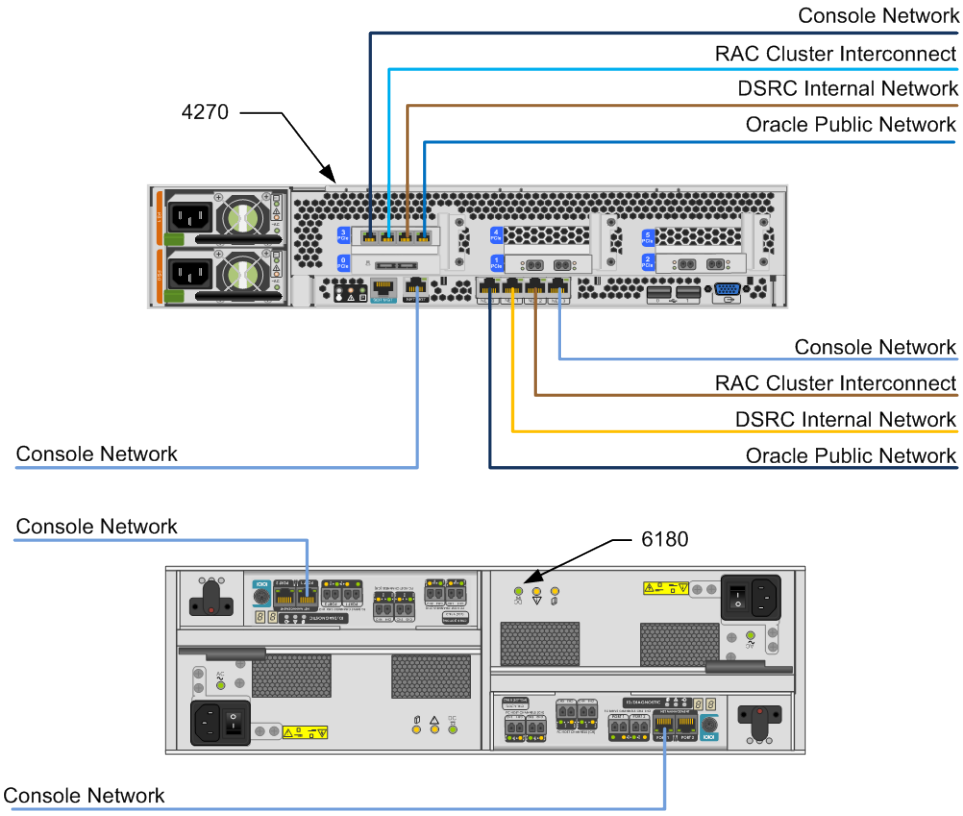


Figure 4-9. Recommended Physical Network Connections.

Figure 4-10 depicts the logical network and FC connectivity of the SLM hardware suite for the production configuration. However, it also summarizes the connectivity of all the SLM architecture components in a typical HPCMP environment.

Figure 4-10. SLM Hardware Logical Network and FC Connectivity Suite.

4.5 Space, Power and Cooling

The MCAT / RAC Servers have the following physical requirements at each site:

Table 4-12. Hardware Dimensions and Weight.

Attribute	X4270	6180 Controller Tray	6180 Expansion Tray
Quantity	4	1	3
Dimensions (each)	3.34”H (2RU) x 16.75”W x 28”D	5.1”H (3RU) x 17.6”W x 25.6”D	5.1”H (3RU) x 17.6”W x 25.6”D
Weight (each)	49 lbs	84 lbs	93 lbs

The MCAT / RAC Servers have the following power and cooling requirements as configured.

Table 4-13. Hardware Power and Cooling.

Attribute	X4270	6180 Controller Tray	6180 Expansion Tray
Quantity	4	1	3
AC Power	90-264 VAC (47-63 Hz)	100-120 / 200-240 VAC @ 50-60 Hz	100-120 / 200-240 VAC @ 50-60 Hz
Rack Power Connector	Dual power supplies Ships with qty 2 C13 / IEC-320 Sheet E	Each tray has dual power supplies Ships with qty. 2 C13 / IEC-320 Sheet E per tray	Each tray has dual power supplies. Ships with qty. 2 C13 / IEC-320 Sheet E per tray
Current @ 240V (each)	Maximum operating: 5.1A	Idle: 1.98A Maximum operating: 2.06A Maximum surge: 2.72A	Idle current: 1.98A Maximum operating: 2.06A Maximum surge: 2.67A
Power Factor	Corrected to minimum 0.98	Corrected to minimum 0.96	Corrected to minimum 0.96
Max Heat Output (each)	4212 BTU / hr	2,047 BTU/hr max. (controller tray)	1,517 BRU/hr max (expansion tray)

A notional rack configuration of the SLM systems is shown in Figure 4-11.

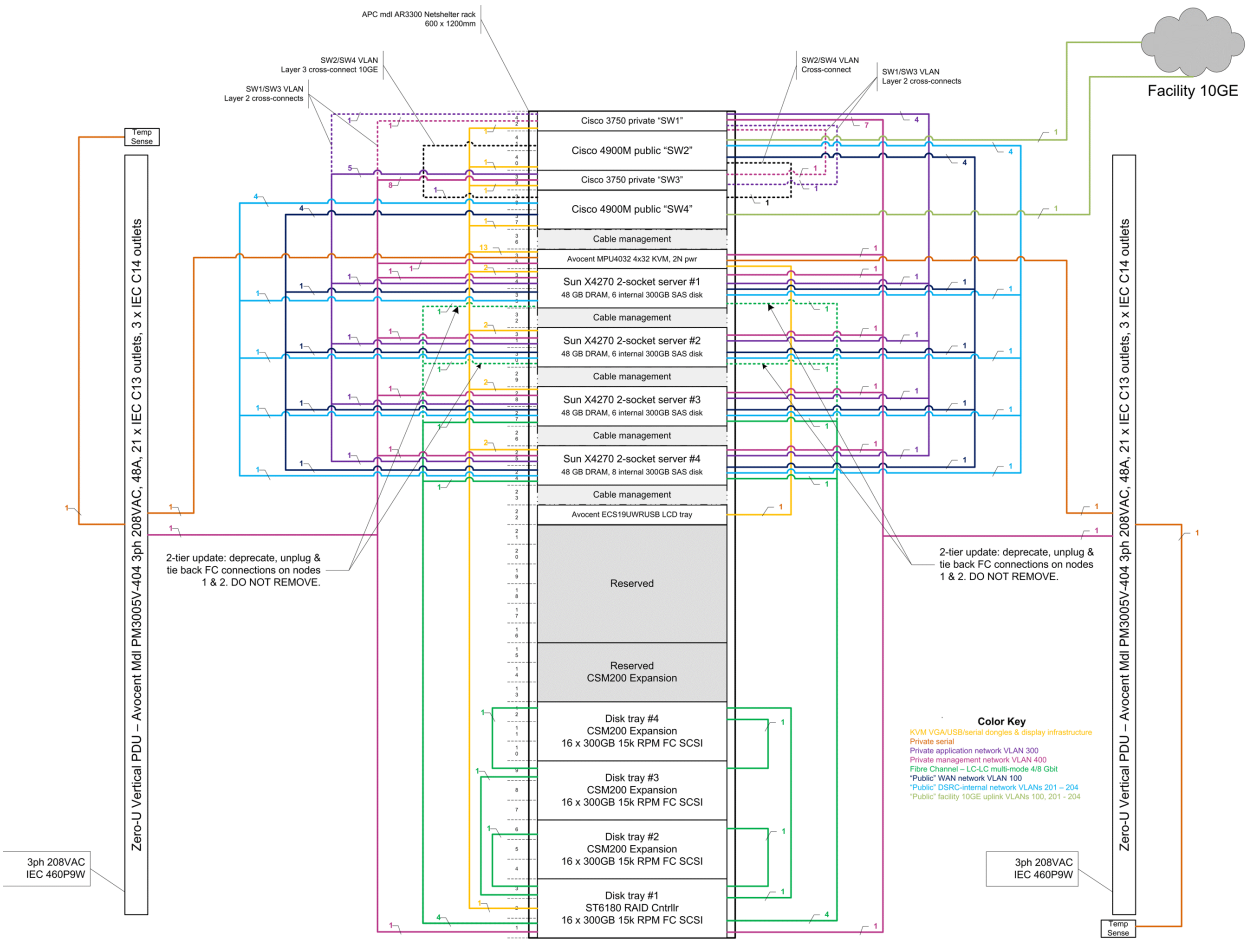


Figure 4-11. Notional Rack Physical Layout.

4.6 Backup Power Requirements

Power loss on a single X4270 is handled by failover in the corresponding MCAT Server / MCAT RAC Node cluster. In case of a power failure to all X4270 cluster nodes, Oracle recovery mechanisms would be used to bring the MCAT database back to a consistent state upon restoration of power.

The X4270's employ caching RAID HBAs for their internal OS and log drives. Write caching on these HBAs is disabled by default. Leaving write caching disabled on these HBAs avoids the risk of data loss and is the recommended setting for the MCAT environment.

The Oracle Sun StorageTek 6180 employs a battery behind its HBA with up to 4GB of data cache. Upon power failure, unflushed write data is destaged to an internal flash disk and subsequently destaged to HDDs when power is restored. This feature avoids a general requirement for UPS for the 6180.

Power failure would abort the Oracle database, but there would be no integrity issue. Uncommitted transactions would be rolled back when database startup after abort.

Transactions interrupted by a power failure may result in file stub entries in the DB without a file system reference. These stubs can easily be found and cleaned up after such a power failure.

4.7 ACLs

There are multiple points where SLM system administrators can filter out undesired entities from accessing the system components. The following table provides a brief comparison of the available methods with their advantages, disadvantages, and granularity of access control:

Table 4-14. ACL Methods

Method	Advantages	Disadvantages	Granularity	Recommended Solution
Hardware Firewall	Reduced load on servers	Reduced control with administration distributed across centers	Subnet Port	Implement if possible
Switch/Router ACLs	Reduced load on servers	Performance impact Manageability	Subnet	Not recommended
TCP Wrappers	More easily managed at the centers.	- No central management	Subnet	Not recommended
IP Filter/Server Level firewall	Load on servers, but low Good past experience	- No central management	Subnet Port	Recommended solution
SRB Server Filters	- Excellent granularity Has central management.	- May not prevent DOS attacks as server resources are used to filter.	Subnet SRB Domain SRB Group (requires query) SRB User	Not recommended

The recommendation is to implement the access control at the lowest level with the highest level of performance. This implies implementing the IP Filter mechanism provided by the Solaris OS and, where possible, a boundary hardware firewall.

5 SOFTWARE/ APPLICATIONS

The software applications introduced into the HPCMP environment, or affected by the SLM solution are listed in the table below. Each of these will be discussed in later sections. Table 5-1 shows which applications are installed on which device, the recommended installation directory and the required storage capacity necessary to host the application.

Table 5-1. SLM Solution Applications.

Application	Server	Installation Directory	Storage Requirement (estimate)
SRB	MCAT	/opt/nirvana/srb	350MB (binaries).
SRB	MCAT	/opt/nirvana/srb/log	150MB (logging, symbolic link to /var/opt/srb).
SRB	SLM, Data Movers	/opt/nirvana/srb	340MB (binaries).
SRB	SLM, Data Movers	/opt/nirvana/srb/log	150MB (logging, symbolic link to /var/opt/srb).
SRB Client	HPC Systems, Utility Servers	/opt/slm	250MB binaries.
Oracle	MCAT	/opt/oracle/	64 GB (additional details are in the cookbook)
Oracle	MCAT	/opt/oracle/stage	64 GB
SAM-QFS	SLM	/opt/SUNWsamfs	33 MB binaries
SAM-QFS	SLM	/etc/opt/SUNWsamfs	89 KB configuration
SAM-QFS	SLM	/usr/lib/fs/samfs	3.9 MB libraries
SAM-QFS	SLM	/var/opt/SUNWsamfs	140 GB Logs

All MCAT and SLM Servers must be configured to run in their local time zone – or UTC in the case of NAVY. All times inside the MCAT database will also be local time or UTC.

5.1 MCAT Server Firmware

5.1.1 ILOM Firmware Upgrade

It is good practice to stay current on ILOM firmware but testing in advance of deployment on production servers is strongly recommended. This is because some ILOM releases have had problems and it is difficult to regress versions.

Install the following revisions which are bundled with release 2.1 of the X4270 software.

Table 5-2. ILOM Firmware Recommendation.

Firmware	Installation Documentation
ILOM 3.0.6.10	http://www.oracle.com/technetwork/systems/patches/firmware/release-history-jsp-138416.html#X4170

5.1.2 Sun-branded Emulex HBA Driver and Firmware

On Solaris platforms, the Emulex HBA drivers are delivered as a Solaris patch. The driver automatically updates the firmware on Oracle-branded cards to the correct supported version.

5.1.3 MCAT Storage Array Firmware

The baseline 6180 Storage Array Controller firmware configuration for the implementation will be 07.80.51.10, which is bundled with Common Array Manager 6.9.0.16. Standard procedures for updating firmware are documented in *Sun StorageTek™ Array Administration Guide for Sun StorageTek Common Array Manager, Release 6.9.0, available at http://docs.oracle.com/cd/E24008_01/index.html*.

5.2 SLM Server

5.2.1 Adjustments to SAM-QFS Configuration

The following configuration adjustments to center SAM configurations are required to enable SRB management of SAM repositories:

1. Extend “archive age” on all archive sets to 30 years (max) to effectively hand over SRB control of archiving
2. The following related changes should be instituted as a group at the time of cutover to the SLM system.
    a. The 'sam_db' option should be added to current mount options.
    b. The SAM event log daemon ('sam-fsalogd') configuration file `'/etc/opt/SUNWsamfs/fsalogd.cmd'` should be created with the following contents:

```
event_interval = 10
log_expire = 345600 # 4 days before logs are deleted
fs = <File_system_name>
# repeat "fs = <File_system_name>" for every file system
```

    c. The SAM event log daemon is coupled to a daemon that updates a MySQL sideband database ('sam-dbupd'). To allow generation of event log files while disabling the database update daemon that is designed to consume them, configure an empty `'samdb.conf'` file as follows:

```
# cp /dev/null /etc/opt/SUNWsamfs/samdb.conf
# chmod 400 /etc/opt/SUNWsamfs/samdb.conf
```

    d. A byproduct of an empty `'samdb.conf'` file is that the SAM trace daemon for 'sam-dbupd' will record an error message for each attempt to read the `samdb.conf` file. Because 'sam-dbupd' is not actually being used, its tracing by SAM can safely be disabled. To do so, incorporate the following lines into `'/etc/opt/SUNWsamfs/defaults.conf'`.

```
trace
sam-dbupd=off
endtrace
```

5.2.2 SRB Configuration

5.2.2.1 SRB Agents

As far as SRB Agent configuration is concerned, it is mostly stored in the MCAT Database. There are a few exceptions:

1) The `$SRB_HOME/config/server.config` file contains Kerberos-specific configurations, which can all be left at their default values if `krb5.conf` and `krb5.keytab` are stored in their default directory (i.e., `/etc` and `$SRB_HOME/config` respectively) and are readable by the owner of the SRB Agent server processes. There should be an SRB Agent-specific `krb5.keytab` file created and maintained under `$SRB_HOME/config`, which would only be readable by the server process owner and would only contain an entry for the "srb" service principal on that host.
2) The `$SRB_HOME/config/.srbServerSession` file contains information such as the hostnames of the MCAT Servers, the name of the SRB Agent's Location in SRB, and the authentication scheme (i.e., `KERBEROS_AUTH`) used by the SRB Agent. This information is automatically filled-in by prompts of the "`srb init`" command, which must be executed after SRB Agent installation. Here is a sample "`srb init`" session for an SRB Agent representative of the HPCMP environment (the bold text is entered by the user):

```
> srb init
The following login information could not be found for this SRB Location.
SRB Location@Domain: seawolf@arsc.edu
SRB Server Type
         0. AGENT [default]
         1. MCAT
 Select: 0
SRB MCAT Host Name[seawolf.arsc.edu]:
mcat1.arsc.edu,mcat2.arsc.edu,mcat3.arsc.edu
SRB MCAT Host Port[5625]: 5625
SRB Data Transfer Scheme
         0. PLAIN_TEXT [default]
         1. SRB_ENCRYPT
         2. KERBEROS_INTEGRITY
         3. KERBEROS_SECURE
         4. GSI_INTEGRITY
         5. GSI_SECURE
 Select: 0
SRB Location Auth Scheme
         0. PASSWD_AUTH [default]
         1. KERBEROS_AUTH
         2. GSI_AUTH
         3. EXTERNAL_AUTH
 Select: 1
See /opt/nirvana/srb/log/srbServer.log file for more details
SRB Server initialized successfully...
```

3) The SAM-QFS driver must be present on the SRB Agent so that it may serve-up and otherwise interact with SAM-QFS file system resources for the SRB Federation. The SAM-QFS driver is `$SRB_HOME/modules/libSrbFileDriverSamfs.so`.

There are a number of SRB Agent configuration parameters, which are stored in the MCAT database and configured using SRB Acommands or the SRB Java Admin. The defaults can be listed as follows:

```
> SgetL -l -config -width 12 seawolf@arsc.edu
 user_name user_domain net_address parameter_order parameter_type_name parameter_name
parameter_value
-------------------------------------------------------------------------------------
 seawolf arsc.edu  seawolf.arsc.edu:5625 0  INTEGER      DEBUG_LEVEL           0
 seawolf arsc.edu  seawolf.arsc.edu:5625 1  STRING       LOG_FILE              srbServer.log
 seawolf arsc.edu  seawolf.arsc.edu:5625 2  SRB_LONG     LOG_FILE_SIZE         10000000
 seawolf arsc.edu  seawolf.arsc.edu:5625 3  STRING       LOG_FILE_GROUP        srb
```

```
 seawolf arsc.edu  seawolf.arsc.edu:5625 4  STRING       LOG_FILE_MODE          0640
 seawolf arsc.edu  seawolf.arsc.edu:5625 5  STRING_LIST  SERVER_MANAGEMENT      PROCESS
 seawolf arsc.edu  seawolf.arsc.edu:5625 6  INTEGER      SERVER_PRE_SPAWN_COUNT_MIN 20
 seawolf arsc.edu  seawolf.arsc.edu:5625 7  INTEGER      SERVER_PRE_SPAWN_COUNT_MAX 40
 seawolf arsc.edu  seawolf.arsc.edu:5625 8  INTEGER      SERVER_PRE_SPAWN_TIMEOUT 120
 seawolf arsc.edu  seawolf.arsc.edu:5625 9  STRING_LIST  SERVER_RECYCLING       YES
 seawolf arsc.edu  seawolf.arsc.edu:5625 10 INTEGER      SERVER_COMM_SOCKETS    3
 seawolf arsc.edu  seawolf.arsc.edu:5625 11 INTEGER      CLIENT_CONN_TIMEOUT    300
 seawolf arsc.edu  seawolf.arsc.edu:5625 12 STRING_LIST  HOST_BASED_AUTH_ENABLED_FLAG YES
 seawolf arsc.edu  seawolf.arsc.edu:5625 13 STRING_LIST  HOST_BASED_AUTH_ACCESS_MODE ALLOW_ALL
```

These parameters can usually be left at their default values. The `SERVER_PRE_SPAWN_COUNT_MIN` and `SERVER_PRE_SPAWN_COUNT_MAX` can be increased on SAM Servers where a larger number of concurrent users are expected, for example:

```
> modifyLocation seawolf@arsc.edu changeConfigValue SERVER_PRE_SPAWN_COUNT_MIN::30
```

The Physical Resource (i.e., file system driver) configuration for each Agent is also controlled through Acommands. For example, the values for a SAM-QFS Resource may look as follows:

```
> SgetR -l -config -width 12 arsc.u2
resource_name resource_type parameter_family_id parameter_order parameter_type_name
parameter_name parameter_value
-------------------------------------------------------------------------------------
 arsc.u2   samfs file system   0   0   INTEGER      DEBUG_LEVEL           0
 arsc.u2   samfs file system   0   1   STRING       LOG_FILE              samfs.log
 arsc.u2   samfs file system   0   2   SRB_LONG     LOG_FILE_SIZE         10000000
 arsc.u2   samfs file system   0   3   STRING       LOG_FILE_GROUP        srb
 arsc.u2   samfs file system   0   4   STRING       LOG_FILE_MODE         0640
 arsc.u2   samfs file system   0   5   STRING_LIST  RESOURCE_CAPABILITY   READ_WRITE
 arsc.u2   samfs file system   0   6   STRING_LIST  PRESERVE_UNIX_INFO    YES
```

The default values can be left unchanged, except to give every SAM-QFS Resource its own log file, for example:

```
> modifyResource arsc.u2 changeConfigValue 0::LOG_FILE::arsc.u2.log
```

As changes are made via Acommands or the Java Admin, they are automatically sent to all the SRB Agents affected by the changes. In order to avoid this behavior, multiple Acommands can be combined into a single bulk file using the `-file` argument. The bulk file can then be executed using `executeBulkCommands`.

5.2.2.2 SRB Daemons

Two SRB Daemons are initially deployed in the HPCMP SRB Federation – the Sync Daemon and the ILM Daemon. Both daemons run on the SLM Servers that mount the file systems that they manage. SRB Daemons are controlled by the SRB Agent server processes so that they are automatically enabled, disabled, or re-configured when tasked to do so by an SRB Administrator using Acommands or the Java Admin.

5.2.2.2.1 SRB Sync Daemons

The Sync Daemon contains many configuration parameters, which are described in the SRB Administration Guide. The important ones to consider are as follows – the output has been modified to fit the formatting of this document:

```
> SgetConf -config -width 12|grep "\<SYNC "
SYNC 0          STRING_LIST   MODE                          LOCAL
...
SYNC 2          STRING        PATH_NAME
SYNC 3          STRING        COLLECTION_NAME
SYNC 4          STRING        REGEX_MATCH                   ^[^.[:cntrl:]][[:print:]]*$
...
SYNC 6          STRING        PHY_RSRC_NAME
...
SYNC 12         STRING_LIST   ADD_FILE_METADATA_TO_OBJECTS  YES
...
SYNC 18         INTEGER       THREADS_MAX_LIMIT             50
...
SYNC 21         STRING        RECURRENCE                    FREQ=SECONDLY;INTERVAL=100
```

The following script will deploy a Sync Daemon to the `seawolf@arsc.edu` Location, configure it to synchronize the `arsc.u2` SAM-QFS Resource in real-time mode, and then run it as the super@root user:

```
daemonName="SRB executables:Scommands:Ssync"
daemonUser=super@root
daemonDeployLocation=seawolf@arsc.edu
SRB_PASSWORD=hpcmp123

echo deploying Sync Daemon to $daemonDeployLocation...
modifyLocation $daemonDeployLocation attachExecutable "$daemonName::$daemonUser"

echo global settings...
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
   "$daemonName::AUTH_INFO::$SRB_PASSWORD"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
   "$daemonName::TIME_ZONE::America/Anchorage"

daemonPolicy=ARSC.U2
echo $daemonPolicy
daemonMode=REAL-TIME
daemonPath=/arsc/archive/u2
daemonCollection=/arsc/archive
daemonResource=arsc.u2
daemonMeta=YES
daemonThreads=50
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser attachArrayRow
   "$daemonName:SYNC::$daemonPolicy"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
   "$daemonName:SYNC::$daemonPolicy::MODE::$daemonMode"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
   "$daemonName:SYNC::$daemonPolicy::PATH_NAME::$daemonPath"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
   "$daemonName:SYNC::$daemonPolicy::COLLECTION_NAME::$daemonCollection"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
   "$daemonName:SYNC::$daemonPolicy::PHY_RSRC_NAME::$daemonResource"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
   "$daemonName:SYNC::$daemonPolicy::ADD_FILE_METADATA_TO_OBJECTS::$daemonMeta"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
   "$daemonName:SYNC::$daemonPolicy::THREADS_MAX_LIMIT::$daemonThreads"

modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser enable
```

Sync Daemons can be deployed to different Locations by modifying the first three code blocks (each block is delimited by an empty line). Additional file systems can easily be added by repeating the code block starting with `daemonPolicy`.

5.2.2.2.2 SRB ILM Daemons

The ILM Daemon contains many configuration parameters, which are described in the SRB Administration Guide. The important ones to consider are as follows – the output has been modified to fit the formatting of this document:

```
> SgetConf -config -width 12|grep "\<Silm:POLICY "
Silm:POLICY 0            STRING        POLICY
Silm:POLICY 1            STRING_LIST   MODE              List Mode
Silm:POLICY 2            STRING_LIST   ACTION            Backup
Silm:POLICY 3            INTEGER       BUFFER_SIZE       2000000
Silm:POLICY 4            STRING        RESOURCE
Silm:POLICY 5            STRING        SOURCE_RESOURCE
...
Silm:POLICY 8            STRING        COMMAND
Silm:POLICY 9            STRING_LIST   FORCE_FLAG        NO
Silm:POLICY 10           STRING_LIST   RECURSIVE_FLAG    NO
Silm:POLICY 11           STRING_LIST   USE_WATERMARK     NO
...
Silm:POLICY 13           STRING        RECURRENCE        FREQ=SECONDLY;INTERVAL=100
```

The following script will deploy an ILM Daemon to the `seawolf@arsc.edu` Location, configure it to perform a policy (expiration on the `arsc.u2` SAM-QFS Resource with retention period), and then run it as the `super@root` user:

```
daemonName="SRB executables:Scommands:Silm"
daemonUser=super@root
daemonDeployLocation=seawolf@arsc.edu
SRB_PASSWORD=hpcmp123

echo deploying ILM Daemon to $daemonDeployLocation...
modifyLocation $daemonDeployLocation attachExecutable "$daemonName::$daemonUser"

echo global settings...
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
   "$daemonName::AUTH_INFO::$SRB_PASSWORD"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
   "$daemonName::TIME_ZONE::America/Anchorage"

daemonPolicy=EXPIRE_RETENTION
echo $daemonPolicy
daemonPolicyQuery=((EXPRESSION.create_age > Admin.Retention_Period) OR
   (EXPRESSION.current_timestamp > Admin.Next_Review_Time))
   AND (Admin.Admin_Hold = 'no') AND (DATA_OBJECT.data_type NOT LIKE '*collection')
daemonMode=Normal Mode
daemonAction=Delete
daemonSourceResource=arsc.u2
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser attachArrayRow
   "$daemonName:POLICY::$daemonPolicy"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
   "$daemonName:POLICY::$daemonPolicy::POLICY::$daemonPolicyQuery"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
   "$daemonName:POLICY::$daemonPolicy::MODE::$daemonMode"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
```

```
    "$daemonName:POLICY::$daemonPolicy::ACTION::$daemonAction"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
    "$daemonName:POLICY::$daemonPolicy::SOURCE_RESOURCE::$daemonSourceResource"

modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser enableExecutable
```

ILM Daemons can be deployed to different Locations by modifying the first three code blocks. Additional policies can easily be added by repeating the code block starting with `daemonPolicy`.

5.3 MCAT Server Operating System and Applications

5.3.1 Service Processor (SP) Configuration

It is anticipated that each Center has experience and practices around managing Sun LOM ports. Each Center will need to configure the service processor's IP addresses, user accounts and access methods (web or CLI or both) to site standards.

Note that without exposing the web management GUI access from the SP, a TFTP server is required for any ILOM firmware upgrade. Note also that web management GUI access requires Java 5 Update 7 on the browser client.

System console access is strongly recommended and is available via the SP command line interface (CLI). Optionally, sites may choose to enable the remote HTTPS-based graphical console as allowed by site standards.

5.3.2 Operating System Installation and Configuration

The baseline version of Solaris for MCAT and RAC deployment is Solaris 10 Update 10 (08/11). Installation is described in the sections below.

5.3.2.1 Installation Parameters

Installation of Solaris will be accomplished via jump start as defined by the SLM Support Team. The following installation parameters require special consideration.

Table 5-3. Solaris Installation Parameters for Special Consideration.

Parameter	Value
Primary Ethernet interface	Onboard GbE port 0 (igb0)
Ethernet interface configuration	Configure first instance of non-cluster networks according to the table in the network section 4.4 above Creation of IP Multipathing groups are recommended but wait until post installation to do so.
NTP server	Time synchronization among nodes is required. Point to Cisco 4900 NTP services in SLM rack.

File system layout is part of the base Solaris install, but described separately in the next section.

5.3.3 4270 Internal RAID Set and Volume Configuration

Initially, the X4270 MCAT systems will be configured with an allocation of two 300GB disk drives and an internal RAID-capable SAS controller. These will serve as storage for the primary booting OS instance and application environment. In its final configuration four (production) or six (test environment) additional drives will be installed for alternate boot environments and logging targets.

The internal RAID controller in the X4270 configurations enables the pair-wise mirroring of disks for redundancy. The figure below illustrates the disk layout of the X4270.

Figure Legend

HDD3	HDD7		HDD12	HDD15
HDD2	HDD6		HDD11	HDD14
HDD1	HDD5	HDD9	DVD and USB Connectors (2)	
HDD0	HDD4	HDD8	HDD10	HDD13

Figure 5-1. X4270 Front Panel Disk Layout.

The following file systems should be created during Solaris installation.

Table 5-4. X4270 Recommended File Systems.

Disk or File System	File System Type	File System Minimum Size	Notes
HDD0 (279GB usable capacity)			Boot zpool 'rpool' (274 GB) ZFS-mirrored to HDD1, HDD2
/	ZFS	274 GB (shared via 'rpool')	OS
/var	ZFS	274 GB (shared via 'rpool')	System logs and scratch area
/export	ZFS	274 GB (shared via 'rpool')	$HOME, mount point for external backup file system
/nosnap	ZFS	274 GB (shared via 'rpool')	Target for crash dumps which will not be included in rpool snapshots to avoid excessive disk consumption.
HDD1 (279 GB usable capacity)			ZFS-mirror of HDD0
HDD2 (279 GB usable capacity)			ZFS-mirror of HDD0 which will be offlined to keep as DR backup. It will be periodically onlined, synched, and offlined again.

Disk or File System	File System Type	File System Minimum Size	Notes
HDD3 (279 GB usable capacity)			zpool 'apool' for applications; ZFS-mirrored to HDD4, HDD5
/opt/nirvana	ZFS	274 GB (shared via 'apool')	Nirvana software and logs (MCAT systems)
/opt/oracle	ZFS	274 GB (shared via 'apool')	Oracle software (all servers)
HDD4 (279 GB usable capacity)			ZFS-mirror of HDD3
HDD5 (279 GB usable capacity)			ZFS-mirror of HDD4 which will be offlined to keep as DR backup. It will be periodically onlined, synched, and offlined again

5.3.3.1 Solaris Patching

The initial MCAT implementation assumes installation of the latest Solaris 10 recommended patch cluster. Beyond the recommended patch bundle, Oracle RAC configuration requires the patches specified in the 11gR2 RAC installation guide.

5.3.3.2 Application Pre-requisites

Because the Oracle Call Interface is utilized by SRB, Oracle RAC 11g R2 requires installation of a compiler on all MCAT Server nodes. Team GA recommends installing the Sun ONE Studio 12 compiler and all associated patches from SunSolve. Optionally, one could also use GCC v3.4.2. The compiler does not need to remain installed in production. Please refer to the Database Installation Guide for Oracle Solaris 11g Release 2 (11.2) at http://www.oracle.com/pls/db112/to_toc?pathname=install.112/e24346/toc.htm.

5.3.3.3 System Tuning Parameters

The MCAT installation requires the following OS-level parameter changes be made.

Table 5-5. System Tuning Parameters.

Modification	Reason
Add to /etc/system line: set no_exec_user_stack=1	Required by Oracle installation
Create an 'oracle' project with the following settings: project.max-sem-ids=100 process.max-sem-nsems=256 project.max-shm-memory=4294967295 project.max-shm-ids=100	Solaris 10 shared memory resource management is implemented with Solaris projects

5.3.3.4 Other Post-Installation Configuration

The following sub-sections highlight configuration steps to be taken immediately following the installation of Solaris and prior to application installation.

5.3.3.4.1 Configure the cluster networks for availability Solaris using IP Multipathing (IPMP).

The following table describes the recommended IPMP configuration for the SLM solution. More detail on the types of IPMP implemented can be found in the Oracle White Paper titled “Highly Available and Scalable Oracle RAC Networking with Oracle Solaris 10 IPMP” published in September 2010 and available under the following URL:

http://www.oracle.com/technetwork/articles/systems-hardware-architecture/ha-rac-networking-ipmp-168440.pdf

Table 5-6. IP Multipath Configuration.

Group Name	Interface 1	Interface 2	Multipath Type	Notes
orapub	On-board port 0	PCIe GbE port 0	Probe-based failure detection with Active-Standby IPMP group (each interface on port 0 is plumbed with a physical IP address; a virtual IP within that group fails over from one interface to the other)	Connects to Center LAN
srb	On-board port 1	PCIe GbE port 1	Link-based failure detection with Active-Standby IPMP group (one IP address in a Solaris group fails over from one interface to another; physical interface gets plumbed; there is no virtual interface)	
orapriv	On-board port 2	PCIe GbE port 2	Not configured for IPMP (Oracle Clusterware expects 2 plumbed devices)	Connects to Center LAN
rackpriv (if multipathing desired)	On-board port 3	PCIe GbE port 3	Link-based failure detection with Active-Standby IPMP group (one IP number in a Solaris group fails over from one interface to another; physical interface gets plumbed; there is no virtual interface)	Connects to console network or Center LAN per site policy

5.3.3.4.2 Create application groups

Table 5-7. UNIX Application Groups to be created.

Group name	Group Description
srb	SRB group
oinstall	Oracle inventory group
dba	OSDBA group

5.3.3.4.3 Create application users

Table 5-8. UNIX Application Users to be created.

User name	Home directory	Primary group	Other groups	Other
srb	/opt/nirvana/home	srb		
oracle	/opt/oracle/home	oinstall	dba	

5.3.4 Oracle Software Suite

Oracle RAC 11gR2 will be used as the underlying database for the MCAT. The Oracle Grid Infrastructure 11g Release 2 (11.2), Automatic Storage Management (ASM) and Oracle Clusterware software are packaged together in a single binary distribution and installed into a single home directory, which is referred to as the Grid Infrastructure home.

In order to use Oracle RAC 11g Release 2 the Oracle Grid Infrastructure must be installed. While the installation of the combined products is called Oracle Grid Infrastructure, Oracle Clusterware and Automatic Storage Manager remain separate products.

Detailed installation and configuration instructions on Oracle Grid Infrastructure and Oracle Database 11gR2 are provided as part of the SLM Cookbook.

5.3.4.1 Oracle Database Design

The database 'mcat' will have two instances: 'mcat<site>1' and 'mcat<site>2'. The database will be configured with the following settings:

- General Purpose Database
- Database block size will be 8k
- UNDO tablespace for each instance.
- Temp tablespace for each instance.
- Multiplexed control files
- Multiplexed redo log files
- Multiplexed archive log files.
- Automatic memory management
- Automatic undo management
- Automatic multi block read
- Use of flashback recovery area for backup
- Full database backup on Saturday night
- Daily incremental backup
- Shared server

Shared server processes provide a better performance and locking behavior for running SRB with Oracle RAC in an environment with multiple concurrent users.

Oracle Enterprise Manager (OEM) is an effective tool to monitor and administer the Oracle database infrastructure and RAC database. It is also integrated with RMAN and its GUI makes it is easy to setup backup and retention periods.

It is recommended that OEM and an RMAN repository database be setup outside the cluster in order to monitor the MCAT RAC Nodes, the database infrastructure, and the RAC database.

The following table summarizes the available database instances at each site:

Table 5-9. Site Database and Instance Names

DSRC	Site Id	Database Name	Instance Names
AFRL	1	mcat	mcat1, mcat2
ARL	2	mcat	mcat1, mcat2
ORS	3	mcat	mcat1, mcat2
ERDC	4	mcat	mcat1, mcat2
MHPCC	5	mcat	mcat1, mcat2
NAVY	6	mcat	mcat1, mcat2
HTL	7	mcat	mcat1, mcat2

Most of the SRB application metadata is stored in the 8 largest tables. The following table provides size estimates of system and application tablespaces and log files. The calculations are based on center file system information as of October 2010. Once the Global Namespace is implemented, the size of the database at each site is expected to be almost equal as data is replicated across other sites by Streams Replication.

Table 5-10. Database Sizing Estimates

#	Tablespace/Log File	Size (GB)	Description
1	SYSTEM	2.00	System tablespace
2	SYSAUX	2.00	Sysaux tablespace
3	USERS	0.20	Users tablespace.
4	UNDO	150.00	Undo TB sizing is determined by UNDO_RETENTION_PERIOD, DB block size and transactions per second. It requires stress testing to identify the right value.
5	TEMP	150.00	Temp datafile sized at 50 GB. An approximate value is derived from stress test.
6	COLLINFOTBS	7.00	MCAT_COLL_INFO table
7	DATAACCSTBS	4.00	MCAT_DATA_ACCS table
8	DATAAUDITTBS	289.00	MCAT_DATA_AUDIT table
9	DATAINFOTBS	433.00	MCAT_DATA_INFO table
10	DATAREPLICATBS	227.00	MCAT_DATA_REPLICA table
11	ADMINTBS	79.00	SRB_ADMIN table
12	HSMTBS	81.00	SRB_HSM table
13	HSMCOPYTBS	278.00	SRB_HSM_COPY table
14	MCATOTHERSTBS	65.00	Other tables in MCAT database
15	COLLINFOIDX	6.00	MCAT_COLL_INFO indexes
16	DATAACCSIDX	3.00	MCAT_DATA_ACCS indexes
17	DATAAUDITIDX	50.00	MCAT_DATA_AUDIT indexes
18	DATAINFOIDX	287.00	MCAT_DATA_INFO indexes
19	DATAREPLICAIDX	157.00	MCAT_DATA_REPLICA indexes
20	ADMINIDX	0.00	SRB_ADMIN indexes. Future indexes unknown
21	HSMIDX	0.00	SRB_HSM indexes. Future indexes unknown
22	HSMCOPYIDX	0.00	SRB_HSM_COPY indexes. Future indexes unknown
23	MCATOTHERSIDX	35.00	MCAT database non large table indexes.
24	Redo log files	36.00	Six redo log files for each instance, size 3 GB, multiplexed.
25	Archive Redo log files	300.00	Duplexed archive log files, archive size 150 GB at each destination with 3GB archive log file sizes.
	Total	**2641.20**	

To ensure there remains ample capacity, the audit records should be kept for only 1 year. Thereafter these should be deleted or archived into SAM-QFS. The following diagram shows the relationships of the tables that are central to the MCAT.

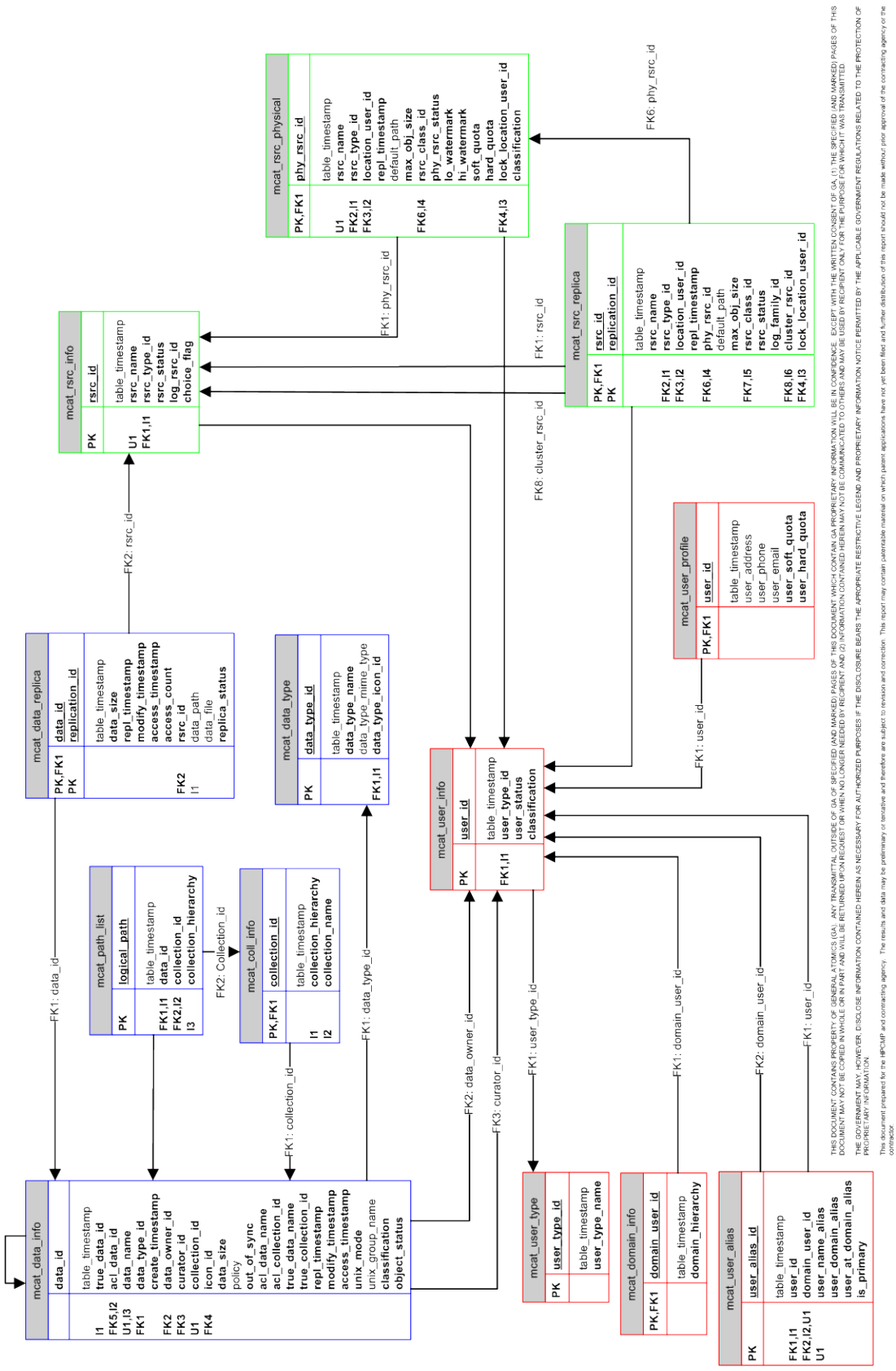


Figure 5-2. Entity Relationship Diagram for MCAT v11

5.3.5 SRB MCAT Server Configuration

The MCAT Server configuration is mostly stored in the MCAT Database. There are a few exceptions:

1. The `$SRB_HOME/config/server.config` file contains Kerberos-specific configurations, which can all be left at their default values if `krb5.conf` and `krb5.keytab` are stored in their default directory (i.e., `/etc` and `$SRB_HOME/config` respectively) and are readable by the owner of the SRB MCAT Server processes. There should be an SRB MCAT Server-specific `krb5.keytab` file created and maintained under `$SRB_HOME/config`, which would only be readable by the server process owner and would only contain an entry for the "srb" service principal on that host.
2. The Oracle driver must be present on every MCAT Server so that it may interact with the MCAT Database – implemented as Oracle RAC. The Oracle driver is `$SRB_HOME/modules/libSrbDBDriverOracle.so`.
3. The `$SRB_HOME/config/mcat.config` file must be configured so that the MCAT Server can authenticate and connect to the Oracle database. This file is discussed in section 3.3.5 MCAT Server to Oracle Authentication.
4. The `$SRB_HOME/config/.srbServerSession` file contains information such as the server type (i.e., `MCAT`), the name of the MCAT Server's Location in SRB, and the authentication scheme (i.e., `KERBEROS_AUTH`) used by the MCAT Server. This information is automatically filled-in by prompts of the "`srb init`" command, which must be executed after MCAT Server installation. Here is a sample "`srb init`" session for an MCAT Server representative of the HPCMP environment:

```
> srb init
The following login information could not be found for this SRB Location.
SRB Location@Domain[mcat1]: mcat1@arsc.edu
SRB Server Type
         0. MCAT [default]
         1. AGENT
 Select: 0
SRB Data Transfer Scheme
         0. PLAIN_TEXT [default]
         1. SRB_ENCRYPT
         2. KERBEROS_INTEGRITY
         3. KERBEROS_SECURE
         4. GSI_INTEGRITY
         5. GSI_SECURE
 Select: 0
SRB Location Auth Scheme
         0. PASSWD_AUTH [default]
         1. KERBEROS_AUTH
         2. GSI_AUTH
         3. EXTERNAL_AUTH
 Select: 1
See /opt/nirvana/srb/log/srbServer.log file for more details
SRB Server initialized successfully ...
```

There are a number of MCAT Server configuration parameters, which are stored in the MCAT database and configured using SRB Acommands or the SRB Java Admin. These are settings, which apply to the whole SRB Federation and which are discussed in the SRB Administration

Guide in more detail. The recommended parameters for the HPCMP SRB Federation are listed below:

```
> SgetConf -config -width 12|grep "MCAT Server"
 0000.0000   MCAT Server  0   STRING_LIST   MCAT_AUDITING_ENABLED             YES
 0000.0000   MCAT Server  1   STRING_LIST   MCAT_TICKETS_ENABLED              NO
 0000.0000   MCAT Server  2   STRING_LIST   MCAT_ACCESS_UPDATING_ENABLED      YES
 0000.0000   MCAT Server  3   INTEGER       MCAT_STD_BUFFER_SIZE              10000000
 0000.0000   MCAT Server  4   STRING_LIST   MCAT_EXTRA_SECURITY_ENABLED       NO
 0000.0000   MCAT Server  5   STRING_LIST   MCAT_CHECK_FOR_NO_ACCESS_ENABLED  NO
 0000.0000   MCAT Server  6   STRING_LIST   MCAT_ACL_BEHAVIOR            ACL_INHERITED
 0000.0000   MCAT Server  7   INTEGER       MCAT_MAX_COLUMN_INPUT_SIZE        10000000
 0000.0000   MCAT Server  8   INTEGER       MCAT_MAX_COLUMN_OUTPUT_SIZE       10000000
 0000.0000   MCAT Server  9   STRING_LIST   MCAT_SYSTEM_SCHEMES_ENABLED       YES
 0000.0000   MCAT Server  10  STRING_LIST   MCAT_CLASSIFICATIONS_ENABLED      YES
 0000.0000   MCAT Server  11  STRING_LIST   MCAT_PRESERVE_UNIX_INFO_ENABLED   YES
 0000.0000   MCAT Server  12  STRING_LIST   MCAT_LOCATION_RANGES_ENABLED      YES
 0000.0000   MCAT Server  13  STRING_LIST   MCAT_AUTO_ADD_USERS_ENABLED       NO
```

The MCAT Server has some additional parameters (identical to SRB Agent parameters), which can be listed as follows:

```
> SgetL -l -config -width 12 mcat71@arsc.edu
 user_name    user_domain  net_address  parameter_order parameter_type_name
parameter_name parameter_value
-------------------------------------------------------------------------------------
-----
 mcat1 arsc.edu  mcat1.arsc.edu:5625 0  INTEGER     DEBUG_LEVEL          0
 mcat1 arsc.edu  mcat1.arsc.edu:5625 1  STRING      LOG_FILE             srbServer.log
 mcat1 arsc.edu  mcat1.arsc.edu:5625 2  SRB_LONG    LOG_FILE_SIZE        10000000
 mcat1 arsc.edu  mcat1.arsc.edu:5625 3  STRING      LOG_FILE_GROUP       srb
 mcat1 arsc.edu  mcat1.arsc.edu:5625 4  STRING      LOG_FILE_MODE        0644
 mcat1 arsc.edu  mcat1.arsc.edu:5625 5  STRING_LIST SERVER_MANAGEMENT    PROCESS
 mcat1 arsc.edu  mcat1.arsc.edu:5625 6  INTEGER     SERVER_PRE_SPAWN_COUNT_MIN 40
 mcat1 arsc.edu  mcat1.arsc.edu:5625 7  INTEGER     SERVER_PRE_SPAWN_COUNT_MAX 80
 mcat1 arsc.edu  mcat1.arsc.edu:5625 8  INTEGER     SERVER_PRE_SPAWN_COUNT_TIMEOUT 120
 mcat1 arsc.edu  mcat1.arsc.edu:5625 9  STRING_LIST SERVER_RECYCLING    YES
 mcat1 arsc.edu  mcat1.arsc.edu:5625 10 INTEGER     SERVER_COMM_SOCKETS 3
 mcat1 arsc.edu  mcat1.arsc.edu:5625 11 INTEGER     CLIENT_CONN_TIMEOUT 300
 mcat1 arsc.edu  mcat1.arsc.edu:5625 12 STRING_LIST HOST_BASED_AUTH_ENABLED_FLAG YES
 mcat1 arsc.edu  mcat1.arsc.edu:5625 13 STRING_LIST HOST_BASED_AUTH_ACCESS_MODE
ALLOW_ALL
```

These parameters can usually be left at their default values. The `SERVER_PRE_SPAWN_COUNT_MIN` and `SERVER_PRE_SPAWN_COUNT_MAX` can be increased on the MCAT Servers as they are dedicated servers and because a large number of concurrent users is expected, for example:

```
> modifyLocation mcat1@arsc.edu changeConfigValue SERVER_PRE_SPAWN_COUNT_MIN::50
> modifyLocation mcat1@arsc.edu changeConfigValue SERVER_PRE_SPAWN_COUNT_MAX::100
```

As changes are made via Acommands or the Java Admin, they are automatically sent to all the MCAT Servers and SRB Agents affected by the changes. In order to avoid this behavior, multiple Acommands can be combined into a single bulk file using the `-file` argument. The bulk file can then be executed using `executeBulkCommands`.

5.4 HPC Clients

HPC Clients such as HPC login nodes, utility servers, or user workstations will have to be prepared in order to interact with the SLM solution. The following sections talk about these preparations.

5.4.1 SRB Installation/ Configuration

Any system interacting with the SLM solution has to install at least one of the three SRB Clients recommended for HPCMP – the Scommands, the Preload Library, or the Java Client. Section 10 provides additional detail on each of those clients. The SRB Installation Guide provides detailed instructions on how to install and configure these clients.

For the HPCMP SRB Federation there are a few important items to consider:

1) The `-host` argument specified when establishing a session with SRB should point to a comma-separated list of each DSRC's MCAT Servers. For example, when using SRB at ORS, it may look like this:

   ```
   > Sinit -host mcat71.arsc.edu,mcat72.arsc.edu -user scheder@HPCMP.HPC.MIL
   ```

   In this way, proper load-balancing and failover between the MCAT Servers is ensured. Alternatively, section 11.2.4 provides a mechanism through aliases to shorten and simplify the users' login command.

2) An alternative to initiating an Scommand session with `Sinit` can be accomplished using `Sshell`. The above example using Sshell may look like this:

   ```
   > Sshell -host mcat71.arsc.edu,mcat72.arsc.edu -user scheder@HPCMP.HPC.MIL

   Welcome to SRB.
   Your User Name is "scheder@HPCMP.HPC.MIL".
   Your Default Classification is "SBU".
   Your Default Resource is "arsc.u2".
   Your Home Collection is "/archive/scheder".
   ```

   The advantage of using `Sshell` over `Sinit` is that the connections to the SRB Servers remain intact between Scommands. This results in a much more responsive interaction with SRB. This is especially important when Kerberos authentication is used and a re-authentication to the Kerberos KDC becomes necessary for every Scommand when `Sinit` is used. `Sshell` only authenticates one time with the KDC and then preserves the Kerberos session for all subsequent Scommands.

3) The Preload Library should be configured in the user login script by setting the appropriate environment variables. A detailed example for all supported operating systems is given in the script `$SRB_HOME/opensrc/preload/bin/srbpreload.sh`. A simple example on 64-bit Linux may be as follows:

   ```
   SRB_HOME=/opt/nirvana/srb
   SRB_PREFIX_DIR=/srb
   ```

```
srbServer=mcat71.arsc.edu,mcat72,arsc.edu
srbUserAtDomain=$USER@arsc.edu
srbAuthScheme=KERBEROS_AUTH
srbReadWriteBufferSize=10000000
LD_PRELOAD=$SRB_HOME/opensrc/preload/bin/64/srbpreload.so
```

It should be noted that not all operating systems properly deal with a mixed environment of preloading 32-bit and 64-bit applications simultaneously. In Linux, for example, the user can either set `$LD_PRELOAD` to a 32-bit or 64-bit library. This causes only 32-bit or 64-bit applications to function properly within the preload shell. If both, the 32- and 64-bit preload library are given, both 32- and 64-bit applications will run, but there is always an error message saying that either the 32- or 64-bit library could not be loaded. Solaris, on the other hand, handles preloading by setting two environment variables `$LD_PRELOAD_32` and `$LD_PRELOAD_64` and then uses the appropriate one depending on the application.

4) All three SRB Clients – Scommands, Preload Library, and Java Client – are designed to work with the HPCMP Kerberos authentication mechanism. But they will only be able to authenticate if an HPCMP Kerberos credential cache already exists before these clients are executed. This can either be accomplished by forwarding the Kerberos TGT from the user's workstation or by running kinit on the client machine where the SRB Clients are to be used.

5.5 HPCMP Software – Kerberos/SSH

The HPCMP maintains its own version of Kerberos – dubbed HPCMP Kerberos. HPCMP Kerberos generally consists of client-side libraries and applications with PIPE or C-API credential cache storage. SRB interfaces with HPCMP Kerberos by statically linking the HPCMP Kerberos libraries into the SRB HPCMP Kerberos driver.

If Kerberos authentication or data transfer is to be used, both parties participating in the communication (i.e., client and server; or two servers) need to be configured for HPCMP Kerberos. When using the term "client" in the context of these sections, "user" could be substituted, meaning an individual person. When referring to "server", "Location" could be substituted, meaning a machine.

The configuration of HPCMP Kerberos is outside the scope of this document although there are some instructions on Kerberos setup in the SRB Installation Guide. These instructions apply to both MIT Kerberos and HPCMP Kerberos.

The following sections provide some quick configuration instructions as they pertain to HPCMP Kerberos authentication.

5.5.1 HPCMP Kerberos User Configuration

There are two ways to associate a Kerberos Authentication Scheme to a user: either at or after user creation. The following commands demonstrate this process. To create a user with Kerberos Authentication enabled, the registerUser command is used as follows:

```
> registerUser scheder@HPCMP.HPC.MIL KERBEROS_AUTH 'scheder@HPCMP.HPC.MIL' staff x x x
```

To enable Kerberos Authentication for an existing user, the modifyUser command is used:

```
> modifyUser scheder@HPCMP.HPC.MIL insertAuthentication
KERBEROS_AUTH::scheder@HPCMP.HPC.MIL
```

This should result in Kerberos authentication being attached to the `scheder` user. This can be verified using the following command:

```
> SgetU -l -auth KERBEROS_AUTH scheder@HPCMP.HPC.MIL
--------------------------- RESULTS -----------------------------
User Name: scheder
User Domain: HPCMP.HPC.MIL
User ID: 500001
User Status: SRB_OBJECT_ENABLED
User Type: staff
User Authentication Type: KERBEROS_AUTH
User Authentication Info: scheder@HPCMP.HPC.MIL
User Classification: SBU
Home Collection Name: /archive/scheder
Default Resource Name: arsc.u2
```

5.5.2 HPCMP Kerberos Server Configuration

On the server side, there are two configuration steps required: a) creation and transfer of a keytab file; and b) attaching Kerberos authentication to the server's SRB Location.

The keytab file creation and transfer onto the server are described in the SRB Installation Guide. The server-side attachment of Kerberos authentication is performed as follows:

```
> modifyLocation seawolf@arsc.edu insertAuthentication KERBEROS_AUTH::seawolf@ARSC.EDU
```

This should result in Kerberos authentication being attached to the seawolf Location. This can be verified using the following command:

```
> SgetL -l -auth KERBEROS_AUTH seawolf@arsc.edu
--------------------------- RESULTS -----------------------------
User Name: seawolf
User Domain: arsc.edu
Location Net Address: seawolf.arsc.edu:5625
Location IS MCAT Flag: 0
Location System and OS information: <truncated>
User Authentication Type: KERBEROS_AUTH
User Authentication Info: seawolf@ARSC.EDU
User Address:
User Phone:
User Email:
-----------------------------------------------------------------
```

5.5.3 HPCMP Kerberos Authentication

With both user- and server-side Kerberos configuration in place, the following command can be used to authenticate a user (e.g., `scheder`) with a preexisting credential cache (as initiated by kinit) to a Kerberos-enabled server:

```
> Sinit -host mcat1.arsc.edu -user scheder@HPCMP.HPC.MIL -auth KERBEROS_AUTH
Welcome to SRB.
Your User Name is "scheder@HPCMP.HPC.MIL".
Your Default Classification is "SBU".
Your Default Resource is "arsc.u2".
Your Home Collection is "/archive/scheder".
```

All subsequent communications are either plain text, integrity-checked, or encrypted depending on the setting of the `-comm` argument of `Sinit`.

5.6 Data Movers

To offload I/O traffic from the main SLM Servers, some DSRCs have implemented Data Mover servers, which are FC attached to the SAM-QFS file systems. The Data Movers can continue to be utilized in the new SLM architecture. There are a number of items to consider:

- Data Movers need an SRB Agent installation.
- For every mount point of a SAM-QFS file system an SRB Physical Resource must be configured on the SRB Agent/ Location where the file system is mounted.
- SRB Cluster Resources must be configured correctly for Data Movers to use the shortest possible I/O path.
- The Cluster Resource configuration must contain all the Physical Resources for a given SAM-QFS file system for direct I/O and automatic failover.
- Sync Daemons must only be deployed and configured once per SAM-QFS file system. If multiple Sync Daemons were enabled for a single file system, it would result in unnecessary MCAT loads and possible metadata conflicts.

Once Data Movers are integrated into the SLM solution as described above, they can utilize Scommands to access the Global Namespace and perform data movement operations. For operations involving locally mounted file systems, the FC network will be utilized ensuring the fastest possible I/O path.

5.7 STIG Compliance

The applicable STIGs and their compliance are outlined below. SLM consists of four primary components:

- Oracle – compliance with the Database STIG is inherited since Oracle is already installed at various DSRCs. The DB Checklist is also part of the inherited compliance. CSA scripts are used to verify compliance.
- Solaris – compliance with UNIX STIG is inherited since Solaris is already installed at various DSRCs.
- SAM - QFS: same as Solaris.
- SRB – The Application STIG pertains primarily to Application development. SRB has been evaluated against all SLM-related IA Controls, and the analysis is documented as

part of the DIACAP Implementation Plan and Validation Plan, the SLM Cookbook, and the SLM Artifact documents.

6 SRB – SAM-QFS INTERACTIONS

SRB and SAM-QFS will interact in the following ways:

- Brokering of user file operations. SRB will broker the file operations (transfer, remove, namespace changes) between users and the managed SAM-QFS archive file systems.
- Information Lifecycle Management (ILM) policy management. SRB will initiate SAM-QFS operations in order to execute ILM policies.
- Metadata Synchronization. To efficiently execute the above operations, the MCAT Database will acquire and maintain complete awareness of relevant file metadata for all files in managed SAM-QFS file systems. Such synchronization operations between existing SAM-QFS file systems and the MCAT discussed in the sections below require the use of SAM-QFS version 5 or higher.

6.1 Local File System Access

Authenticated SRB Clients using secure and integrity-checked HPCMP Kerberos sessions (maintained in the Kerberos PIPE cache and SRB session file) communicate with SRB Agents to access their files. SRB Agents in turn communicate with the underlying file systems – SAM-QFS in the case of this solution.

6.1.1 Direct File System Access vs. SRB-Controlled Access

Direct local file system access after the initial registration of pre-existing files into SRB will be disallowed. The following paragraphs support this position.

One advantage of direct file system access is that applications can continue to interact with the local file system as they have in the past. Users enjoy the familiar looking UNIX mode flags together with user and group ownership of their files. Browsing and listing directories and files is very responsive to interactive user shell commands.

However, there are some severe limitations in the UNIX access control model and the usage of UNIX ACLs is very complicated. There is no inheritance (without UNIX ACLs), which would allow for access control to be managed from a small number of directories. UNIX users are typically limited to belonging to a maximum of 16 UNIX groups. It is altogether impossible to enforce mandatory access controls to prevent the declassification of files from high to low.

Although interactive user shell commands are very responsive, they are also very limiting. The UNIX file system structure is fixed and forever tied to a single, local, hierarchical organization. With SRB, those ties are loosened and the flexibility of relational database queries is introduced into the organization of user files – across all DSRC sites.

SRB command-line utilities (Scommands) are available for many UNIX shell commands (see section 10.2) and in many cases offer additional capabilities over UNIX (see for example Sscheme, Smv, Sreplicate).

Another perceived advantage of such direct access is performance. Without the SRB Agent process acting as a mediator, file access should be faster in particular when working with many smaller files. When going through SRB, there is overhead in communicating with the MCAT Servers for metadata operations. But in fact, tests have shown that large file operations may actually be faster when going through SRB compared to using NFS mounts as shown in internal tests. Small file operations are made more efficient by using SRB's bulk protocol. They may still lag behind NFS performance as shown in internal tests but are continuously being improved.

The synchronization of direct local file system access with an SRB Federation is the most significant issue. Although the SRB Sync Daemon can handle some out-of-synch conditions (see section 6.2), the following is a list of issues, which are not yet resolved when synchronizing the two:

- There are two authority-granting entities when direct local file system access is permitted. This invariably results in out-of-sync conditions for the access permissions. For example, access permission and owner changes in SRB are not reflected in the underlying file system.
- Files can be logically moved or renamed in SRB but are not automatically moved or renamed in the underlying file system, which invariably results in the two namespaces getting out-of-synch. Such automatic moves or renames would cause unacceptable performance degradation.
- In Sync Daemon walk mode, the movement or renaming of files in the local file system results in un-registration and registration of the Data Objects in SRB causing non-file system metadata (and access permissions) to be lost. The Sync Daemon real-time mode can recognize such moves but only works for SAM-QFS file systems.

6.1.2 UNIX Group Mapping

It is not anticipated that UNIX groups will map directly to SRB Groups because there are many groups in use at the different DSRCs, which are either user groups (only containing a single UNIX user) or are not coordinated between DSRCs. With the GIDs not coordinated among DSRCs, there is a risk that files retrieved from the global namespace at the non-originating site may be owned by the incorrect group. Hence, it is recommended that a consistent set of new SRB Groups be created, which can then be utilized within and across centers. Examples for such SRB Groups are provided in the following table:

Table 6-1. SRB Group Examples

SRB Groups within DSRC	SRB Groups across DSRCs
ORS	HPCMP
ORS Admins	HPCMP Admins
ORS Freshwater Project	"Project X spanning DSRCs"
AFRL	ITAR Controlled
AFRL HEART Project	SBU

6.1.3 GID Requirements

Due to the different access control mechanisms between local UNIX file systems and SRB, GIDs cannot easily be mapped into SRB Groups because:

1. SRB does not support the concept of group ownership,
2. Group permissions would have to be applied via ACLs, and
3. It is not practical to apply such ACLs to every single file.

Hence, it would be preferred to apply ACLs to just a few critical branching points within the Global Namespace.

It is proposed that the following steps be implemented to extract and utilize the local file system GID information:

1) Each DSRC creates a list of non-trivial GIDs/groups names (see section 6.1.2), which excludes single-user groups and groups that are not in use.
2) The local file system branching points are determined using a script. Branching points are directories or files in the file system where the GID is different from the GIDs in the files and directories at the same level.
3) The output of the script in step 2 is then used to create SRB ACLs.

Note that for reference purposes the group names are stored as an attribute in the MCAT database.

6.1.4 Chown and Chgrp

Part of the discussion on direct file system access is to consider how local file ownership (UID and GID) is preserved. The SRB Sync Daemon initially uses this local file ownership information to associate the file and directory entries in the MCAT database with the appropriate owners (see section 3.7.2).

The SRB Agent runs under a community user (i.e., “root” see section 6.1.5) in order preserve the privilege to chown file ownership. This would enable local file ownership to be preserved as the SRB Agent would then be able to change the file ownership upon file creation. The MCAT database contains mappings for user names between sites and hence the SRB Agent can determine the correct user name for each site. After performing the chown, the SRB Agent can look up the primary group of the file owner and perform a chgrp.

Therefore it will be possible to preserve local file ownership and maintain local file system access. However, there are some side effects, as discussed in section 6.1.1 above.

When retrieving files from SRB, the UNIX file ownership (user/group) and mode will be reapplied.

6.1.5 Root vs. Unprivileged User

As SRB Agent processes interact with the local file system, they need to have permissions to access files and directories locally. There are two modes how SRB Agent processes can be run

– either as “root” or as an unprivileged UNIX user (e.g., “srb”). In the HPCMP solution the SRB Agent processes run as “root”.

When running as an unprivileged UNIX user, the agent process needs to obtain file and directory access via group memberships. The UNIX operating systems have a limitation on the number of groups that a user can be a member of – typically 16. If there are more than 16 UNIX groups in use, which is the case for all DSRC sites, the “srb” user will not be able to obtain access to all user files via group membership. However, RBAC can give the “srb” user elevated privileges to read/search/write files and directories and to chown files. The privileges are PRIV_FILE_DAC_READ, PRIV_FILE_DAC_SEARCH, PRIV_FILE_DAC_WRITE, and PRIV_FILE_CHOWN.

6.2 Sync Daemon Functionality and Limitations

The Sync Daemon will be used to synchronize the MCAT Database and the metadata from the underlying SAM-QFS file systems. It can operate in Walk or in real-time mode. The following steps describe how to initiate synchronization for SAM-QFS file systems with SRB:

a. Unmount SAM-QFS file systems.
b. Remount SAM-QFS file systems with the -o sam_db option.
c. Start Sync Daemon in walk mode to perform one-time initial file system registration in MCAT.
d. After walk mode has completed, start Sync Daemon in real-time mode to account for any changes that the walk mode missed and to perform all synchronization from thereon out.
e. Analyze potential errors from real-time mode run to ensure synchronization is complete. Errors might occur due to double-application of changes in the file system.

The following sections will describe the walk mode and the real-time mode of the Sync Daemon in more detail.

6.2.1 Sync Daemon Walk Mode

The Sync Daemon walk mode is used to perform either a one-time initial registration, or subsequent re-synchronization, of SAM-QFS metadata into the MCAT database. It functions as follows:

The Sync Daemon stats (or in the case of SAM-QFS sam_stats) each directory in a locally mounted file system. It then performs a readdir (man 3 readdir) of the directory contents and registers the returned d_names (file or directory name) in SRB. It recursively walks an entire directory tree in this manner until it reaches the initial starting point again.

The Sync Daemon synchronizes attributes such as the following from the local file system into the Global Namespace (i.e., the MCAT Database): file/dir owner by user name and group name,

creation time, modify time, last access time, the UNIX mode, and the SAM-QFS HSM status flags such as the existence of the file in the SAM-QFS disk cache and the location and state of any archive copies. The user name is mapped into an SRB user ID and name according to the procedure described in section 3.7.2.

To use the Sync Daemon for re-synchronization, the following steps need to be strictly observed:

1. Ensure the issue that caused the sam-fsalogd failure is resolved and that the sam-fsalogd daemon is running and remains running.
2. Run the Sync Daemon in walk mode, which will synchronize the file system with the MCAT database. Any events occurring while running the Sync Daemon in walk mode will be logged by the sam-fsalogd daemon, but not applied to the MCAT database.
3. Once the walk mode is finished, run the Sync Daemon in real-time mode. This will apply the changes in the sam-fsalogd daemon logs to the MCAT database.

6.2.2 Sync Daemon Real-Time Mode

The real-time mode will function as the production mechanism to synchronize SAM-QFS with the MCAT Database. This mode currently only functions in connection with SAM-QFS file systems v5.0 or higher. It relies on the underlying file system's capability to provide event notifications to the Sync Daemon. In the case of SAM-QFS, the Sync Daemon utilizes the sam-fsalogd daemon to write-out log files of any file system modification events. It then reads this event log and notifies the MCAT Database of any changes.

If the sam-fsalogd daemon is disabled, the SAM-FS daemon (sam-fsd) will attempt to restart the daemon. In the meantime, the Sync Daemon can be run in walk mode to re-synchronize the file system changes while the sam-fsalogd daemon was disabled.

The following events are currently supported by the sam-fsalogd daemon: file creation, attribute change (UID, GID), file close, file rename, file remove, residency changed to offline, residency changed to online, archive copy made, archive copies stale, archive copy changed, file was restored, and file system was unmounted.

However, not all modifications on the SRB side are reflected back into the SAM-QFS file system. Changes that are not currently propagated back are changes to the SRB owner, ACLs, and logical moves. The authorization information is kept in Oracle independent of the underlying file system's settings. This enables a persistent access control mechanism, which prevails across sites, file system migrations, and even storage resources that do not support access control such as tape. Logical moves in SRB allow for a fast rename or re-organization mechanism without having to change the organization of the underlying storage. This mechanism also works across sites, different file systems, and storage resources that do not support naming or hierarchical organization of files such as content addressed storage or cloud storage. A physical move (also called migration) performed in SRB is reflected in the local file system(s).

6.2.3 Sync Daemon Check Mode

The Sync Daemon check mode is similar to the Sync Daemon walk mode except that no changes are made to the MCAT database and a report is generated. This report can then be used to verify that the file system is in sync with the database.

The next section elaborates on which attributes/events in SAM-QFS are synchronized with SRB.

6.3 Metadata Synchronization

A SAM file system can present any of the following metadata for each file via API calls, any of which is registered into the MCAT initially and then synchronized on a continuous basis:

- File ownership (user name).
- File ownership (group name).
- File mode (permissions).
- File project ID.
- File inode number.
- File inode generation number (Since inode numbers can get reused, a generation number for each inode is additionally tracked to help preserve use of inode as a unique identifier, particularly in archiving and staging operations).
- File size in bytes.
- Time of last access, last modification and **status** change.
- Time **attributes** last changed.
- Time inode was created.
- Time file last changed residence.
- Checksum value and algorithm used.
- Checksum properties (whether to generate on archive, whether to use when staged).
- Staging status for file (whether stage is pending or whether last stage attempt failed).
- Online disk cache storage parameters (whether file is online).
- File release parameters (release after archive, release never).
- File staging parameters (file should never be staged, file should be stage associatively).
- Miscellaneous query-able attributes tracked in memory that are not stored persistently in inode (file is rearchiving, file has one archive copy, file should be archived immediately).
- Volume serial number (VSN), media type, position.

The following is a list of attributes, which remain only inside the SAM file system and are not registered into MCAT or synchronized on a continuous basis:

- File ACL (whether one is present, whether it is set to default).
- Number of links to the file.
- File size in blocks.
- Online disk cache storage parameters (stripe width, stripe group, partial size, whether a files partial extents are online, allocate ahead value, whether direction should be used for file access).
- Properties for segmented files (segment size, number, segments to stage ahead).
- The admin set, which, for the purpose of quotas, allows for site-specific grouping of files above and beyond groupings that user and group IDs allow.
- WORM file system properties (whether a file is WORM, whether WORM file is in read only state, WORM start time, and WORM duration).

- File archive parameters (archiving OK even if file open for write, inconsistent copy OK if file modified during archival).
- File release parameters (release partially).
- Miscellaneous query-able attributes tracked in memory that are not stored persistently in inode (file requires data verify, file is an AIO character device).

6.4 HSM Operations

The following sections discuss the HSM operations supported by the SLM solution and how SRB and SAM-QFS interact within these operations.

6.4.1 Archive

In the SLM solution, SRB is the primary administrator of archive policy. Site archiving policies in SAM-QFS will be adjusted such that they do not autonomously initiate archive operations. Instead the SRB ILM Daemons, via SAM-QFS drivers, will initiate archiving via API calls orchestrated via ILM policies. SAM-QFS retains its media management function and archiving directives that address efficient delivery. The layout of data on archive media remains intact. Site assignments of SAM archive copy roles (e.g. “copies 1 and 2 are local, copy 4 is DR”) also remain intact.

Additionally, the SLM solution offers users an Sarchive command to archive a set of files as SAM-QFS offers today via its archive(1) command.

6.4.2 Release/Purge

By default, with the SRB metadata attribute Purge_Behavior set to ‘no’, the existing SAM-QFS release behavior is not affected. When set to ‘yes’, then the “release after archive” flag will be set on a per-file-basis. SAM-QFS will retain its ability to release data from cache independent of what the user specifies for Purge_Behavior.

Additionally, the SLM solution offers users an Spurge command to release a set of files as SAM-QFS offers today via its release(1) command.

6.4.3 Stage

The existing staging behavior of site SAM-QFS environments will remain unchanged. User accesses of offline files in the SLM environment will still transparently trigger their staging by SAM-QFS (or in the case of “stage never” files, their delivery to the application without being staged). Additionally, the SLM solution will offer users an Sstage command to stage a set of files as SAM-QFS offers today via its stage(1) command. The SLM solution will also allow specifying an ILM policy that automatically stages a set of files according to ILM criteria.

6.4.4 Recycling

SAM-QFS recycling policies will remain unchanged. The SLM solution is designed to be unaffected by any rewriting of archive copies on archive media which the SAM recycling process causes to occur.

6.4.5 Disaster Recovery (DR)

The DR copies are created in a transparent fashion to SRB because they are no different from any of the other SAM-QFS copies. In the case of a disaster, the current recovery strategy demands that any SAM-QFS file systems be re-created on the replacement servers (the servers replacing the function of the lost servers) using the last good `samfsdump`. It needs to be ascertained that the mount points are exactly the same as they were on the original production servers. The DR copy can then be accessed through those restored file systems. An SRB Agent on the replacement servers will serve as the alternate access path for those file systems through SRB. This requires a simple configuration change in SRB to re-point the old Locations to the new SRB Agents on the replacement servers.

After re-mount, the SAM-QFS file systems need to be checked for consistency with the archive. There is a very real possibility that the archive has changed since the last `samfsdump`. In order to account for the differences, the SRB Sync Daemon can be run in check mode (read-only), which will cause a report to be generated. The report will contain a listing of all the files which were created, modified, or deleted in MCAT since the disaster event. This report can then be used to augment the established recovery procedures for the SAM-QFS file systems. Finally, the same process described in the Sync Daemon walk mode, section 6.2.1, can be used to ensure the synchronization of the MCAT with the SAM-QFS file system.

Users of a disabled MCAT Server will need to be re-pointed to one (or several) of the other MCAT Servers. This is typically accomplished automatically as SRB Clients and SRB Agents should have a comma-separated list of MCAT Servers in their configurations.

7 ADMINISTRATION

7.1 User Administration

The following commands are a few examples that illustrate the user administration using SRB Acommands, which can be executed interactively or via scripts that can be integrated into the pIE environment:

1) create a new “HPCMP.HPC.MIL” domain:

```
> registerDomain HPCMP.HPC.MIL root
```

2) create a new user “scheder” in that domain:

```
> registerUser scheder@HPCMP.HPC.MIL KERBEROS_AUTH "" staff "address" "phone"
"email"
```

3) register another domain “STORAGE.HPC.MIL”:

```
> registerDomain STORAGE.HPC.MIL root
```

4) add “STORAGE.HPC.MIL” domain alias to “scheder” user:
```
> modifyUser scheder@HPCMP.HPC.MIL insertAlias scheder@STORAGE.HPC.MIL
```

5) add alias “tino@HPCMP.HPC.MIL” to user “scheder”:
```
> modifyUser scheder@HPCMP.HPC.MIL insertAlias tino@HPCMP.HPC.MIL
```

6) change alias “scheder@STORAGE.HPC.MIL” to “tino@STORAGE.HPC.MIL” for user “scheder":
```
> modifyUser scheder@HPCMP.HPC.MIL changeAlias
scheder@STORAGE.HPC.MIL::tino@STORAGE.HPC.MIL
```

7) change primary alias for user “scheder” to “tino@HPCMP.HPC.MIL”:
```
> modifyUser scheder@HPCMP.HPC.MIL changePrimaryAlias tino@HPCMP.HPC.MIL
```

8) remove alias “scheder@HPCMP.HPC.MIL” from user “tino":
```
> modifyUser tino@HPCMP.HPC.MIL deleteAlias scheder@HPCMP.HPC.MIL
```

9) disable user “tino”:
```
> modifyUser tino@HPCMP.HPC.MIL disable
```

10) print list of files and directories owned by user “tino” to stdout:
```
> SgetD –owner tino@HPCMP.HPC.MIL
```

11) change ownership of two files to user “admin”:
```
> Schown admin@HPCMP.HPC.MIL file_1 file_2
```

12) recursively change ownership of a directory to user “admin”:
```
> Schown –R admin@HPCMP.HPC.MIL directory_1
```

Note that all Acommands (i.e., registerXXX, modifyXXX) support a –file argument, which can be used to aggregate the commands into a batch file. They can then all be executed in bulk using executeBulkCommands. For more information on the various Acommands used and for additional Acommands available for user lifecycle management please refer to the SRB Administration Guide.

7.1.1 SRB Super User Handling

Generally, Team GA recommends that the SRB Super User would only be utilized during SRB installation and initial configuration. Afterwards, the system should be administered by named administrative SRB user accounts. The following is a recommendation on how the SRB Super User can be handled.

KERBEROS_AUTH is configured with multiple Kerberos user principal mappings for the SRB Super User. These mapped principals now have the privileges of the SRB Super User. This has the following advantages:

- There are multiple named Kerberos user accounts who can log into SRB as the Super User.
- There is no need to convey the passwords through e-mails or other transfer mechanisms.
- If a system administrator leaves the organization his/her principal will be automatically disabled.

Privileged functions may be executed remotely once a user has obtained their normal Kerberos credentials and has authenticated as the SRB Super User. The audit function is active for every session as long as auditing is enabled for SRB. The procedure for SRB Super User login is as follows:

- The users (with administrative SRB privileges) log in using their Kerberos credentials.
- The users next execute an "Sinit -user super@root -auth KERBEROS_AUTH -gssup <kerberos_user_principal>"

7.2 Kerberos Keytabs

Kerberos keytabs are used to prove the authenticity of the SRB services (e.g., MCAT Servers, SRB Agents, ILM Daemons, and Sync Daemons) to end users or other SRB services. It is recommended that the SRB service keytab be stored in a private keytab under `$SRB_HOME/config` called `krb5.keytab`, which must be owned by `srb:srb` and have a UNIX mode of 400.

The private keytab must only be readable by the UNIX user/group running the SRB services (i.e., srb:srb). This will protect the keytab from unauthorized access.

7.3 Disaster recovery

The administrative responsibilities around disaster recovery as they relate to the SLM solution are covered in section 6.4.5.

7.4 Failover

The following sections elaborate on the different failover scenarios supported by the SLM solution. Each section provides step-by-step instructions on how administrators can configure and perform these failover scenarios.

7.4.1 MCAT Server Failover (automatic, local)

The objective for this scenario is to prevent any SRB Federation outage in case one of the MCAT Servers fails. SRB components can be configured to automatically load balance and fail over between multiple MCAT Servers. The setup for SRB Agents and SRB Clients to utilize multiple MCAT Servers is shown in detail in sections 5.2.2.1 and 5.4.1 respectively. The load balancing and fail over happens automatically if SRB Agents and SRB Clients are configured as shown in those sections.

Steps for automatic MCAT Server failover:

1) Perform "`srb init`" for each SRB Agent using a comma-separated list of MCAT Servers.
2) Perform "`Sshell`" (or equivalent) for each SRB Client using a comma-separated list of MCAT Servers.

7.4.2 MCAT Server Failover (manual, remote)

If the entire Oracle RAC database – and hence all MCAT Servers – at a site fail, the SRB Agents and SRB Clients can be re-pointed to the MCAT Servers at another site.

The most efficient way to accomplish this re-pointing, is to change the IP address mapping for the MCAT Servers in the local DNS setup. The local MCAT Server host names would simply have to be re-pointed to the IP addresses of the remote MCAT Servers.

Steps for manual MCAT Server failover:

1) Re-point MCAT Server host names in DNS to remote MCAT Server IP addresses.
2) Restart SRB Agents to pick up the new IP addresses.
3) Notify users to logout and log back into their systems to pick up the new IP addresses.

7.4.3 SAM-QFS Archive Copies

The primary fail over mechanism for user data in the HPCMP is the creation of multiple HSM copies at local and remote sites. These copies are created by SAM-QFS with SRB initiating the copy process based on ILM policies and user-controlled metadata attribute settings.

SAM-QFS will automatically fail over to other HSM copies if the primary copy is inaccessible or corrupted. This mechanism happens transparently to the SRB Agents interacting with the SAM-QFS file systems.

The following steps need to be performed to initiate the creation of multiple HSM copies:

1) Deploy ILM Daemon to each SLM server, which manages a SAM-QFS file system.
2) Configure ILM Daemon with archival and DR policies as described in sections 0 and 6.4.5.
3) Enable ILM Daemon.

7.4.4 SRB Agent Failover (manual)

If a local SRB Agent fails, administrators can bring-up another server as the failed SRB Agent. This server can either be local or at a remote DR facility. Aside from restoring the SAM-QFS file systems at the new servers only the SRB Agent's Location host name needs to be changed.

The following steps are required for manual SRB Agent fail over:

- Restore SAM-QFS file system(s) as described in section 6.4.5.
- (Optional) Change the host name for the failed SRB Agent. This is only necessary if the replacement server has a different host name or IP address. For example, if "`seawolf`" had failed and the replacement server's hostname were "`seawolf_dr`", one could use:

  ```
  > modifyLocation seawolf@arsc.edu changeNetAddress seawolf_dr@arsc.edu:5625
  ```

- Ensure SRB software is installed and configured on the new SRB Agent host.
- Initialize the replacement SRB Agent using "`srb init`" as described in section 5.2.2.1.
- Start the replacement SRB Agent.

7.4.5 SRB Agent Failover (automatic)

SRB has a built-in mechanism to manage and synchronize multiple replicas (i.e., synchronized copies) of files across multiple file systems – local or remote. If multiple Replicas exist for a Data Object in SRB, the SRB protocol will automatically fail-over to a working copy if its primary local copy is inaccessible.

Although this mechanism will not be the primary method for multi-copy file storage in the HPCMP, it can be easily setup on a case-by-case basis. There are generally two methods how to create Replicas in SRB: 1) on demand or 2) using ILM policies.

The on-demand method requires the following steps, which can be performed by every user with access to multiple file systems:

1) Issue a replicate command for a file or directory specifying a target resource for the to-be-created Replica. An Scommand example, which replicates the entire “Project 1” Collection to the “erdc1” file system is given below:
```
> Sreplicate –R –rsrc erdc1 "Project 1"
```

The ILM policy method would be performed as follows:

1) Deploy ILM Daemon to either the source or target site for the data replication.
2) Configure ILM Daemon with replicate policy and appropriate target resource (i.e., file system).
3) (Optional) Configure ILM Daemon with asynchronous synchronization policy.
4) Enable ILM Daemon.

7.5 Debugging Applications

The SLM solution consists of many different components, which interact with each other. When a problem occurs it is essential to determine the component responsible for the problem. Sometimes the problem may also lie in the interaction of two of the components.

The first indicator of a problem is some unexpected behavior – slow performance, an error message, or an output that was not expected. SRB provides a number of mechanisms to resolve the root cause of a problem.

7.5.1 SRB Error Messages

If an operation fails, SRB will return a negative numeric error code. Error codes can be interpreted using the `Serror` Scommand, for example:

```
> Serror -242
SRB_RESOURCE_INFO_FORMAT_ERROR "A format error for Resource Quotas occurred. Please specify quotas as a percentage of space usage in the following format - first the soft quota then the hard quota: 60::80."
```

The error interpretation usually consists of a detailed error description and a suggestion how to rectify the situation that caused the error. Under certain circumstances, SRB will return an error stack as part of the SRB session. For example, the following error stack is returned when an attempt is made to write into a file system without appropriate OS permissions:

```
> Sput -rsrc arsc.u2 /home/scheder/localFileName srbFileName
FILE_CREATE_ERROR "An error occurred during file create."

ERROR[0] DISK_FILE_EACCES "OS level. Permission denied."
    SvrFileCreate()
    DISK_FILE_EACCES "OS level. Permission denied."
```

The error stack contains information about the API, which returned the error. Each level in the API stack may return a different error message but with the error stack it is easy to trace the error back to its origin.

OS level-errors (such as the one above) can point the user or administrator to a lower level component such as the underlying operating system or file system (i.e., SAM-QFS).

A comprehensive list of SRB error messages with their detailed descriptions and suggested problem resolutions can be obtained from `$SRB_HOME/include/SrbErrorExtern.h`.

7.5.2 SRB Debug Levels

If no error message is returned or the behavior of the application is not as expected, it is often helpful to enable a more verbose command output. SRB provides six different levels of verbosity for all of its components – 0 through 5. A level of 0 means that no verbose debugging information is printed and that only error messages are returned. A level of 5 is the maximum network-level debugging output.

A level of 1 can usually be used to verify whether the arguments were parsed properly or whether they were mistakenly expanded by the shell. Many times arguments values are also associated with the wrong argument simply by putting a space into an argument value but forgetting to enclose the argument in quotes. The following example demonstrates how the target Collection (i.e., “Project Collection”) was incorrectly parsed so that its first portion (i.e., “Project”) was interpreted as another `Local Dir/File Name` input:

```
Sput localFileName Project Collection -v 1
DEBUG:03-04-21.51.32.3306:11274: Sput : Verbose mode on
DEBUG:03-04-21.51.32.3336:11274: Sput : Directory Path ''
DEBUG:03-04-21.51.32.3657:11274:
DEBUG:03-04-21.51.32.3800:11274: Sput localFileName Project Collection -v 1
DEBUG:03-04-21.51.32.4109:11274: Arguments Value:
DEBUG:03-04-21.51.32.4133:11274: -v   Log Debug Level                  (1)
DEBUG:03-04-21.51.32.4418:11274:      Local Dir/File Name
(localFileName, Project)
DEBUG:03-04-21.51.32.4422:11274:      Dest Collection/DataObject Name   (Collection)
...
```

More verbose debug levels can also be utilized to interpret authentication errors. In some cases they may reveal an incorrect configuration.

All SRB Clients and SRB Servers can be configured to run at these debug levels. The configuration method may vary based on the component.

7.5.3 SRB Logging

If an error message and a higher debug level do not reveal the desired root cause of a problem, SRB provides a very detailed logging mechanism for all of its components. All of the SRB logs reside under `$SRB_HOME/log`. They may reside either on HPC clients (for Preload Library or Java Client logs), SRB Agent machines (i.e., the SLM Server, which may contain logs for Physical Resources and server process logs), or the MCAT Server(s) (for MCAT Database logs or MCAT server process logs).

Each of the logs will be written at a debug level from 0 through 5 as described in the previous section. The components can be configured to utilize a certain debug level when logging. Typically, for performance reasons, the debug level should be set to 0. However by default, debug level is being set to 1 to facilitate mining of Ssync and Silm logs. And, when users specify a higher debug level from an Scommand (e.g., Sput -v 3 localFile srbFile), then this higher log level propagates to all components that are involved in the transaction (e.g., oracle.log and srbServer.log on the MCAT Server; srbServer.log and samfs.log on the SLM Server). As this higher log level will only be used for the transaction in question, it is easy to find the logs associated with this transaction and to identify the problem.

The default behavior for SRB log files is to grow up to a certain size (10MB by default), then rename the log file with a ".bak" extension, and restart a new log file. On the second rollover, the old ".bak" file would get deleted and overwritten by the current log file.

The log file name for an SRB Location and Physical Resource can be changed like this:

```
modifyLocation seawolf@arsc.edu changeConfigValue 0::LOG_FILE::seawolf.log
modifyResource arsc.u2 changeConfigValue 0::LOG_FILE::arsc.u2.log
```

The rollover size can be specified for each log file. For an SRB Location and a Physical Resource, the following commands change the rollover log size to 20MB:

```
modifyLocation seawolf@arsc.edu changeConfigValue 0::LOG_FILE_SIZE::20000000
modifyResource arsc.u2 changeConfigValue 0::LOG_FILE_SIZE::20000000
```

SRB logging can also be configured so that `$SRB_HOME/log` represents a symbolic link to a log directory on a separate logging file system. This offloads logging activity from the root file system providing better I/O performance for reading the SRB binaries.

In order to accommodate the requirement of individuals with different roles having to get access to SRB log files, there are configuration parameters that can be adjusted to change the mode and UNIX group owner of the various log files. The following provides an example for changing the mode and UNIX group owner on the srbServer.log for the seawolf@arsc.edu Location:

```
modifyLocation seawolf@arsc.edu changeConfigValue LOG_FILE_MODE::640
modifyLocation seawolf@arsc.edu changeConfigValue LOG_FILE_GROUP::srbhelpdesk
```

7.5.4 Debugging SLM Performance

There are two broad performance symptoms that users may notice:

1. slow file transfer performance
2. slow metadata performance or database response time

Depending on which of those symptoms is observed, there are different approaches to be taken to troubleshoot the performance issue.

7.5.4.1 Slow File Transfer Performance

For slow file transfer performance, the troubleshooting should be started at the SLM Server where the data originated. The network connection from SRB Client to the SLM Server needs to be investigated as well.

The following are some troubleshooting methods that can be used in connection with slow file transfer performance:

1. The Solaris iostat(1M) command can be used to probe for possible bottlenecks in underlying storage, whether used for SAM or Oracle. Long service times (asvc_t column) and/or significant queueing (actv column) for devices that are used for latency sensitive functions such as SAM metadata or Oracle redo may point to a bottleneck there.
2. The Solaris prstat(1M) command can be used to observe general system CPU load and memory usage. This will help in identifying CPU and memory bandwidth bottlenecks.
3. For writes to T10000 tape media, if I/O's are being delivered in the optimal 2MB size (or multiple thereof), the w/s and kw/s columns in iostat will divide out to 2048KB (+/- rounding error).
4. Where permitted, some of the I/O-oriented utilities in the DTrace toolkit, a set of publicly available DTrace scripts not bundled with Solaris, may allow administrators to compare actual system I/O profiles against the expected profile and thereby uncover potential bottlenecks. As an example, such scripts may allow administrators to identify a throughput oriented application thought to be issuing I/O's that are sized and aligned to a RAID5 full stripe (generally the optimal I/O profile for throughput on RAID5 storage) when they are not actually doing so.
5. In order to observe network performance, the netstat (1M) command can be used. A high and growing number of Ierrs, Oerrs, and Collis when using "netstat -i" denotes a problem with one of the network interfaces.

7.5.4.2 Slow Metadata Performance

For slow metadata performance, the investigation should start with the MCAT Servers, the MCAT RAC Nodes, and the Oracle database. The following are some troubleshooting methods that can be used in connection with slow metadata performance:

1. The $SRB_HOME/log/oracle.log files on the MCAT Servers flag slow queries - meaning queries that take longer than 120 seconds to execute by default. This threshold can be

adjusted down to 1 second (using the OPERATION_ALERT_TIME parameter in $SRB_HOME/config/mcat.config) in order to capture additional slow queries. Those queries can be identified by the string "QUERY MAX TIME EXCEEDED". Those queries can then be analyzed and optimized by, for example, index creation.

2. The $SRB_HOME/log/srbServer.log files on the MCAT Servers contain very fine-grained timing information that is useful in order to detect how long certain function calls are taking. This can indicate a problem with those function calls. SRB Support should be notified of such occurrences or other error messages in the srbServer.log, which might indicate a problem. For example, srbServer.log flags problems with DNS queries, which are identified by an entry such as "Query host entry information took '20' seconds for 'mcat1.arl.hpc.mil', you may need to setup DNS properly".
3. The CPU and memory load on the MCAT Servers and MCAT RAC Nodes should be observed using, for example, prstat(1M). Additional memory or CPUs might have to be added.
4. Oracle OEM and Enterprise Manager provide alerts and performance tuning utilities that flag inefficient queries and detect I/O bottlenecks. They can also provide recommendations as to database tuning. These are probably the most valuable tools in order to detect slow metadata performance causes.
5. OS Watcher (OSW) is a tool available from the Oracle Metalink site [see OS Watcher User Guide ID 301137.1]. OSW is a collection of UNIX shell scripts intended to collect and archive operating system and network metrics to aid the support staff in diagnosing performance issues. OSW operates as a set of background processes on the server and gathers OS data on a regular basis by invoking UNIX utilities such as vmstat, top, memstat, netstat and iostat. Data collection intervals are configurable by the user. The utility is installed by root and requires root privilege to collect performance statistics.

7.6 SAM-QFS/ SRB manual resync and verification

In the event that a SAM-QFS file system gets out-of-sync with the MCAT database, the steps in sections 6.2.1 and 6.2.3 should be followed for re-synchronization and verification respectively. The Sync Daemon performs both tasks and can be invoked with a “LOCAL” walk or “CHECK” mode policy for those purposes.

7.7 Resource management

Resource management in SRB is a complex topic when taking into account different Resource Types, Resource Classes, Resource configurations, Physical Resources, Logical Resources, and Cluster Resources. A detailed explanation of Resource management is given in the SRB Administration Guide.

For the purpose of this document the discussion shall be limited to the most frequent tasks anticipated for the SLM solution.

7.7.1 New File System Configuration

A new SAM-QFS file system can easily be added to an existing SRB Location (i.e., SLM Server) by executing an Acommand such as the following:

```
> registerResource arsc.u3 'samfs file system' seawolf@arsc.edu
      '/export/archive/u3/?COLLECTION_STRIP_02/?DATANAME' archive 0
```

Appropriate access needs to be given to any new Resource. For example, the following command gives write access for the new Resource to the entire "ARSC.EDU" domain:

```
> modifyResource arsc.u3 changeAccess "ARSC.EDU::write"
```

7.7.2 Mount Point Changes

Starting with SRB 2010, only the variable portion of a file's path is stored in the MCAT Database. This enables administrators to easily change the mount point of a file system without causing any disruptions to file access in that file system.

The following command changes the mount point for the "arsc.u3" Resource to "/archive/u3":

```
> modifyResource arsc.u3 changePhysicalPath
'/archive/u3/?COLLECTION_STRIP_02/?DATANAME'
```

Now the administrator simply un-mounts the file systems and re-mounts it under "/archive/u3".

7.7.3 Consolidation of Multiple File Systems

SRB can perform the task of consolidating multiple file systems without disrupting user file access or changing the directory structure that users are used to seeing in any way.

The following command migrates all files under the "/projects" Collection from the arsc.u2 file system to the arsc.u3 file system:

```
> Smv -R -srsrc arsc.u2 -rsrc arsc.u3 /projects /projects
```

In this fashion multiple file systems can be consolidated into a single new file system enabling the retirement of the old file systems.

7.8 Maintenance tasks

The following sections describe repetitive maintenance tasks required of SLM administrators.

7.8.1 SAM-QFS Media Recycling

The SLM solution will experience recycling as simply a background change in the location of the archive copy of files that are relocated by the recycler. It will be notified of the change via the

SAM event logging mechanism at which point the MCAT will be updated to reflect the new location. Therefore, there are no changes required to existing recycling processes.

7.8.2 SAM-QFS Media Migration

Similar to recycling, a rearchive operation that is used to achieve migration of a set of files from one piece of media to another can be carried out without interfering with SRB's control of the SAM-QFS system. As the batch of file-level rearchive operations propagate through the archive system, the event log mechanism will update the copy locations in the MCAT Database.

Furthermore, with the MCAT Database holding archive copy locations, a relational query instead of a linear search with 'sfind' can turn up all files located on a particular piece of media, making migrations easier.

7.8.3 Kerberos Upgrades/Patches

As HPCMP Kerberos is embedded into SRB, there is typically no need to upgrade the SRB HPCMP Kerberos library separately. New releases of SRB will come with new versions of the HPCMP Kerberos libraries if they are made available.

However, if new HPCMP Kerberos upgrades become available well before a new version of SRB is released, the libraries can be replaced manually using the following steps:

1. On all MCAT Servers, SRB Agents, and SRB Client installations rename the file `$SRB_HOME/lib/64/libSrbGSSDriverKerberos.so` with an old extension in order to keep a backup of the working library.
2. Copy the new library as `$SRB_HOME/lib/64/libSrbGSSDriverKerberos.so`.
3. On all MCAT Servers and SRB Agents, restart the SRB servers using "`srb restart`".

7.8.4 RMAN Backup and Recovery

RMAN backup and recovery is almost identical to other Oracle database and recovery operations. RMAN is the recommended backup utility for the Oracle RAC database. Integrated with the database kernel, Oracle RMAN verifies data blocks during backup and restore operations.

RMAN is integrated with OEM. Access to RMAN backup is available on OEM Maintenance Page.

The RMAN environment consists of the utilities and databases that play a role in backing up your data. As a minimum, the environment for RMAN must include the following:

- The target database. This is the database to be backed up by RMAN (mcatdb1).
- The RMAN client is a command-line-oriented database client, much like SQL*Plus, with its own command syntax. From the RMAN client you can issue RMAN commands and some SQL statements to perform and report on backup and recovery operations.

RMAN maintains metadata about the target database and its backup and recovery operations in the RMAN repository. Among other things, RMAN stores information about its own configuration settings, the target database schema, archived redo logs, and all backup files on disk or tape. RMAN's LIST, REPORT, and SHOW commands display RMAN repository information.

The primary store for RMAN repository data is always the control file of the target database. The CONTROL_FILE_RECORD_KEEP_TIME initialization parameter controls how long backup records are kept in the control file before those records are re-used to hold information about more recent backups.

Another copy of the RMAN repository data can also be saved in the recovery catalog.

Using a recovery catalog preserves RMAN repository information if the control file is lost, making it much easier to restore and recover following the loss of the control file. (A backup control file may not contain complete information about recent available backups.) The recovery catalog can also store a much more extensive history of your backups than the control file, due to limits on the number of control file records.

In addition to RMAN repository records, the recovery catalog can also hold RMAN stored scripts, sequences of RMAN commands for common backup tasks. Centralized storage of scripts in recovery catalog can be more convenient than working with command files.

It is strongly recommended that a Recovery Catalog be created and used in a different machine. If creating the Recovery Catalog database in different machine is not possible, then ensure that the recovery catalog and target databases do not reside on the same disk.

7.8.4.1 Create recovery catalog

Creating recovery catalog is a 3 step process. The recovery catalog is stored in the default tablespace of the recovery catalog schema. SYS cannot be the owner of the recovery catalog.

1) Creating the Recovery Catalog Owner
2) Creating the Recovery Catalog
3) Registering the target database

The size of the recovery catalog schema depends on

- The number of databases monitored by the catalog.
- The rate at which archived redo log generates in the target database
- The number of backups for each target database
- RMAN stored scripts stored in the catalog

Step 1: Creating the Recovery Catalog Owner

Begin by creating a database schema (usually called rman). Assign an appropriate tablespace to it and grant it the recovery_catalog_owner role. The following is an example:

```
$ sqlplus '/ as sysdba'
```

```
SQL> CREATE USER rman IDENTIFIED BY password
     DEFAULT TABLESPACE tools
     TEMPORARY TABLESPACE temp
     QUOTA UNLIMITED ON tools;

SQL> GRANT CONNECT, RECOVERY_CATALOG_OWNER TO rman;
```

Step 2: Creating the Recovery Catalog

Creating the Recovery Catalog is a matter of logging into rman and creating the catalog schema. However, before creating the recovery catalog ensure that the tnsnames.ora entry for the catalog database is in the target server and that the listener is up and running in the catalog database server. You must be able to connect to the catalog database from sqlplus from the target (mcatdb) server. In the example below, "catdb" is the catalog database connection string.

```
$ rman catalog rman/password @catdb
RMAN> CREATE CATALOG;
```

Step 3: Registering the Target Database

After ensuring the recovery catalog database is open, connect RMAN to the target database and recovery catalog database and register the database. Also ensure that your target database is open.

```
$ rman TARGET / CATALOG rman/password @catdb
RMAN> REGISTER DATABASE;
```

RMAN creates rows in the catalog tables to contain information about the catalog database. Copy all the pertinent data from the control file into the catalog, synchronizing the catalog with the control file.

```
$ rman TARGET / CATALOG rman/password @catdb
```

7.8.4.2 Recovery Catalog Backup

The Recovery Catalog database is similar to other databases. As such, a backup of this database needs to be done after any backup of the target database. The backup of the catalog database may be a Physical backup or a Logical backup and can be done using RMAN.

When creating or working with the Recovery Catalog database, these guidelines are recommended:

- Run the recovery catalog database in ARCHIVELOG mode to permit point-in-time recovery when needed.
- Set the retention policy to a REDUNDANCY value greater than 1.
- Do not use another Recovery Catalog as the repository for the backups.
- Configure the control file autobackup feature to ON.

7.8.4.3 Flash Recovery Area

Oracle recommends using the Flash Recovery Area (FRA) to store backup datafiles and archived redo logs. Some features of Oracle database backup and recovery, such as the Oracle Flashback Database and guaranteed restore points, require the use of an FRA.

Planning the Size of the Flash Recovery Area it should be large enough to hold the following:

- A copy of all datafiles
- Incremental backups, as used by your chosen backup strategy
- Online redo logs
- Archived redo logs not yet backed up to tape
- Control files
- Control file autobackups (which include copies of the control file and SPFILE)

The following commands can be included in a script to setup the FLASH RECOVERY AREA:

```
SQL> alter system set db_recovery_file_dest = '+FLASH';
SQL> alter system set DB_RECOVERY_FILE_DEST_SIZE = 6 TB;
SQL> alter database enable block change tracking;
```

7.8.4.4 Backup Strategy

The backup strategy should include the following:

- Weekly full hot database backups on Saturday night.
- Daily Incremental backups.
- Archive log retention period 7 days.

Daily incremental backup settings backup the data that is not backed up since the last backup. They provide the following benefits:

- Reduced disk space usage (smaller backup-pieces generated).
- Somewhat faster backup completion times (although the number of blocks eventually written to the backup-piece are less, Oracle still reads all the blocks to determine if they changed).

Use RMAN with the following commands to enable weekly and incremental backups:

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO
'+CONTROL';

RMAN > Backup database format '+ FLASHBKUP/mcat1/%d_%t_%s_%p' tag
'DB_weekly_backup' plus archivelog format '+DATA2/dbbackup/%d_%U' tag
'Arch_daily_backup' delete input;
RMAN> backup incremental level 0 format '+
FLASHBKUP/mcat1/%d_%t_%s_%p' tag 'DB_daily_incremental'
```

7.8.5 Monitoring Applications

SRB currently provides the ability to monitor the online/ offline status of SRB Servers, file systems, and file system capacities. This monitoring capability is provided by the SRB Java Admin. The following screenshot demonstrates its capabilities:

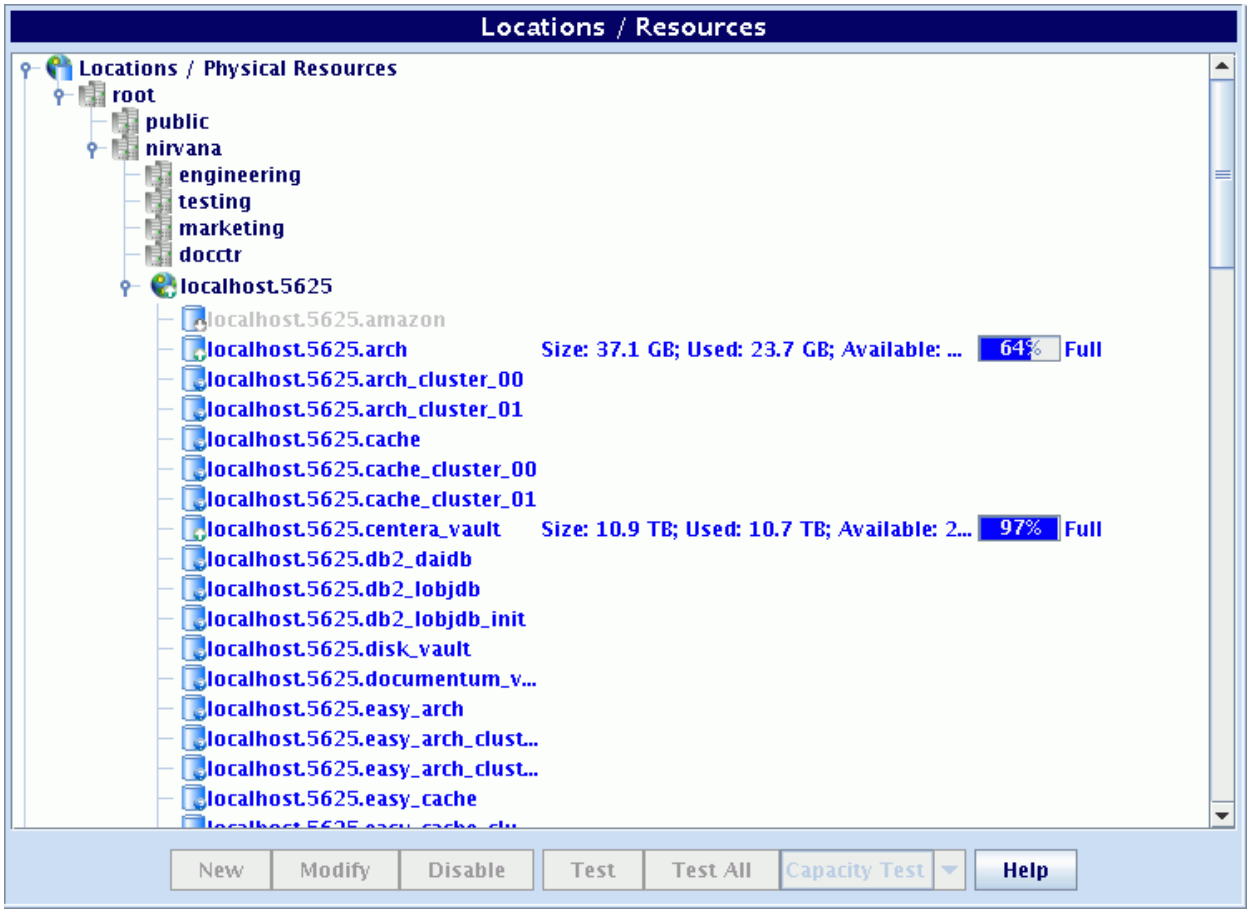


Figure 7-1. SRB Java Admin monitoring of servers and file systems.

Similar capabilities can be achieved using shell commands. For example, the following script tests the connectivity to an SRB Agent, lists its Physical Resources, and prints their capacity:

```
#!/bin/sh

SRB_USER=super@root
SRB_PASSWORD=srb123
SRB_HOST=seawolf
SRB_DOMAIN=arsc.edu

Sinit -user $SRB_USER -host $SRB_HOST.$SRB_DOMAIN -pass $SRB_PASSWORD > /dev/null
RESOURCE_LIST="`SgetR -locn $SRB_HOST@$SRB_DOMAIN`"
RESOURCE_LIST="`echo $RESOURCE_LIST|sed -e 's/resource_name//g' -e 's/-//g'`"

for i in $RESOURCE_LIST
do
  SgetR -sys $i
done
Sexit
```

The output may be as follows:

```
> ./test_resources.sh
PHYSICAL RESOURCE NAME   TOTAL BLOCKS   FREE BLOCKS   AVAILABLE BLOCKS   BLOCK SIZE
AVAILABLE BYTES
arsc.u2                  9725894        3494936       2993653            4096
12262002688

PHYSICAL RESOURCE NAME   TOTAL BLOCKS   FREE BLOCKS   AVAILABLE BLOCKS   BLOCK SIZE
AVAILABLE BYTES
arsc.u3                  11988001       270538        270538             1000000
270538000000
```

All SRB server installations also provide a script, which prints the SRB-related processes on the server to `stdout`:

```
> srb status

USER       PID  PPID S COMMAND
srb      21402     1 S srbServer
srb      25694 21402 S Silm
srb      25695 21402 S Ssync
srb      22482 21402 S srbServer
srb      24552 21402 S srbServer
srb      24964 21402 S srbServer
srb      25621 21402 S srbServer
```

The output of this script shows the SRB Master Server and its child processes. It also shows any Daemons, which are currently running on the server – in this case the ILM and Sync Daemons.

7.8.6 Backup/Restore for User Accidental Removes

If users mistakenly remove files from the archive (i.e., “Srm file1”), the administrator may still have the ability to restore such files from tape. Metadata may still be restored from Oracle database backups. The following procedure describes the steps to perform such a recovery:

1) Look at SRB audit trails (table MCAT_DEAD_AUDITS) to determine the SLM server, SAM-QFS file systems, and physical file path of the mistakenly deleted file. Audit trails can be filtered by user and Collection.

2) If the file's SAM metadata information has been captured in a samfsdump and the SAM recycler has not reclaimed the space where the archive copy resides, the techniques in the SAM-QFS Troubleshooting manual can be used to recover a copy of the file.

3) The Sync Daemon picks-up the mistakenly deleted file and registers it into the MCAT database as a new Data Object. At this point users will have access to their file from SRB.

4) An Oracle database backup can be used to restore a Data Object’s metadata back into the MCAT database. The metadata is uniquely associated with the Data Object by its data_id, which will have changed after step 3).

a) The metadata can only be imported into the MCAT database after disabling the foreign key constraint linking the data_id on the metadata table to the data_id on the MCAT_DATA_INFO table.

b) For each Data Object, rename the data_id for re-imported metadata to the new data_id from step 3).

c) Re-enable the foreign key constraint linking the data_id on the metadata table to the data_id on the MCAT_DATA_INFO table.

7.8.7 Regular Backups for Non-Oracle File Systems

The SRB installation directory contains no files, which are truly irreplaceable and hence need to be backed-up. However, the following list provides the files that should be backed-up:

- `$SRB_HOME/config/mcat.config (MCAT Servers only)`
- `$SRB_HOME/config/license.txt (MCAT Servers only)`
- `$SRB_HOME/config/server.config`

For SAM-QFS the following list provides the files that should be backed-up:

SAM-QFS configuration:

- All files in /etc/opt/SUNWsamfs
- Catalog files
- stk parm files

Log files:

- archive log
- sam-log

- trace logs
- stage logs
- recycler logs
- release logs

Backups:

- samfsdump dumps

All other files can be re-installed using the HPCMP SRB installer kits and all remaining SRB configuration resides inside the MCAT Database.

7.9 Administrative Reports

There are a number of different types of reports that can be generated from SRB. The following report types shall be described: resource utilization, audits, and file expirations. All reports in this section are formatted for terminal output and digestion. Other output formats such as HTML, XML, or other custom formats are supported as well.

Reports in SRB can either be expressed as SQL queries against the MCAT Database or by using a Virtual Collection with a policy, which uses pseudo-SQL query syntax.

For SQL queries against the MCAT Database, it is necessary to create and configure an MCAT Database Resource in SRB:

```
> registerResource MCATDB 'generic dai driver' mcat1@arsc.edu '' database 0
> modifyResource MCATDB changeConfigValue 1::DB_NAME::mcatdb
> modifyResource MCATDB changeConfigValue 1::DB_USER::limited_report_user
> modifyResource MCATDB changeConfigValue 1::DB_AUTH::user123
> modifyResource MCATDB changeConfigValue 1::DB_SCHEMA::srb
```

Once the MCAT Database Resource (in this case named ‘mcatdb’) is created and configured, query objects on the Resource can be registered into the Global Namespace as shown in the subsequent sections.

Virtual Collections do not require an MCAT Database Resource.

7.9.1 Resource Utilization Reports

Resource utilization can be reported on by different criteria – for example, by Resource, by user, or by file type. The following commands register three query objects into the Global Namespace `/reports` Collection. Those three queries perform the reporting on Resource utilization by Resource, user, and file type respectively:

```
> Sregister –rsrc MCATDB –path "<TEMPLATETYPE>VALUES</TEMPLATETYPE>select distinct
MCAT_RSRC_INFO.rsrc_name, sum(MCAT_DATA_REPLICA.data_size) as resource_size,
count(MCAT_DATA_INFO.data_name) as resource_count from MCAT_DATA_INFO,
MCAT_DATA_REPLICA, MCAT_RSRC_INFO where MCAT_DATA_INFO.data_id =
MCAT_DATA_REPLICA.data_id and MCAT_DATA_REPLICA.rsrc_id=MCAT_RSRC_INFO.rsrc_id group
by rsrc_name" /reports/resource.csv
> Sregister –rsrc MCATDB –path "<TEMPLATETYPE>VALUES</TEMPLATETYPE>select distinct
owner_at_domain, sum(last_data_size) as last_data_size, count(data_name) from
mcat_data_object group by owner_at_domain" /reports/user.csv
```

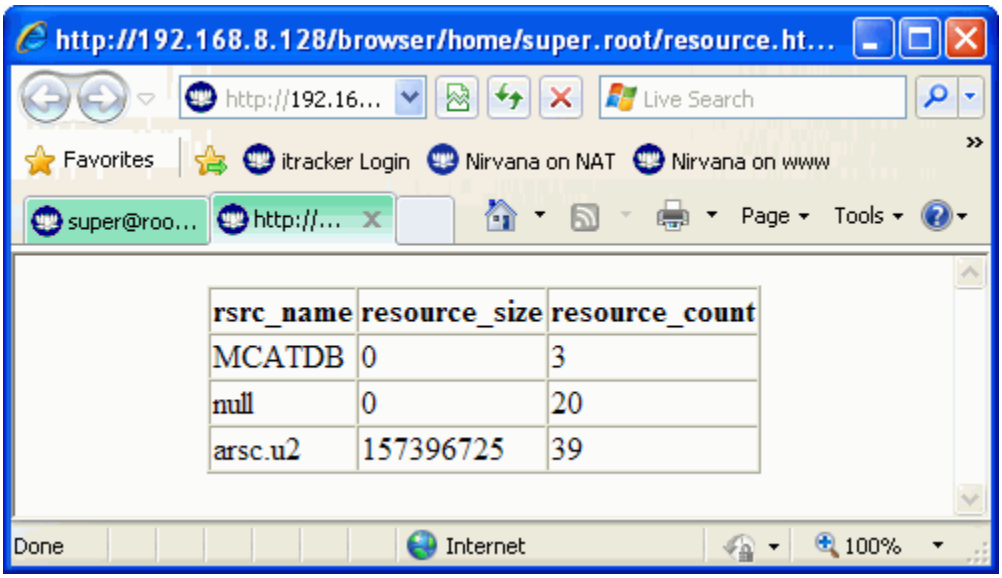
```
> Sregister -rsrc MCATDB -path "<TEMPLATETYPE>VALUES</TEMPLATETYPE>select distinct
data_type_name, sum(data_size), count(data_type_name) from mcat_data_info,
mcat_data_type where mcat_data_info.data_type_id = mcat_data_type.data_type_id group
by data_type_name" /reports/type.csv
```

The output of the resource.csv report may look like this:

```
> Scat /reports/resource.csv
'MCATDB','0','3';
'null','0','20';
'arsc.u2','157396725','39';
```

The 3 query objects show-up under the MCATDB Resource with 0 size. All SRB Objects not associated with a Resource (i.e., Collections) show-up under the '`null`' Resource. The report shows 20 such objects. The '`arsc.u2`' Resource has currently 39 objects with a space usage of 157,396,725 Bytes.

The same report with an HTML template applied would look like this:



rsrc_name	resource_size	resource_count
MCATDB	0	3
null	0	20
arsc.u2	157396725	39

Figure 7-2. Example SRB Resource Utilization Report.

Because the query objects use SQL as a query language, any functionality available in SQL, is also available as part of these reports. Only users with at least 'write' access to the MCAT Database Resource can create reports. Only users with at least 'read' access to the query objects can read reports.

7.9.2 Audit Reports

In a similar fashion, reports can be created on SRB audit trails, which can be filtered by many different criteria such as user, domain, SRB Collection, and time.

The following commands register and print the audit trail for the "`/reports`" Collection and filters out any audit entries performed by users in the "`HPCMP.HPC.MIL`" domain:

```
> Sregister -rsrc MCATDB -path "<TEMPLATETYPE>VALUES</TEMPLATETYPE>select data_name,
AUDITOR_AT_DOMAIN, TIMESTAMP, COMMENTS, ACTION_NAME from mcat_data_audits,
mcat_data_object where collection_name like '/reports/%' and mcat_data_audits.DATA_ID
= mcat_data_object.data_id and auditor_at_domain not like '%@HPCMP.HPC.MIL' order by
TIMESTAMP DESC, AUDITOR_AT_DOMAIN ASC" /reports/reports_audit.csv
> Scat /reports/reports_audit.csv
'reports_audit.csv','super@root','2010-03-08-13.23.42.3373','registered as
"/reports/reports_audit.csv" on "MCATDB"','DataObjectRegister';
'resource.html','super@root','2010-03-08-13.18.26.9030','opened for "read" from
"MCATDB"','DataObjectOpen';
...
```

Besides data auditing, there are also audit trails on users and physical storage resources (i.e., file systems). All three are kept in tables inside the MCAT database – MCAT_DATA_AUDIT, MCAT_USER_AUDIT, and MCAT_RSRC_AUDIT respectively.

As audit table rows can grow rapidly, Team GA recommends that those tables are not replicated between all of the DSRC sites. For cross-site auditing inquiries, queries can always be made to multiple MCAT databases at different DCRCs. Those (up to six) audit trails can then be consolidated into a single view.

SRB auditing does not differentiate between downgrading or upgrading classification levels when it comes to auditing. But when consulting multiple audit entries, auditors can easily determine when file classification was upgraded or downgraded. The following audit entries show the creation, classification, and downgrading events for a file:

```
select * from mcat_data_audits where data_id = 233;
```

```
 data_id |            action_name            | error_name  |        timestamp
|                          comments                          | auditor_name |
auditor_domain | auditor_at_domain
---------+-----------------------------------+-------------+-------------------------
+----------------------------------------------------------------------+--------------+-
---------------+-------------------
     233 | DATA_OBJECT_CHANGE_CLASSIFICATION | SRB_SUCCESS | 2010-05-06-09.33.44.1124
| new classification: "SRB_LEVEL_01"                                   | super        |
root           | super@root
     233 | DATA_OBJECT_CHANGE_CLASSIFICATION | SRB_SUCCESS | 2010-05-06-09.33.39.8047
| new classification: "SRB_LEVEL_02"                                   | super        |
root           | super@root
     233 | DataObjectCreate                  | SRB_SUCCESS | 2010-05-06-09.26.28.3976
| created as "/home/super.root/make_00" on "p193182.5625.disk_vault" | super        |
root           | super@root
(3 rows)
```

Similarly, a query can be constructed to determine failed user logon attempts. For example, the following query lists the failed super@root attempts and shows when the failures occurred. Also, since 'success' messages are logged, more complex queries could be constructed to determine, for example, daily utilization, or peak time usage, as well as other metrics.

```
select action_name, error_name, timestamp, comments
from mcat_user_audits, mcat_user_alias
where mcat_user_audits.user_id = mcat_user_alias.user_id and
mcat_user_alias.user_at_domain_alias = 'super@root' and
error_name = 'SRB_AUTHENTICATION_INVALID'
```

```
order by TIMESTAMP asc;


ACTION_NAME  ERROR_NAME                  TIMESTAMP                COMMENTS
------------ ------------------------- ------------------------ --------------
Authenticate SRB_AUTHENTICATION_INVALID 2010-07-06-19.18.08.9192 Authentication Failed

Authenticate SRB_AUTHENTICATION_INVALID 2010-07-06-19.27.48.7235 Authentication Failed

Authenticate SRB_AUTHENTICATION_INVALID 2010-07-06-19.39.53.3067 Login Failed for
super@root
(3 rows)
```

7.9.3 File Expiration Reports

Another type of report, which was not yet demonstrated, is a Virtual Collection. Virtual Collections look just like a regular directory except that they have a policy associated with them. Upon listing the Virtual Collection, the query is dynamically executed and the query results are displayed as the contents of the Virtual Collection.

For example, the following command registers a Virtual Collection called “expiring_files_user” which contains all the files of a user that are about to expire from the archive:

```
> Smkcoll –policy "((EXPRESSION.create_age > Admin.Retention_Period –
Policy.Warning_Period_User) OR (EXPRESSION.current_timestamp > Admin.Next_Review_Time
– Policy.Warning_Period_User)) AND (Admin.Admin_Hold = 'no') AND (DATA_OBJECT.owner_id
= EXPRESSION.current_user_id) AND (DATA_OBJECT.data_type not like '*collection')"
expiring_files_user
```

A listing of this Collection would show all expiring files that belong to the current user:

```
> Scd expiring_files_user
> Sls
/expiring_files_user
  file1.txt              561
  file2.txt              0
  file3.bin              10000000
  file4.txt              1232
```

A similar Virtual Collection can be created for the PI or S/AAA to show them files of users whom they oversee. This relationship – PI to users whom they oversee – is defined by Collection curatorship of PIs. The following Virtual Collection shows expiring files for a PI if the Collection curatorship is properly setup:

```
> Smkcoll –policy "((EXPRESSION.create_age > Admin.Retention_Period -
Policy.Warning_Period_PI) OR (EXPRESSION.current_timestamp > Admin.Next_Review_Time -
Policy.Warning_Period_PI)) AND (Admin.Admin_Hold = 'no') AND (DATA_OBJECT.data_type
not like '*collection')" expiring_files_pi
```

7.10 Use of Anti-Virus Scanners (or prohibition thereof)

There is no need for AV scanning on the MCAT Servers or MCAT RAC Nodes. The MCAT Servers or MCAT RAC Nodes contain only application binaries and ASM-controlled raw file systems with database files, which are not scanable. User data would not be stored on MCAT Servers and interactive logins are generally disabled. Furthermore, virus scanning on the

database servers would reduce the I/O performance of the database and make the MCAT Servers and MCAT RAC Nodes less responsive.

Due to the low risk of viruses on these types of files and access and the high impact on I/O performance we do not recommend scanning on the MCAT Servers or MCAT RAC Nodes.

7.11 SLM Startup and Shutdown

7.11.1 Normal Startup and Shutdown Procedures

The SLM startup and shutdown is typically performed in an automated fashion. All the components are started independently of each other. However, the following steps describe the manual SLM system startup and shutdown. The order in which Oracle, SRB, and SAM-QFS are started is arbitrary (CRS_HOME refers to Grid Infrastructure home).

1. Start MCAT
   a. Start Oracle
   - Start RAC

     Login as root on participating MCAT RAC Nodes and type the command:
     ```
     # $CRS_HOME /bin/crsctl start crs
     ```
     The command will startup CRS, nodeapps, ASM, database, listener and all valid registered services registered with cluster.
     Check the service status by running the "`crs_stat -t`" command as user 'oracle':
     ```
     $ $CRS_HOME/bin/crs_stat -t
     ```
     The status of all services should show as 'valid'.
     If the cluster service does not start up the registered cluster components then use the following commands to bring up individual components.
     - Start nodeapps individually on each node

       Login as user 'oracle' on the database server.
       Check whether nodeapps is running from each node:
       ```
       $ $CRS_home/bin/srvctl status nodeapps -n <node name>
       ```
       If the status is down, then start it by:
       ```
       $ $CRS_home/bin/srvctl start nodeapps -n <node name>
       ```
     - Start ASM individually on each node

       As user 'oracle' start ASM on each node.
       Check whether ASM is running from each node:
       ```
       $ $CRS_HOME/bin/srvctl status asm -n <node name>
       $ $CRS_HOME/bin/srvctl start asm -n <node name>
       ```
       Check the alert log file to find the status.
     - Start Listener

       Login as user 'oracle' on the database server.
       Check the database listener status by running the following command on each node:
       ```
       $ $CRS_HOME/bin/crs_stat  -t
       ```
       Start the listener using the following command from each node.
       ```
       $ $ORACLE_HOME/bin/srvctl start listener -n <node_name>
       ```
     - Start database

       Login as user 'oracle' on the database server.
       Run the following command only once to start instances on all database nodes:
       ```
       $ $ORACLE_HOME/bin/srvctl start database –d <database name>
       ```
       Check the database status from database alert log file.
   - Start Oracle Streams for a site (optional)

Streams are started automatically as RAC is started. For manual start instructions please refer to the SLM Cookbook section 5.2.

b. Start SRB
Login as user ‘srb’ on each of the MCAT Servers and issue the following command:
```
$ srb start
```
Verify that SRB started successfully. If not, try to restart SRB using:
```
$ srb restart
```
If that does not work, consult the log file `$SRB_HOME/log/srbServer.log` and take appropriate action.
Verify that SRB is running:
```
$ srb status
```
This command should show all SRB MCAT Server processes.

2. SLM Server
a. Start and mount SAM-QFS
1. On the SAM Server, mount all SLM-managed SAM file systems using the 'mount' command:
```
samserver# mount <fs_name_1>
...
samserver# mount <fs_name_n>
```
It is assumed here that all mount options are specified in either the '/etc/vfstab' file or the SAM 'defaults.conf' file.
2. Where centers have QFS clients to SLM-managed file systems, mount the file systems on the clients.
```
client# mount <fs_name_1>
...
client# mount <fs_name_n>
```
b. Start SRB
Login as user ‘srb’ and issue the following command:
```
$ srb start
```
Verify that SRB started successfully. If not, try to restart SRB using:
```
$ srb restart
```
If that does not work, consult the log file `$SRB_HOME/log/srbServer.log` and take appropriate action.
Verify that SRB is running:
```
$ srb status
```
This command should show all SRB Agent Server processes and SRB Daemons.

The following steps describe the SLM system shutdown. These steps should be performed in the given order:

- SLM Server
  - Unmount and stop SAM-QFS
  - Where centers have QFS clients to SLM-managed file systems, unmount the file systems on the clients:
    ```
    client# umount <fs_name_1>
    ...
    client# umount <fs_name_n>
    ```
    To cleanly unmount a file system, it is necessary to stop all processes using files on the file system, including sharing via NFS, if enabled
  - As circumstances allow, allow archiving and staging operations to stop cleanly using the following commands:
    ```
    # samcmd aridle
    # samcmd stidle
    ```

- From the SAM server, unmount all SLM-managed SAM file systems using the 'mount' command

```
samserver# umount <fs_name_1>
...
samserver# umount <fs_name_n>
```

  Similarly, to cleanly unmount a file system, it will be necessary to stop all processes using files on the file system, including sharing via NFS, if enabled.

  - Stop SRB

    Login as user ‘srb’ and issue the following command:

```
$ srb stop
```

    Verify that SRB is stopped by using:

```
$ srb status
```

    This command should not show any SRB Agent Server processes or SRB Daemons.

- Stop MCAT
  - Stop SRB

    Login as user ‘srb’ on each of the RAC nodes and issue the following command:

```
$ srb stop
```

    Verify that SRB is stopped by using:

```
$ srb status
```

    This command should not show any SRB MCAT Server processes.
  - Stop Oracle

1. Stop Oracle Streams for a site (optional)

   Streams are stopped automatically as RAC is stopped. For manual stop instructions please refer to the SLM Cookbook section 5.2.

2. Stop RAC

   Login as user ‘oracle’ on one of the MCAT RAC Nodes.

   Shut down all Oracle RAC instances on all nodes. To shut down all Oracle RAC instances for a database, enter the following command, where db_name is the name of the database. Run this command only once to shutdown the database. Note that ORACLE_HOME is the home directory of the Oracle database server:

```
$ $ORACLE_HOME/bin/srvctl stop database -d <db_name>
```

   Shut down all ASM instances on all nodes. To shut down an ASM instance, enter the following command, where node is the name of the node where the ASM instance is running. Run the following command once on each node:

```
$ $ORACLE_HOME/bin/srvctl stop asm -n <node name>
```

   Stop all node applications on all nodes. To stop node applications running on a node, enter the following command, where node is the name of the node where the applications are running. Run the following command once on each node.

```
$ $ORACLE_HOME/bin/srvctl stop nodeapps -n <node name>
```

   Log in as the ‘root’ user, and shut down the Oracle Clusterware or CRS process by entering the following command on all nodes. Run the following command once on each node. Note that CRS_HOME is the home directory of Clusterware software install directory.

```
# $CRS_HOME/bin/crsctl stop crs
```

   At this point the server can be used for operating system or storage maintenance

7.11.2 Abnormal Shutdown & Recovery Procedures

An abnormal shutdown is assumed to be a situation where the system is terminating unexpectedly and/ or unintentionally.

According to IA Control DCSS-1, *"System State Changes: Ensure that the system initialization, shutdown, and aborts are configured to ensure that the system remains in a secure state"*. Critical data for SRB resides in the MCAT, which is stored on an Oracle database. When an abnormal shutdown occurs, Oracle saves the committed transactions, and the "in-process" transactions are rolled back. During the abnormal shutdown period, Oracle does not permit any logins nor allow any new transactions to be executed.

However, the DBA should comply with the Database STIG as specified by the following requirement within the context of DCSS-1 (Ref: Section 3.1.12, Database STIG):

> *(DG0155: CAT II) The DBA will ensure all applicable DBMS settings are configured to use trusted files, functions, features, or other components during startup, shutdown, aborts, or other unplanned interruptions.*
>
> ***NOTE**: This requirement includes the prevention of scanning for automated job submissions at startup and settings to allow only trusted known good data files at startup.*

What needs to be ensured after any abnormal shutdown is that all systems (i.e., Oracle, SRB, and SAM-QFS) come back to their normal operational state and that the Oracle database is in synch with the SAM-QFS metadata.

Generally, after an abnormal shutdown, DBAs and system administrators should follow the normal startup procedures as described in paragraph 7.11 above. In the case of a discrepancy from the normal procedures, product-specific steps need to be taken to get the system back to normal operations. This might involve consultation of the product's manuals, training procedures, and opening of support cases with the respective vendor.

In order to synchronize Oracle with SAM-QFS, the following steps need to be strictly observed:

1. Ensure the issue that caused the sam-fsalogd failure is resolved and that the sam-fsalogd daemon is running and remains running.
2. Run the Sync Daemon in walk mode, which will synchronize the file system with the MCAT database. Any events occurring while running the Sync Daemon in walk mode will be logged by the sam-fsalogd daemon, but not applied to the MCAT database.
3. Once the walk mode is finished, run the Sync Daemon in real-time mode. This will apply the changes in the sam-fsalogd daemon logs to the MCAT database.

The following items should be verified in order to ensure that the system has been recovered:

1. Verify that all MCAT Servers are operational by using "srb status" and a subsequent "Sshell" on all MCAT Server nodes. The output of "srb status" should show one master srbServer process and at least one child srbServer process. The "Sshell" command should succeed with a welcome message.

2. Verify that all SRB Agents are operational by using "srb status" and a subsequent "Sinit –host <agent server hostname>" on all SRB Agent nodes. The output of "srb status" should show one srbServer master process, at least one srbServer child process, and processes for Silm and Ssync – the ILM and Sync Daemons. The "Sinit –host <agent server hostname>" command should succeed with a welcome message.
3. Both Daemons should be reporting that they are either sleeping or executing a policy. This can be verified by tail'ing the Silm.log and Ssync.log files under $SRB_HOME/log on the SRB Agents. For the Ssync.log in real-time mode the reported latency should approach 0.
4. Optionally, the Sync Daemon can be run in Check Mode in order to verify that MCAT is consistent with the SAM-QFS file systems.
5. Verify that Replication is running without errors by looking at the state of the various streams and by observing the error queues. Also, verify that replication has caught-up synchronizing with the other sites by looking at the Stream latencies. Both of these tasks can be performed as user ‘SYS’ or ‘repadmin’ either via sqlplus command line or the Oracle Enterprise Manager.

7.12 Adding to and Removing Sites (in Replicated Mode)

The size of the SRB Federation in Replicated Mode can be changed by adding the SLM solution to or removing the SLM solution from a site. The Oracle Streams Replication Administrator's Guide (chapter 15, Best Practices for Capture) recommends that daily builds (DBMS_CAPTURE_ADM.BUILD) are made on a periodic basis at a time of minimal load on the environments. This makes the addition of sites easier. There are several potential scenarios depending on the action desired:

1. Adding a new site.
2. Temporarily removing a site (archive logs available).
3. Reconnecting a temporarily removed site.
4. Extended Period site removal (archive logs unavailable).
5. Reconnecting an Extended Period removed site.
6. Permanent site removal (discontinue SLM support).
7. Reconnecting a permanently removed site.

The following sections outline these procedures for adding and removing sites to/ from the SLM Global Namespace.

7.12.1 Adding a New Site

Adding a new site requires the addition of the MCAT Servers, MCAT RAC Nodes, and MCAT database; and the integration of the MCAT Servers into the site's environment. An important step in adding a new site is to ensure that the new site will have an initial database image based on a point in time where all existing databases are in sync.

The following steps are designed to minimize downtime and ensure data consistency among all sites with no data loss occurring:

1. Install and configure the MCAT Servers and MCAT RAC Nodes with Solaris OS, Oracle RAC database, and Nirvana SRB ILM software (detailed steps are identified in the SLM Cookbook).
2. From one of the existing sites, add the new site's SRB Locations (for MCAT Servers and SRB Agents), Resources (for all SLM-managed SAM-QFS file systems), Sync Daemons, Users, and Collections to the Global Namespace.
3. Make sure that the new site's database is clean (no SRB data or streams setup existing).
4. Stop MCAT Server processes on all sites ('srb stop').
5. Allow all databases to finish replication and get in sync (this will take some time to finish depending on the latency).
6. Stop all Streams (see Cookbook section 5.2).
7. Run DBMS_CAPTURE_ADM.BUILD on all sites and note the System Change Number (SCN) for all sites. This is important in order to setup Streams for the new site.
8. Resume operations on all existing sites (i.e., start Streams, 'srb start').
9. Take a Data Pump export on one site consistent to that site's SCN that was noted in step 5.
10. Import Dump at new site.
11. Setup and start Streams between the new site and all existing sites using individual SCNs noted in step 7 (see Cookbook section 5.1).

12. Initialize MCAT Servers and SRB Agents at the new site. This is described in sections 5.3.5 (SRB MCAT Server Configuration) and 5.2.2.1 (SRB Agents) respectively.
13. Start SRB and SAM-QFS and synchronize SAM with SRB by following the steps in sections 7.11 (SLM Startup and Shutdown).

7.12.2 Temporary Site Removal

A temporary planned or unplanned site disconnect does not require any manual intervention with Oracle Streams. As long as archive logs from the other source sites are available, Streams will automatically recover upon re-connect and re-synchronize the disconnected site. However, if it is desired one can manually stop Streams replication (see Cookbook section 5.2).

In both cases the archive logs at all sites will be used to store transactions for later re-sync. Hence, the storage capacity of all the sites needs to be carefully examined and monitored. By default the archive logs at all databases are resident on the on-line storage array for seven days. Logs older than seven days are kept in the backups and can be restored if necessary. It is important to ensure there is enough on-line storage capacity if it becomes necessary to restore older logs.

7.12.3 Re-connect of Temporarily Removed Site

As long as archive logs from the other source sites are available, Streams will automatically recover upon re-connect and re-synchronize the disconnected site. However, if Streams were stopped manually, they need to be restarted (see Cookbook section 5.2).

Conflict resolution will be handled as always but may require a larger amount of manual intervention than usual. If the MCAT is thought to be out of sync with the SAM-QFS metadata, follow the steps identified in section 7.11.2 Abnormal Shutdown & Recovery Procedures.

7.12.4 Extended Period Site Removal (Archive Logs Unavailable)

Extended period removal is defined as an instance where a site was removed from the SRB Federation for a period of time that is greater than the program's Archive Log Retention Policy. The steps to remove a site for an extended period are as follows:

1) Stop Oracle Streams at the permanently removed site.
2) Remove Oracle Streams replication with the permanently removed site.
3) Shutdown SRB (with Sync Daemon, SRB Agents, and MCAT Servers).

7.12.5 Re-connect of Extended Period Site Removal

In the case where an Extended Period Removal Site needs to be reconnected, the site is treated as a new site. Follow the steps under section 7.12.1, Adding a New Site.

7.12.6 Permanent Site Removal (SLM metadata support for site discontinued)

During a permanent site removal, the addition of new metadata is no longer allowed as SRB will not be operational at that site. At the end of this procedure, only local SAM-QFS file system access will remain operational.

Permanent removal is defined as an instance where a site was removed from the SRB Federation for an extended period of time that is greater than the program's Archive Log Retention Policy.

1. Stop Oracle Streams at the permanently removed site.
2. Remove Oracle Streams replication with the permanently removed site.
3. Shutdown SRB (with Sync Daemon, SRB Agents, and MCAT Servers) and uninstall SRB from MCAT Servers and Agents; and remove Oracle RAC from MCAT RAC Nodes.
4. Remove all objects (Files, Directories, Metadata Schemes and attributes, Users, Locations, and Resources) that either physically resided on the permanently removed site or are no longer needed. The drop_site.sql script can be used for that purpose.
5. Save backup of MCAT database at removed site so that metadata may be restored at a later time if site is to be re-connected.

7.12.7 Re-connect of Permanently Removed Site

In the case where a permanently removed site needs to be reconnected, the site must essentially be treated as a new site – but with some deviations. In order to restore valuable information that was previously stored as metadata, the database metadata that was backed-up as part of the site removal procedures may be partially re-associated.

Follow the steps under section 7.12.1, Adding a New Site. Then an attempt can be made to re-associate backed-up file and directory metadata with the new database. This may be a very elaborate process where file and directory names in the metadata backup are attempted to be matched-up with the file and directory names in the new database.

7.13 Upgrade Strategies

This section covers upgrades for all software components that are involved in SLM.

7.13.1 SRB Upgrades

SRB generally distinguishes between major, minor, and maintenance upgrades. For example, in SRB v3.2.05 the major version is 3, the minor version is 2, and the maintenance version is 5. Upgrade procedures for SRB vary based on the type of upgrade and the following sections will show upgrade procedures for major, minor, and maintenance upgrades.

Table 7-1. SRB Upgrade Types

Upgrade Type	Frequency (Mo.)	Duration (est. hrs.)
Major	18-24	12
Minor	6-12	6
Maintenance	1-12	6

Note that the duration for Major upgrades will vary with the number of files managed by MCAT.

7.13.1.1 SRB Major Upgrades

Major SRB upgrades may involve an MCAT Database schema change, SRB protocol changes, and new features. Major releases require an MCAT Database schema update, MCAT Server,

SRB Agent (with Daemon) and SRB Client software updates, and updates to customer or SRB open source applications such as the Preload Library. All components need to be upgraded at once.

MCAT Servers will not accept incoming connections unless they find an MCAT Database schema version number that matches their release version. The following steps apply to all sites for a Major SRB version upgrade:

3. Stop services and applications
   a. (Replicated Mode only) Stop Oracle Streams replication.
   b. Stop custom and open source applications.
   c. Stop SRB Agents.
   d. Stop MCAT Servers.
4. Upgrade MCAT
   a. Backup MCAT Database using Oracle utilities (i.e., rman, exp).
   b. Install new MCAT Server software.
   c. Update MCAT schema using mcatUpdate Mcommand.
      1. … at each site, or
      2. … at one master site.
6. Upgrade SRB Agents
   a. Install new SRB Agent software.
7. Recompile and redeploy customer and SRB open source applications.
8. (Replicated Mode only) Once all sites are updated, restart the Oracle Streams replication.
   - …in the "each site mcatUpdate case" by performing an Oracle Streams build at each site and then restarting Streams skipping to the new SCNs.
   - …in the "one master site mcatUpdate case" by starting Streams and waiting for all sites to catch-up with the master site.

- Restart SRB
  - Initialize and restart MCAT Servers
  - Initialize and restart SRB Agents, which automatically restarts Daemons.

Different major versions are not interoperable. Hence, downtime needs to be scheduled during the upgrade process. In Replicated Mode, sites must be upgraded simultaneously as otherwise the Oracle Streams replication fails once it is restarted.

7.13.1.2 SRB Minor Upgrades

Minor SRB upgrades may involve SRB protocol changes, and new features. Minor releases require MCAT Server, SRB Agent (with Daemon) and SRB Client software updates, and updates to customer or SRB open source applications such as the Preload Library. The MCAT Database schema remains unchanged. All components need to be upgraded at once. The following steps apply to all sites for a Minor SRB version upgrade:

1. Stop services and applications
1. Stop custom and open source applications.
2. Stop SRB Agents.
3. Stop MCAT Servers.

2. Upgrade MCAT
    1. Install new MCAT Server software.
    2. Initialize and restart MCAT Servers.
3. Upgrade SRB Agents
    1. Install new SRB Agent software.
    2. Initialize and restart SRB Agents, which automatically restarts Daemons.

Recompile and redeploy customer and SRB open source applications.

Different minor versions of the same major release are not interoperable. Hence, downtime needs to be scheduled during the upgrade process. Although sites can theoretically be upgraded independently, inter-site data transfers will only work if the same minor versions exist between sites.

7.13.1.3 SRB Maintenance Upgrades

Maintenance SRB upgrades may involve new features but make no changes to the MCAT database schema or the SRB protocol. This permits rolling upgrades of different SRB components. Maintenance releases usually require MCAT Server, SRB Agent (with SRB Daemons) and SRB Client software updates, and updates to customer or SRB open source applications such as the Preload Library.

The Maintenance SRB versions upgrade process is identical to the Minor SRB version upgrade process with the exception that components may be upgraded at different times. Different maintenance versions of the same minor release *can* interoperate. This allows for the SRB Federation to continue operating without any downtime. This is possible due to the usage of multiple MCAT Servers.

7.13.2 SAM-QFS / ACSLS

7.13.2.1 SAM-QFS Upgrades

SAM-QFS upgrades, whether major or minor revisions, are delivered as new packages. Subversions are typically delivered as patches. The estimated frequency and duration of these upgrades is as shown in the following table:

Table 7-2. SAM_QFS Upgrade Types

Upgrade Type	Frequency (Mo.)	Duration (est. hrs.)
Major	>60	1-12
Minor	4-12	1
Maintenance	1-12	1

All upgrades to SAM-QFS require the following sequence of steps:

- Stop any applications that use the file systems, including the SRB Agent.
- Stop SAM operations.
- Unmount the file systems.

- Apply the software update.
- Remount the file systems.
- Restart SAM operations.
- Restart applications that use the file systems, including the SRB Agent.

Prior to version 5 of the software, all clients and servers that accessed the same file system were required to run the same subversion of the software and therefore, the file system had to be unmounted on every node prior to software upgrade. Once all nodes are upgraded to version 5, nodes must still unmount their file systems during software upgrades, but SAM-QFS version 5 accommodates upgrading each node one at a time.

While these rules generally apply, version release notes or the patch installation instructions should always be checked to ensure that there is no variation from these standard procedures.

Additionally, where large step upgrades (three minor versions or further) of SAM-QFS are being contemplated, consult with Oracle support prior to proceeding.

The disk-based file system component of SAM-QFS is qualified with any disk supported by the underlying Solaris operating system. The SAM archive management component does explicitly qualify tape and library hardware as well as support applications such as ACSLS. Upgrades to SAM-QFS, its managed hardware, or ACSLS should factor in this qualification matrix before being undertaken.

Additionally, proper functioning of SRB-based SLM solution will require qualification of new SAM versions as they are released. Therefore, General Atomics should be consulted prior to SAM-QFS upgrades.

7.13.2.2 ACSLS Upgrades

In many environments, SAM directs tape library operations via the ACSLS software.

At the core of ACSLS is a database. Upgrades of ACSLS roughly proceed by exporting the database, stopping and removing the old software, installing and starting the new software, configuring the new software to recognize the tape library (or libraries) and then re-importing the database. As such, ACSLS upgrades do require downtime and therefore, will require SAM media operations to halt. Operations on the SAM-QFS file system that do not depend on ACSLS may proceed during ACSLS upgrade. On the other hand, the resilience of applications that may experience long delays accessing offline files while ACSLS is being upgraded should be considered.

New versions of ACSLS will, at a minimum, be qualified with all previously qualified tape libraries that are still supported by Oracle. As with SAM-QFS, the release notes and installation guide for new versions of ACSLS should be reviewed thoroughly for exceptional procedures. The estimated frequency and duration of these upgrades is as shown in the following table:

Table 7-3. ACSLS Upgrade Types

Upgrade Type	Frequency (Mo.)	Duration (est. hrs.)
Major	>24	1
Minor	9-12	1
Maintenance	1-12	1

7.13.3 Oracle Upgrades

Oracle, as the database of choice for the MCAT database, will need to be upgraded as new patches or releases become available according to each site's maintenance schedule. The update frequency and duration for Oracle upgrades is reflected in the following table.

Table 7-4. Oracle Upgrade Types

Upgrade Type	Frequency (Mo.)	Duration (est. hrs.)
Major	18-24	2-3 (shared $ORACLE_HOME) 9 (separate $ORACLE_HOME)
Minor	6-12	1 (shared $ORACLE_HOME) 9 (separate $ORACLE_HOME)
Maintenance	1-12	0.5 (shared $ORACLE_HOME) 2 (separate $ORACLE_HOME)

The following paragraphs will elaborate on the upgrade process.

Rolling upgrade is available only for patches that have been documented in patch readme file. Typically, patches that can be installed in a rolling upgrade include:

- Patches that do not affect system dictionary.
- Patches not related to cluster inter-node communication.
- Patches related to client-side tools such as SQL*PLUS, Oracle utilities, development libraries, and Oracle Net.
- Patches that do not change shared database resources such as datafile headers, control files, and common header definitions of kernel modules.
- Rolling upgrade of patches is currently available for one-off patches only. It is not available for patch sets.

Since Oracle software is installed on each node, all rolling upgrade patches will be able to apply in HPCMP RAC environment. Note that all rolling upgrade patch must be applied to all nodes in 24 hours.

7.13.4 OS Upgrades/Patches

When considering upgrades or patches to the Operating System, the SLM software suite must be recertified to run on these versions. The administrator must carefully evaluate the compatibility of the major software components within the suite and with new versions of the Operating System.

8 METADATA SCHEMES

Metadata Schemes in SRB allow for the association of custom system- and user-level metadata attributes with Data Objects or Collections. System Master Schemes are automatically populated for every Data Object or Collection whereas User Master Schemes are filled-in by the end users.

At the time when metadata attributes are populated, there is a certain algorithm, which determines the final attribute value. This algorithm is shown in the following table:

Table 8-1. Attribute Population Algorithm

Step	Description
1.	If the attribute is a string list then the value must be one of the strings specified for the string list.
2.	The system uses values supplied directly by the user (does not apply for System Master Schemes).
3.	If the user does not supply a value, i.e. the value is “null”, then the system uses the user default value associated with the attribute column.
4.	If there is no user default value, the system inherits the value from the Parent Collection.
5.	If there is no Parent Collection value, the system uses the administrator default value associated with the attribute column.
6.	If none of the above conditions are met then the system leaves the value as “null”.

When adding an attribute (i.e. column) to a User Master Scheme, the attribute values for existing Data Objects and Collections are NULL. Consequently, the attribute at the local DSRC and remote DSRCs need to be manually inserted based on the attribute’s default value.

8.1 Policy Table

A policy table will be used for storing administrator configurable attributes that can be different at each site.

Table 8-2. Policy Attributes.

Attribute	Default Value	Description
Warning_Period_User	28	Number of days before file gets deleted that user will receive a warning.
Warning_Period_PI	15	Number of days before file gets deleted that PI will receive a warning.
Review_Period	1096	Number of days before next review is required.
Max_Name_Value_Rows	20	Maximum number of rows per object in the Name Value Scheme.
Max_Retention_Period	30000	Maximum number of days an object can be retained in the archive.
Group_Soft_Quota	1000000000	Soft Quota in Bytes not to be exceeded by any UNIX group name.

Group_Hard_Quota	1500000000	Hard Quota in Bytes not to be exceeded by any UNIX group name.
Project_Soft_Quota	10000000000	Soft Quota in Bytes not to be exceeded by any HSM Project ID.
Project_Hard_Quota	15000000000	Hard Quota in Bytes not to be exceeded by any HSM Project ID.

8.2 Admin Scheme

The Admin Scheme is a system scheme, which means that all the attributes are automatically applied to every file. The Admin Scheme contains attributes to support the following desired archive behaviors:

Table 8-3. User Requested Behavior

1) Don't set retention date and file disappears after 30 days (archive copy is created, no DR) – this is the default behavior if no attributes are set by the user.
2) Set retention days and system will keep file and ask user about status periodically. With regards to purge and DR behavior, there are two options:
2a) Archive and keep on cache (not guaranteed if space is needed). - DR - no DR
2b) Archive and release from cache immediately after archive. - DR - no DR

The Admin Scheme contains the following attributes:

Table 8-4. Admin Scheme Attributes

Attribute Name	Attribute Value	Description
Archive_Behavior	yes (default), no	Is the object to be archived? User modifiable.
DR_Behavior	yes, no (default)	Is a DR copy to be made for the object? User modifiable.
Purge_Behavior	yes, no (default)	Is the object immediately released after archival? User modifiable.
Retention_Period	integer (default), 30	Identifies the number of days the object is retained in the archive. User modifiable.
Retain_Time	= object creation time + Retention Period	Date-time until which object will be retained.
Last_Review_Time	= attribute value modification date	Current date-time automatically updated on scheme modify.
Next_Review_Time	= least(Last Review + Review Period, Retain_Time)	Next date-time when archive policy of the file is to be reviewed by the user.
Admin_Hold	yes, no (default)	Is the object prevented from expiration and purge?
Warning_Note	yes (default), no	Does the user receive a notification at the Warning_Period_User before file expiration? User modifiable.
HPCMP_Project_ID	string[16]	13 character HPCMP project code.

8.3 HSM Schemes

There are two HSM Schemes – one called “HSM” for storing HSM attributes, which are identical for all HSM copies; one called “HSM_Copy” for storing HSM attributes, which are different among HSM copies. Both HSM Schemes are User Master Schemes. They should not be System Master Schemes because their values are not available right away upon object registration into SRB. Also, the attributes serve mostly informational purposes. The two Schemes contain the following attributes:

Table 8-5. HSM Scheme Attributes.

Attribute	Description	Variable (struct sam_stat)
Inode	File serial number.	uint_t st_ino
Gen	File generation number.	unit_t gen
Residence_Timestamp	Timestamp HSM residency changed last.	time_t residence_time
Attribute_Timestamp	Timestamp attributes changed last.	time_t attribute_time
Checksum	128bit checksum value.	u_ll_t csum_val[2]
Checksum_Algorithm	Checksum algorithm used.	uchar_t cs_algo
HSM_Project_ID	Project Identification	projid_t projid

Table 8-6. HSM Copy Scheme Attributes.

Attribute	Description	Variable (struct sam_copy_s)
Copy_Number	HSM copy number (1-4).	<index>
Create_Timestamp	Timestamp archive copy created.	time_t creation_time
VSN_Count	Number of VSNs used.	short n_vsns
VSN	Volume Serial Number.	char vsn[32]
Media_Type	Media Type.	char media[4]
Position	Position of archive file on media.	u_ll_t position
File_Offset	Location in archive file (in units of 512 Bytes).	uint_t offset

Note that the “HSM_Copy" Scheme is a multi-row scheme, which can store multiple rows (or attributes for multiple HSM copies) per Data Object

The remaining HSM attributes – not listed in the tables above but referenced in section 6.3 – are stored in the MCAT_DATA_INFO and MCAT_DATA_REPLICA tables, which contain the SRB system attributes.

The following script creates the HSM Schemes and their attributes:

```
schemeName="HSM"
registerScheme "$schemeName"

registerColumn -default_to_null "Inode"               "$schemeName" long 5
registerColumn -default_to_null "Gen"                 "$schemeName" long 5
registerColumn -default_to_null "Residence_Timestamp" "$schemeName" timestamp 30
registerColumn -default_to_null "Attribute_Timestamp" "$schemeName" timestamp 30
registerColumn -default_to_null "Checksum"            "$schemeName" string[64] 40
registerColumn -default_to_null "Checksum_Algorithm"  "$schemeName" string[8] 1
registerColumn -default_to_null "HSM_Project_ID"      "$schemeName" integer 10

schemeName="HSM_Copy"
registerScheme -multiple "$schemeName"

registerColumn -default_to_null "Copy_Number"         "$schemeName" integer 1
registerColumn -default_to_null "Create_Timestamp"    "$schemeName" timestamp 30
registerColumn -default_to_null "VSN_Count"           "$schemeName" integer 1
registerColumn -default_to_null "VSN"                 "$schemeName" string[32] 8
registerColumn -default_to_null "Media_Type"          "$schemeName" string[8] 4
registerColumn -default_to_null "Position"            "$schemeName" long 10
registerColumn -default_to_null "File_Offset"         "$schemeName" integer 5
```

8.4 Users Scheme

The Users Scheme allows users to associate large volumes of metadata with their object. It can be extended at a later time.

Table 8-7. User-Defined Metadata Attributes.

Attribute	Description
XML (CLOB)	User-defined XML formatted string of metadata
Source (string[256])	Link to related Information file
Relation (data_id)	Link to related Information file

8.5 Dublin_Core Scheme

The Dublin_Core scheme is an optional set of attributes that users can associate with their files or directories.

Table 8-8. Dublin_Core Scheme Attributes.

Type	Attribute	Description
System	Creator	Document creator.
	Create_Date	Date document was created (default: current time).
	Type	Type of document.
	Contributor	Entity providing contributions to the document.
	Document_ID	Document Identifier – creator provided.
	Publisher	Person, organization, or service. Used to indicate the entity.
User	Description	Description of the document.
	Subject	Document subject (keywords, key phrases or classification codes).
	Title	Document Title.
	Rights	Statement about property rights.

8.6 Name_Value Scheme

The Name_Value Scheme called "Name_Value" allows for an alternate, flexible, metadata entry mechanism to the XML attribute in the User Scheme. The Name_Value Scheme permits multiple rows of metadata attributes per Data Object or Collection (i.e., per data_id). Multi-row schemes in SRB automatically maintain a row_id sequence number to keep track of the rows.

Table 8-9. Name Value Scheme Attributes.

Attribute	Description
Name (string[16])	User-specified attribute name.
Value (string[256])	User-specified attribute value.

The following script creates the User Scheme and its attributes:

```
schemeName="Name_Value"
registerScheme -multiple "$schemeName"

registerColumn -default_to_null "Name"     "$schemeName" string[16] 16
registerColumn -default_to_null "Value"    "$schemeName" string[256] 30
```

The following trigger can be used to enforce that only a maximum number of rows (as specified in the Policy Table) is associated with each object:

```
CREATE OR REPLACE TRIGGER max_name_value_rows_trigger
BEFORE INSERT OR UPDATE ON SRB_NAME_VALUE
FOR EACH ROW
DECLARE
       m integer;
BEGIN
       select max_name_value_rows into m from SRB_T_POLICY;

       if (:new.row_id >= m) then
              raise_application_error(-20000, 'Maximum number of rows per object (' ||
m || ') was exceeded.', true);
       end if;
```

```
END;
```

8.7 System-Level Metadata

The following are examples of system-level metadata attributes, which are generally stored in the MCAT_DATA_INFO and MCAT_DATA_REPLICA tables:

Table 8-10. System-Defined Metadata Attributes.

Attribute	Description
Name	Name of the object.
Timestamps	The date and time when the Data Object or Collection was created in the SRB Federation.
Size	Size of the object.
File system path	Path to the object.
Access count	The number of times a Replica was accessed since its creation.

9 DATA MANAGEMENT POLICIES

The SLM solution provides a flexible way of specifying and automatically executing data management policies. A set of relevant parameters (and an example of each) that are used to specify policies is shown in the following table.

Table 9-1. Examples of Relevant Policy Parameters

Parameter Name	Parameter Value
Actions	Backup, Delete, Migrate, Replicate, Synchronize, Archive, Stage, Purge
Modes	List (generates report on all files that would be affected by all policies); Normal (executes the specified policies)
Policies	(EXPRESSION.modify_age > **30**); ((DATA_OBJECT.collection_name = '**/afrl**') OR (DATA_OBJECT.collection_name LIKE '**/afrl/***')) ; ((EXPRESSION.access_age > **7**) AND (EXPRESSION.replica_count > **1**) AND (DATA_RESOURCE.resource_class = '**archival**')); (Admin.Retention_Period > EXPRESSION.modify_age)
Recurrence	Frequency, Interval, Until, Count, Time, day, weekday, other restrictions

The subsequent sections provide example policies which implement the desired archive behaviors in this solution. All policy examples assume they are run at ARL on a SAM-QFS file system Resource called arl.msas8.samfs1. Also, the SAM-FS copy 1 is assumed to be a local archive copy and copy 4 is assumed to be DR.

For any of the policies to run, an ILM Daemon needs to be deployed, configured, and enabled. The following script deploys an ILM Daemon to the ARL `msas8` server as the super@root user and configures it to be ready to receive policies. The `addPolicy()` function will be used by the scripts in later sections:

```
addPolicy()
{
  modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser attachArrayRow
```

```
      "$daemonName:POLICY::$daemonPolicy"
  modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
      "$daemonName:POLICY::$daemonPolicy::POLICY::$daemonPolicyString"
  modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
      "$daemonName:POLICY::$daemonPolicy::MODE::$daemonMode"
  modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
      "$daemonName:POLICY::$daemonPolicy::ACTION::$daemonAction"
  modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
      "$daemonName:POLICY::$daemonPolicy::SOURCE_RESOURCE::$daemonSourceResource"
  modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
      "$daemonName:POLICY::$daemonPolicy::COMMAND::$daemonCommand"
  modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
      "$daemonName:POLICY::$daemonPolicy::RECURRENCE::$daemonRecurrence"
}


daemonName="SRB executables:Scommands:Silm"
daemonUser=super@root
daemonDeployLocation=ARL.msas8

modifyLocation $daemonDeployLocation attachExecutable "$daemonName::$daemonUser"

modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
      "$daemonName::AUTH_SCHEME::KERBEROS_AUTH"
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser changeConfigValue
      "$daemonName::TIME_ZONE::America/New_York"
```

Once the ILM Daemon is configured with all policies (see next sections), the following command (at the end of the previous script after all policies are configured as described in the next sections) will enable the Daemon on the server where it was deployed:

```
modifyExecutable "$daemonName" $daemonDeployLocation $daemonUser enable
```

The following sections will describe how to use Acommands to configure the ILM Daemon with the data management policies intended for HPCMP operations. Besides the Acommands, the Java Admin GUI can be used to accomplish the same.

9.1 Archival

The Archival policy initiates local archival of files.

Table 9-2. Archival

Condition:	Only if Retention_Period and Archive_Behavior are set, create local archive copy
POLICY:	(Admin.Retention_Period IS NOT NULL) AND (Admin.Archive_Behavior = 'yes') AND (DATA_REPLICAS.replica_status & 'SRB_STAT_ARCHIVED_01 | SRB_STAT_ARCHIVING | SRB_STAT_ARCHIVE_NEVER' = ' SRB_STAT_NONE ') AND (DATA_OBJECT.data_type NOT LIKE '*collection')
ACTION:	Archive
SOURCE_ RESOURCE:	arl.msas8.samfs1
COMMAND:	SRB_CMD_COPY_01

The following script (in combination with the script in section 9) creates this archival policy in SRB:

```
daemonPolicy=ARCHIVE
```

```
daemonPolicyString=(Admin.Retention_Period IS NOT NULL) AND
(Admin.Archive_Behavior = 'yes') AND
(DATA_REPLICAS.replica_status   &   'SRB_STAT_ARCHIVED_01   |   SRB_STAT_ARCHIVING   |
SRB_STAT_ARCHIVE_NEVER' = 'SRB_STAT_NONE')
daemonMode=Normal Mode
daemonAction=Archive
daemonSourceResource=arl.msas8.samfs1
daemonCommand=SRB_CMD_COPY_01
daemonRecurrence=FREQ=MINUTELY;INTERVAL=60
addPolicy
```

9.2 DR

The DR policy initiates archival of files to the DR site.

Table 9-3. DR

Condition:	Only if DR_Behavior is set, create a DR copy
POLICY:	(Admin.Retention_Period IS NOT NULL) AND (Admin.DR_Behavior = 'yes') AND (DATA_REPLICAS.replica_status & 'SRB_STAT_ARCHIVED_04 | SRB_STAT_ARCHIVING | SRB_STAT_ARCHIVE_NEVER' = 'SRB_STAT_NONE') AND (DATA_OBJECT.data_type NOT LIKE '*collection')
ACTION:	Archive
SOURCE_ RESOURCE:	arl.msas8.samfs1
COMMAND:	SRB_CMD_COPY_04

The following script (in combination with the script in section 9) creates this archival policy in SRB:

```
daemonPolicy=DR
daemonPolicyString=(Admin.Retention_Period IS NOT NULL) AND
(Admin.DR_Behavior = 'yes') AND
(DATA_REPLICAS.replica_status   &   'SRB_STAT_ARCHIVED_04   |   SRB_STAT_ARCHIVING   |
SRB_STAT_ARCHIVE_NEVER' = 'SRB_STAT_NONE')
daemonMode=Normal Mode
daemonAction=Archive
daemonSourceResource=arl.msas8.samfs1
daemonCommand=SRB_CMD_COPY_04
daemonRecurrence=FREQ=MINUTELY;INTERVAL=60
addPolicy
```

9.3 Release/ Purge

The SAM release policies will remain in effect. However, release attributes such as "release after" could be set using a policy such as the following:

Table 9-4. Release/Purge

Condition:	Only if Purge_Behavior is set, set "relase after" attribute
POLICY:	(Admin.Purge_Behavior = 'yes') AND (DATA_REPLICA.replica_status & 'SRB_STAT_RELEASE_AFTER' = 'SRB_STAT_NONE') AND (DATA_OBJECT.data_type NOT LIKE '*collection')
ACTION:	Purge
SOURCE_ RESOURCE:	arl.msas8.samfs1
COMMAND:	SRB_CMD_RELEASE_AFTER

The following script (in combination with the script in section 9) creates this release/purge policy in SRB:

```
daemonPolicy=PURGE
daemonPolicyString=(Admin.Purge_Behavior = 'yes') AND
(DATA_REPLICAS.replica_status & 'SRB_STAT_RELEASE_AFTER' = 'SRB_STAT_NONE')
daemonMode=Normal Mode
daemonAction=Purge
daemonSourceResource=arl.msas8.samfs1
daemonCommand=SRB_CMD_RELEASE_AFTER
daemonRecurrence=FREQ=MINUTELY;INTERVAL=60
addPolicy
```

9.4 Expiration

The expiration policy deletes files from archived and cached tiers, including DR. The following are example Expiration policies.

Table 9-5. Expiration

Condition:	Expire after retention period is met
POLICY:	((EXPRESSION.create_age > Admin.Retention_Period) OR (EXPRESSION.current_timestamp > Admin.Next_Review_Time)) AND (Admin.Admin_Hold = 'no') AND (DATA_OBJECT.data_type NOT LIKE '*collection')
ACTION:	Delete
SOURCE_ RESOURCE:	arl.msas8.samfs1
COMMAND:	

Expiration in the SLM solution is controlled by user-level metadata attributes (see section 8.2). Upon reaching the Warning_Period_User, the user will be notified of files expiring from the archive. If the user does not extend the Retention Period and it is reached, the ILM Daemon will remove the files from the SAM-QFS file system. The now expired files become candidates for being overwritten on their tape media.

The following script (in combination with the script in section 9) creates these expiration policies in SRB:

```
daemonPolicy=EXPIRATION
daemonPolicyString=((EXPRESSION.create_age > Admin.Retention_Period) OR
(EXPRESSION.current_timestamp > Admin.Next_Review_Time)) AND
(Admin.Admin_Hold = 'no') AND
(DATA_OBJECT.data_type NOT LIKE '*collection')
daemonMode=Normal Mode
daemonAction=Delete
daemonSourceResource=arl.msas8.samfs1
daemonCommand=
daemonRecurrence=FREQ=MINUTELY;INTERVAL=60
addPolicy
```

9.5 Staging (order on tape, overall size of files being staged, etc...)

Section 6.4.3 describes the stage process. Team GA does not currently anticipate a staging policy. Staging will either be implicit when users request files from SAM-QFS, which are only resident on tape; or it can be initiated by the users using the Sstage Scommand.

9.6 Moving Data between Centers

Team GA does not anticipate a policy for cross-center data movement. Such an operation would be user-driven as shown in section 11.2.8.2.

9.7 Recycling

Section 6.4.4 describes the recycling process. Team GA does not anticipate an ILM-level recycling policy. Recycling will be performed at the HSM level as described in section 7.8.1. Recycling activities are transparent to SRB. SRB will synchronize the HSM scheme and the HSM_Copy scheme on recycling.

10 USER INTERFACES

The SLM solutions provides several user interfaces out-of-the-box. Additional interfaces, such as web portals will be customized to meet HPCMP security requirements.

10.1 Java Interface

Figure 10-1 is an illustration of the Java interface that a user may utilize. This interface resides on the login nodes or utility servers and can be tunneled over SSH. The Java interface can utilize the existing Kerberos credential cache.

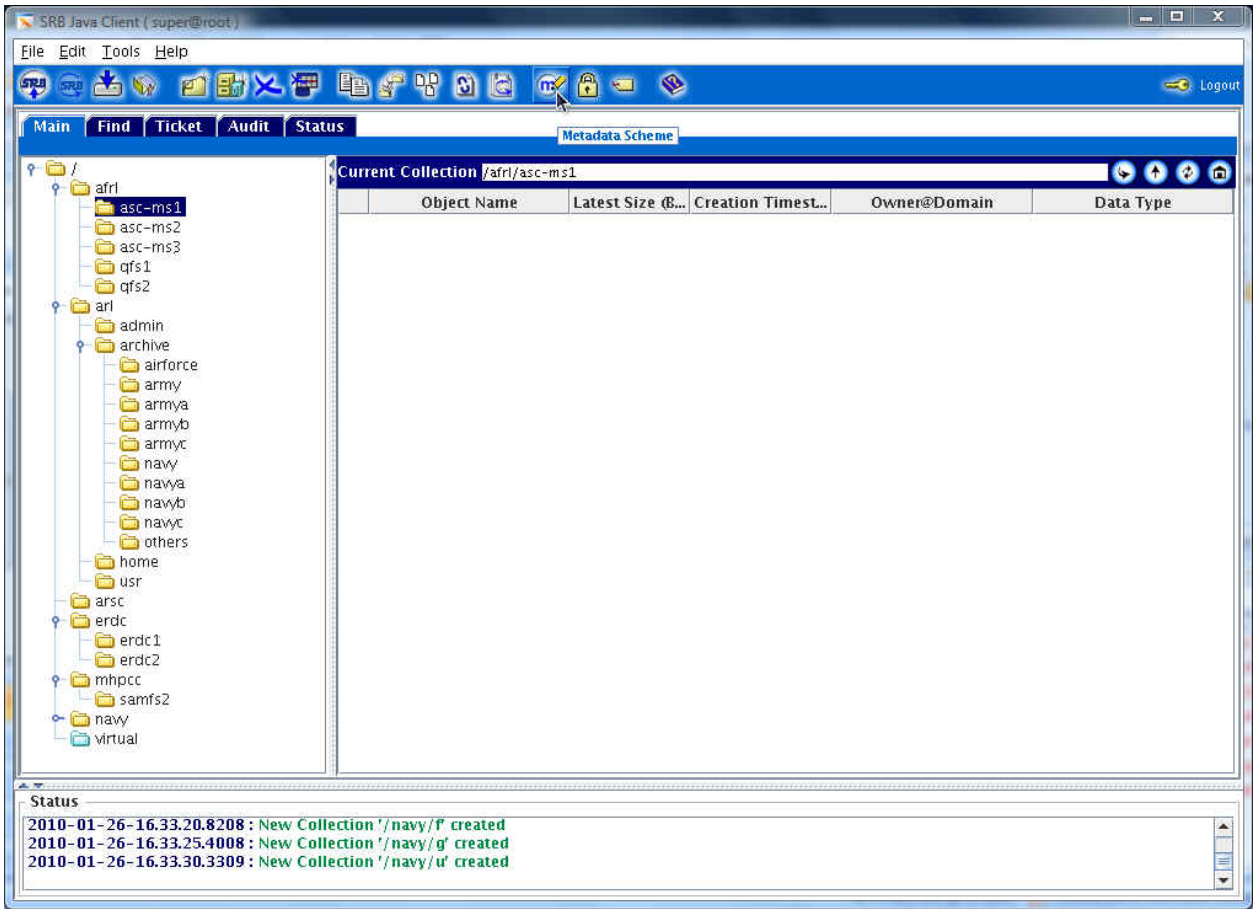


Figure 10-1. Java Interface

10.2 Scommands

The preferred access mechanisms for the Global Namespace are the Scommands to provide the equivalent of UNIX functionality and additional enhancements. These commands use the Kerberos credential cache for authentication. An illustration of the Sls command is shown in Figure 10-2.

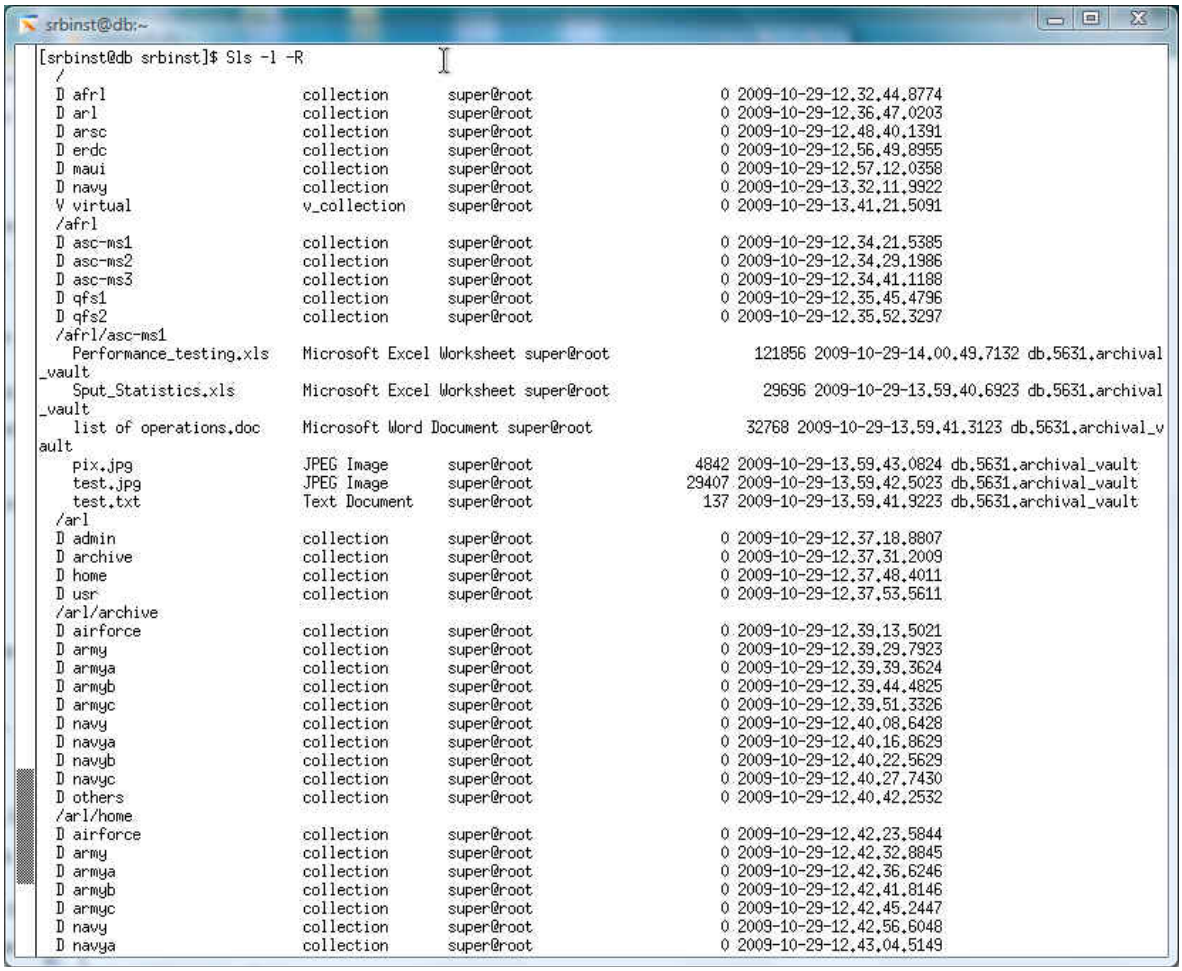


Figure 10-2. Scommands and SRB Mounts

The following table maps UNIX to Scommand functionality.

Table 10-1. UNIX to Scommand Functionality Mapping

UNIX Command	SRB Command	Comments
login	Sinit/Sshell	login into SRB, generally added into system login script
exit	Sexit	logout script
whoami	Senv	displays the current SRB session information
passwd	Spasswd	only changes SRB PASSWD_AUTH token, doesn't support Kerberos password change yet.
pwd	Spwd	displays working collection name
ls	Sls	displays Data Objects list from SRB Collections
sls –D	SgetD -x	displays HSM-specific information
cd	Scd	allows navigation within SRB Collection hierarchy
mkdir	Smkcoll	creates an SRB Collection
chmod	Schmod	grants/revokes SRB Object/Collection access control
chown	Schown	changes the SRB Object/Collection ownership
---	Sput	ingest/upload local Files/Directories into SRB Resources
---	Sget	download SRB Data Objects/Collections into local system disk
---	Sreplicate	create hard linked copies of an SRB object within the same SRB Collection
cat	Scat	displays the SRB Data Object's contents
rm	Srm	deletes SRB Data Objects/Collections

UNIX Command	SRB Command	Comments
cp	Scp	copies SRB Objects/Collections
mv	Smv	moves SRB Objects logically (name change) or physically between SRB Resources
ln	Slink	creates soft links on SRB Objects/Collections
getfattr/setfattr	Sscheme	lists or changes custom metadata values for SRB Objects/Collections
release	Spurge	issues the purge/release command for SRB Objects/Collections
archive/unarchive/rearch	Sarchive	issues the archive command for SRB Objects/Collections
stage	Sstage	issues the stage command for SRB Objects/Collections

10.3 File System Mounts

The SRB Preload Library provides a mechanism for some UNIX operating systems to pseudo-mount Global Namespace Collections. Figure 10-3 shows a listing of the pseudo-mount /srb with its sub-directories and files. The /srb pseudo-mount represents the root of the Global Namespace. The directories directly under /srb (i.e., /afrl, /erdc) are SRB Collections at the root of the Global Namespace:

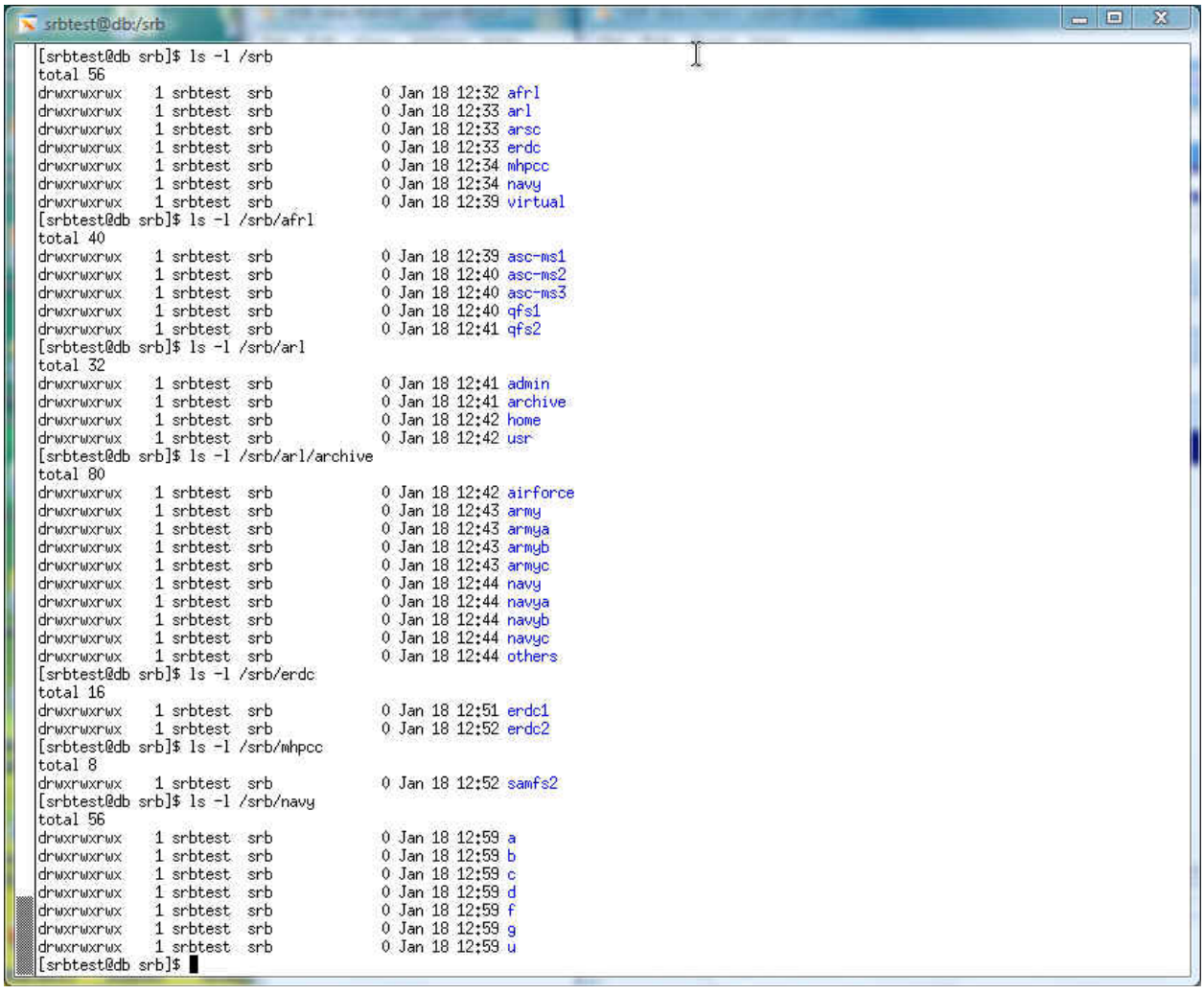


Figure 10-3. File System Mounts

The table below describes the support status for the preload libraries on various platforms.

Table 10-2. Support Status for Preload Libraries

UNIX OS	App Bit	EnvVarName	Status	OS Version
Linux	32	LD_PRELOAD	Yes	Red Hat Linux release 9 and SUSE Linux Enterprise 11
Linux	64	LD_PRELOAD	Yes	Red Hat Enterprise Linux ES release 3 and SUSE Linux Enterprise 11
Solaris Sparc	32	LD_PRELOAD_32	Yes	Solaris 10
Solaris Sparc	64	LD_PRELOAD_64	Yes	Solaris 10
Solaris Intel	32	LD_PRELOAD_32	Yes	Solaris 10
Solaris Intel	64	LD_PRELOAD_64	Yes	Solaris 10
HP UX	32	LD_PRELOAD	Yes	HP-UX 11i v1
HP UX	64	LD_PRELOAD	Yes	HP-UX 11i v1
AIX	32	---	There is no support for system API redirection	
AIX	64	---	There is no support for system API redirection	
IRIX	32	---	Not supported	
IRIX	64	---	Not supported	
CLE	32/64	--	Not supported	V2.1
UNICOS/lc	32/64	--	Not supported	V1.5, V2.0

The Preload Library is meant for advanced users only as it does not operate with every UNIX application and on all platforms. Although it is not recommended, the Preload Library can be used in the production environment. It is an open source package within the SRB.

The Preload Library functionality may be affected between OS versions (e.g. RH 4 vs. RH 5) because there may be some new system APIs being called by UNIX commands which are not yet implemented in the Preload Library.

Mixed bit mode applications i.e. one UNIX command is 32 and another is 64 bit, may not work in the same preload session.

The presentation of some UNIX attributes (file owner, group, mode, etc.) is not yet properly mapped to SRB Users and ACLs due to the limitations of the Preload Library at this time.

11 USER INTERACTION

11.1 HPCMP Kerberos

Access to SRB is available as long as a Kerberos credential cache exists on the local system. SRB's `Sshell/Sinit` Scommands will pick-up this Kerberos credential cache and authenticate the user in SRB by mapping the user principal in the credential cache to an SRB user.

As long as the credential cache is valid, all SRB transactions are properly authenticated and data transfers may be optionally integrity-checked or encrypted. If the credential cache expires, users will have to renew it just like they do today (e.g., `kinit`). As soon as the credential cache is valid again, SRB transactions and operations can continue.

To avoid such credential cache expirations during long-running processes, users must use the STORAGE Kerberos realm (see section 11.2.2).

11.1.1 .k5login

The .k5login file is used to map Kerberos user principals to local UNIX users. This is not a functional requirement for SRB to work. It is rather used for Kerberized SSH or similar services. SRB maintains the mapping of Kerberos user principals to local user names via user aliases.

11.2 User workflows

The following sections describe various workflows that users may perform within the SLM environment.

11.2.1 Metadata Entry

There are two Scommands available in SRB to manipulate metadata on files and directories – `Sscheme` and `SmodD -S`. These Scommands are described in the SRB User Guide. However, the HPCMP will provide its users an easier, more tailored mechanism to perform metadata entries and manipulations using the Sretain and Sdata commands. The Sretain and Sdata commands are described in detail by man pages and via command help. Following is a summary description:

Sretain is the command line interface to set a Retain_Time attribute on Storage Resource Broker (SRB) objects. The Sretain command updates the 'Admin' Scheme which is a System Scheme that is automatically applied during ingestion of objects into SRB. The Sretain command requires a days or date argument for the Retain_Time attribute and objects or collections to act upon. The Last_Review_Time is set automatically by the Sretain command. Sretain can also set HPCMP_Project_ID, the 13 character HPCMP project ID that this object is associated with.

Sdata allows one to display, set, change, or delete keyword-value pairs or the project in the Storage Resource Broker (SRB) metadata. Sdata accepts a keyword=value syntax and the scheme name is not required. For the Title, Creator, Subject, Description, Publisher, Contributor, Creation Date, Type, Document ID, and Rights keywords metadata values will be stored in the Dublin_Core scheme. The values for all other keywords will be stored in the Name_Value scheme.

11.2.2 Batch processing

The STORAGE Kerberos realm can be utilized to ensure that long-running operations have a continuous Kerberos session available to them. Such long-running operations typically occur during batch processing.

The workflow would be as follows: Users submit a job to the queuing system (i.e., PBS Pro). The queuing system schedules the job and remembers the user, which initiated it. When the job is ready to be executed, the queuing system retrieves an encrypted keytab file for the user that

initiated the job, decrypts the keytab, and authenticates the user to the STORAGE Kerberos realm, obtaining a long-running Kerberos session on behalf of the user. Any Sshell/Sinit Scommands within the job will now be able to make use of this long-running Kerberos session and authenticate the user to SRB mapping the Kerberos user principal name to an SRB User. At this point, the job has all the access permissions within SRB, which the interactive user would otherwise have.

11.2.3 Cron-initiated jobs

User-initiated cron jobs are compatible with the STORAGE Kerberos realm design if they leverage the queuing system.

11.2.4 Authentication/encryption options

The SLM solution will only implement a single authentication option – HPCMP Kerberos. However, once authenticated, the user has the option between multiple different data transfer mechanisms: plain text, integrity checked, and encrypted.

The following commands provide an example how to request these respective data transfer mechanisms from the system:

```
> Sinit –user scheder@HPCMP.HPC.MIL –auth KERBEROS_AUTH –comm PLAIN_TEXT
> Sinit –user scheder@HPCMP.HPC.MIL –auth KERBEROS_AUTH –comm KERBEROS_INTEGRITY
> Sinit –user scheder@HPCMP.HPC.MIL –auth KERBEROS_AUTH –comm KERBEROS_SECURE
```

Administrators may pre-set the default authentication and data transfer mechanisms; Home Collections; and Default Resources for each user. This can, for example, be accomplished by using aliases. The following alias provides an example on how to establish an SRB session with one of the local MCAT Server nodes using Kerberos authentication, setting the Default Resource to a local file system, and dropping the user into his/her local Home Collection:

```
alias Sinit='Sinit –host mcat1.arsc.edu,mcat2.arsc.edu –auth
KERBEROS_AUTH–rsrc arsc.u2; Scd /arsc/home/scheder'
```

11.2.5 File Access

SRB provides a Global Namespace for all files within the HPCMP across all the centers. Users do generally not have to pay attention to file systems or servers. Everything can be logically organized inside the Global Namespace. There is a quite a variety of file access options available in SRB. The following paragraphs shall discuss some of the options, which are preferred by the HPCMP.

11.2.5.1 Using SRB, including ingest

To store and retrieve files, there are basically two command-line options available – `Sput` and `Sget`. In both cases SRB handles the routing of I/O to or from the appropriate file system. Factors such as capacity, administrator preference, locality, and speed are taken into consideration. Administrators assign their users a Default Resource, which automatically determines which server and file systems files get stored on.

For example, if users wanted to archive the local directory "`Project 1`", all they would have to do is to issue the following command (`-R` stands for a recursive operation; the "`.`" represents the Current Collection in the Global Namespace):

```
> Sput -R "Project 1" .
```

In order to retrieve the same directory back from the archive into a local file system, the following command could be issued:

```
> Sget -R "Project 1" .
```

If users want to influence on which file system or site their files are stored or retrieved from, they can do so with the `-rsrc` and `-srsrc` arguments respectively. In all cases, an SRB session needs to be established with `Sshell` or `Sinit` first before any of the other Scommands (i.e., `Sput`, `Sget`) will work.

11.2.5.2 Using POSIX, including SRB registration

Besides Scommands, users can also utilize POSIX interfaces such as the SRB Preload Library. Anything prefixed with the /srb directory, will automatically be processed through the Global Namespace.

For example, the following command stores the local directory "Project 1" into the user Collection (as defined by environment variable $SLM_COLLECTION) of the archive:

```
> cp -r "Project 1" /srb/$SLM_COLLECTION
```

The following command retrieves the same directory back from the archive into a local file system:

```
> cp -r "/srb/$SLM_COLLECTION/Project 1" .
```

Finally, users can also just register local files or directories into the SRB Global Namespace so that, for example, metadata attributes can be associated with the files or directories. The registration process does not perform any file I/O but rather just creates a database entry pointing to the files or directories in their file systems. Registration can only function properly if the file system where the to-be-registered files or directories reside was properly mapped into an SRB Physical Resource.

The following command, when executed on the SLM Server, registers the existing "Project 2" directory into SRB using the arsc.u2 Physical Resource, which is mapped to the /export/archive/u2 file system on the SLM Server (-R stands for a recursive operation):

```
> Sregister -R -rsrc arsc.u2 "/export/archive/u2/Project 2" "Project 2"
```

Files registered in such a way are still accessible through the local file system but at the same time are registered in SRB and can hence be managed by ILM and associated with metadata.

11.2.6 Queries

One of the advantages of a database-resident file system such as SRB is the ability to perform queries against the database using all the metadata attributes associated with the files and directories in the file system.

There are generally two mechanisms to perform such queries in SRB: a) `SgetD` or `Sls -policy <query>` or b) Virtual Collections.

Both mechanisms utilize SRB's pseudo SQL query language, which is described in the SRB User Guide.

For example, to query for all Data Objects containing the word 'tino', the following Scommand can be used:

```
> SgetD -R -policy "(DATA_OBJECT.data_name like '*tino*')"
[Collection Name]   [Object Name] [Data Type]               [Last Size]
-------------------------------------------------------------------------
/home/scheder       tino.txt      Text Document             1200
/projects/scheder   by_tino.doc   Microsoft Word Document   45339
```

The same query could be used to create a Virtual Collection, which virtually contains all Data Objects with the word 'tino':

```
> Smkcoll -policy "(DATA_OBJECT.data_name like '*tino*')" tino_files
> Scd tino_files
> Sls
/home/scheder
  tino.txt      1200
/projects/scheder
  by_tino.doc   45339
```

Virtual Collections can be then be used in subsequent data retrieval or transfer operations such as this data retrieval:

```
> Sget -R tino_files .
```

Queries can also be used to find objects that match individual attributes in the Name_Value scheme. For example, if there are two attributes – color and size – in the Name_Value scheme, the following query could find all objects that are red and large:

```
SgetD -R -policy "DATA_OBJECT.data_id IN (select DATA_OBJECT.data_id where
name_value.name = 'color' and name_value.value = 'red') AND DATA_OBJECT.data_id IN
(select DATA_OBJECT.data_id where name_value.name = 'size' and name_value.value =
'large')"
 [Collection Name]            [Object Name]      [Data Type]        [Latest Size]
 ------------------------------------------------------------------------------
 /htl/htl.spike.samfs1/ianni  marble_large_red   generic            0
```

11.2.7 Reports

Virtual Collections as described in the previous paragraph can also be used to accomplish a reporting function. For example, the following command registers a Virtual Collection called

“expiring_files_user” which contains all the user’s files that are about to expire from the archive. The contents of this particular Virtual Collection will only contain files owned by the current user:

```
> Smkcoll –policy "((EXPRESSION.create_age > Admin.Retention_Period –
Policy.Warning_Period_User) OR (EXPRESSION.current_timestamp > Admin.Next_Review_Time
- Policy.Warning_Period_User)) AND (Admin.Admin_Hold = 'no') AND (DATA_OBJECT.owner_id
= EXPRESSION.current_user_id) AND (DATA_OBJECT.data_type not like '*collection')"
expiring_files_user
```

A listing of this Collection would show all the user’s expiring files like this:

```
> Scd expiring_files_user
> Sls
/arl/samfs1/home/tkendall
  file1.txt               561
  file2.txt               0
/arl/samfs1/home/tkendall/project_x
  file3.bin               10000000
  file4.txt               1232
```

The retention period for those files might then be extended using a command such as the following:

```
> Sscheme –w –val "Admin.Retention_Period::600" expiring_files_user
```

Note that the general users will not use this command directly. They will most likely use the Sretain command that is being developed with input from the Customer Experience Workgroup.

A similar Virtual Collection can be created for the PI or S/AAA to show them files of users whom they oversee. This relationship – PI to users whom they oversee – is defined by Collection curatorship of PIs. The following Virtual Collection shows expiring files for a PI if the Collection curatorship is properly setup:

```
> Smkcoll –policy "((EXPRESSION.create_age > Admin.Retention_Period -
Policy.Warning_Period_PI) OR (EXPRESSION.current_timestamp > Admin.Next_Review_Time -
Policy.Warning_Period_PI)) AND (Admin.Admin_Hold = 'no') AND (DATA_OBJECT.data_type
not like '*collection')" expiring_files_pi
```

11.2.8 File Transfers

SRB provides a high-speed data transfer protocol, which can accommodate LAN or WAN data transfers. The following sections will show how SRB can be used to transfer files within or across the centers.

To transfer files within SRB, there are commands equivalent to the UNIX `cp` or `mv` – `Scp` and `Smv`. An Sreplicate command provides the creation of synchronized copies across multiple file systems or centers. The SRB commands function similar to their UNIX cousins with the exception that they also have the ability to perform transfers across servers.

11.2.8.1 Internal to DSRC

If users have access to multiple file systems within a DSRC, they can transfer their data across such file systems with ease. For example, the following command creates a synchronized

Replica of all files in the "`Project 3`" directory on the `arsc.u3` Physical Resource, which represents the `/export/archive/u3` file system on `seawolf`:

```
> Sreplicate –R –rsrc arsc.u3 "Project 3"
```

11.2.8.2 Between DSRCs (in Replicated Mode)

Similar to internal data transfers within a DSRC, data can be transferred between DSRCs. The only difference, as far as the user is concerned, is that the Resource specified in the transfer command represents a remote file system. For example, the following command executed at ARSC would physically migrate the "Project 3" directory to the erdc1 Resource at ERDC:

```
> Smv –R –rsrc erdc1 "Project 3"
```

Note that all such data transfers are protected from being written to open-research storage when coming from a sensitive site. Any such file transfer will automatically be denied unless the Mandatory Access Control bits are changed to flag the Data Object as Open Research first.

11.2.8.3 In/Out of DSRCs/HPCMP

Files can be ingested into or retrieved from DSRCs by using a Kerberized workstation or laptop, which has SRB Client software installed. This access may be restricted by rules as defined in Table 4-8 and other firewall rules. Scommands such as `Sput` and `Sget` can be used from such a machine just as described in section 11.2.5.1 above.

The fact that data enters or leaves the system is audited by the SLM solution.

11.2.9 File Retention Renewal/ Notification Process

The periodic notification for files that are subject to removal works as follows. All files, subject to removal, will be aggregated to allow a single notification be sent out to the user on a monthly basis. This notification may simply contain a Virtual Collection that shows all of the to-be-expired files of the user instead of sending a long list of files. The user has the choice to either receive or decline this notification. The choice to either receive or decline a notification is set in the metadata associated with each file (Warning_Note attribute in the Admin scheme). If after receipt of a notification the user does not update the metadata associated with the files within the Warning_Period_User, the PI and the S/AAA will receive a notification within the Warning_Period_PI to consider an update of the metadata. The files will be removed from the HPCMP SLM Systems on the Retain_Time or the Next_Review_Time – whichever comes first.

The following policies are directly related to this File Retention Renewal / Notification Process:

- All files registered or ingested into the SLM system will automatically obtain "Admin" scheme attributes (see section 8.2) according to user defaults, administrator defaults, or parent collection.
- Users are free to change many of these attributes to, for example, extend the Retention_Period of a set of files.

- Users can change the Last_Review_Time without making changes to the Retention_Period or other attributes of the Admin scheme that affect archive and retention of the file. Changing of the Last_Review_Time alone represents a review.
- Default Retention_Period is 30 days.
- Warning_Period_User is 28 days.
- Warning_Period_PI is 15 days.
- Review_Period 3 years.

12 CROSS-SITE METADATA REPLICATION

Although the initial implementation of SLM will be in Isolated Enclave Mode, this section describes the implementation of Replicated Mode so that a true Global Namespace may be implemented in the long term. Please refer to 15APPENDIX C - for recommendations on how to implement Isolated Enclave Mode in a manner to preserve the future capability of migrating to fully Replicated Mode.

In Replicated Mode metadata stored in the MCAT Database at each site is replicated to all other sites so that all sites can share a single Global Namespace, which enables data sharing, organization, and discovery across sites.

The mechanism to perform this replication is Oracle Streams, which basically logs all transactions, which have modified the MCAT Database and ships those to the other sites. If connectivity to other sites is severed, all modifying transactions will be logged until connectivity is re-established. In the meantime, local MCAT Database access continues to operate with full functionality.

While a full description of the Oracle Streams is out of the scope of this document, the concept is well described in Oracle Streams documentation. However, a brief description is outlined in the next few paragraphs.

In its simplest terms, there are three processes employed for each replication stream from one site database to another. These are named capture, propagation and apply processes. The capture process monitors the redo log for changes, reformats these changes into logical change records (LCR) and enqueues these LCRs into a capture queue. The propagation process has the responsibility of propagating the LCRs captured to the target database. The capture and propagation processes operate on the source database.

The LCRs propagated from the source database are enqueued into an apply queue at the target database. At this point, the apply process dequeues each LCR and applies it to the target database. These processes are illustrated in the following diagram.

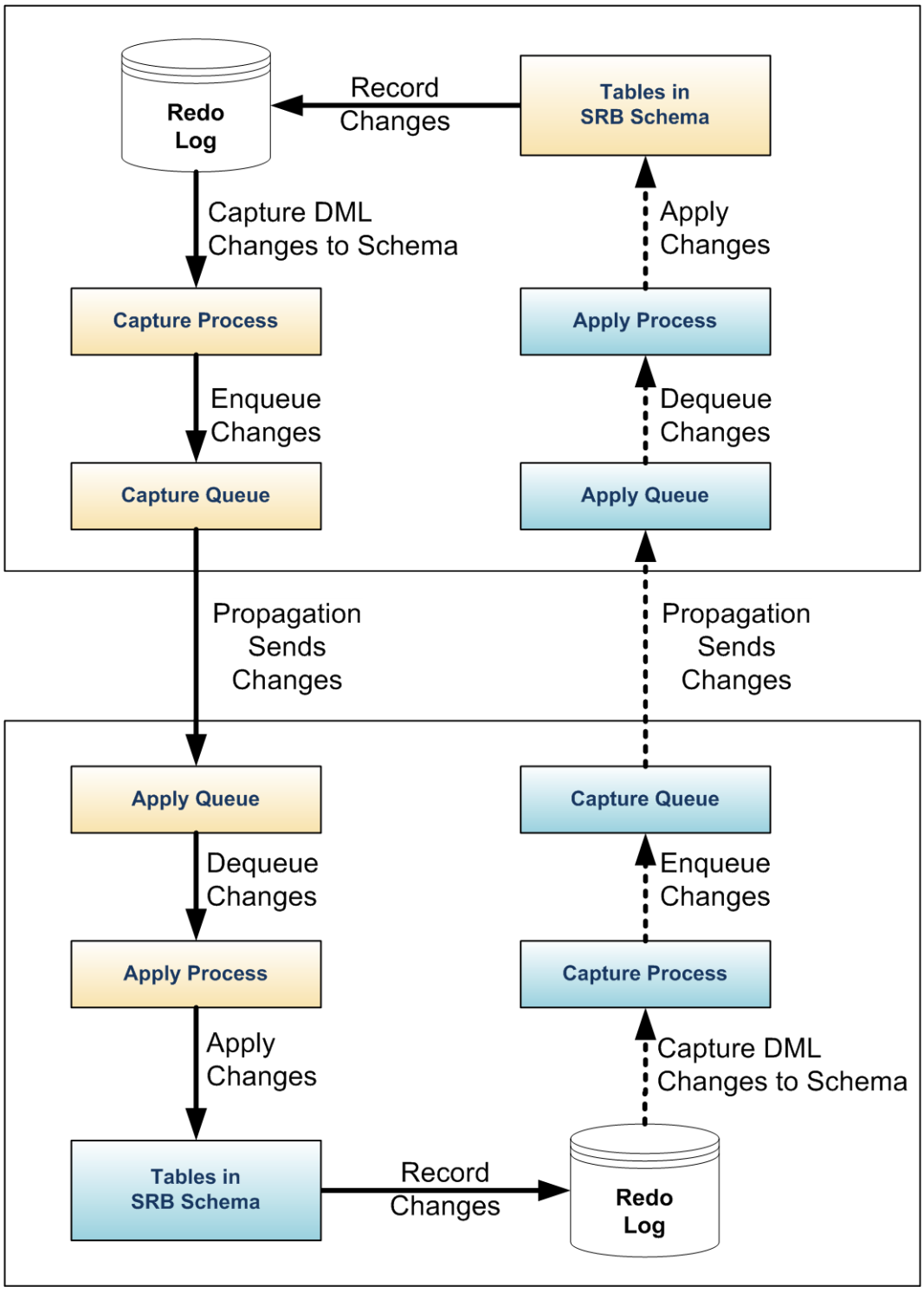


Figure 12-1. Oracle Streams Flow

It is important to note that each stream is unidirectional. Thus, for two databases to fully replicate actions originating at that database, there must be capture and propagation processes at that database and a corresponding apply process at the other database.

12.1 Replication Objects

It is recommended that replication occur at the "SRB" schema level for all Data Description Language (DDL, statements that make changes to the schema; by default NOT starting with "MCAT_") and all Data Manipulation Language (DML, inserts, updates, and deletes) transactions. This means that all tables, views, triggers, constraints, etc. within the SRB schema are replicated. There are two exceptions of tables that will not be replicated: MCAT_COUNTER and MCAT_DATA_AUDIT. These tables will be excluded via negative capture rules. Since each site has its own table counters, it is important that the MCAT_COUNTER table not be replicated. The MCAT_DATA_AUDIT table will be excluded in order to reduce streaming bandwidth.

12.2 Conflict Resolution

With multiple masters allowing independent write access to their local MCAT Databases, there is a potential for conflicts to occur. Conflicts in Oracle Streams Replication are handled using automated conflict and error handlers. Different SRB Objects (i.e., files, users, metadata) are treated in different automated ways. Conflict resolution is addressed at length in section 12.6.

12.3 Counters

Counters in the MCAT_COUNTER and MCAT_LONG_COUNTER tables need to be unique per site. Hence, it is recommended to start the counters at different offsets for the different sites. A possible offset scheme could be as follows:

Table 12-1. Counter Offsets Across DSRCs

DSRC	USER_ID	RSRC_ID	SCHEME_ID	DATA_ID Ranges
AFRL	100,000	100,000	100,000	10,000,000,000 11,000,000,000
ARL	200,000	200,000	200,000	20,000,000,000 21,000,000,000
ORS	300,000	300,000	300,000	30,000,000,000 31,000,000,000
ERDC	400,000	400,000	400,000	40,000,000,000 41,000,000,000
MHPCC	500,000	500,000	500,000	50,000,000,000 51,000,000,000
NAVO	600,000	600,000	600,000	60,000,000,000 61,000,000,000

The counters are chosen so that an overlap is highly unlikely. If an overlap does occur, the user inserting a new SRB Object (e.g., a resource or user) will receive a unique constraint violation error. A database administrator can then easily adjust the counter offset for the counter, which received the unique constraint violation. For example, if AFRL created more than 100,000 users, it would run into the USER_ID values for ARL and would hence get a unique constraint violation error. The database administrator for AFRL could then simply set the USER_ID counter value to 1,000,000.

12.4 Oracle Streams Replication Approaches

Oracle Streams replication can be configured to replicate the entire database schema or individual tables. The advantage of replicating the entire schema is that schema changes (e.g. column additions to schema tables) are automatically replicated to all sites.

There are two popular topologies that can be implemented, hub and spoke and n-way. Each has its own advantages and disadvantages. As stated above, replication is unidirectional per stream. One advantage of the hub and spoke is that it reduces the number of streams emanating from each database. This topology, as depicted in the figure below uses an intermediate database in a hub-spoke configuration where the hub collects data from all sites and then distributes to all sites. Note that the streams communication path is only between the site database and the hub database.

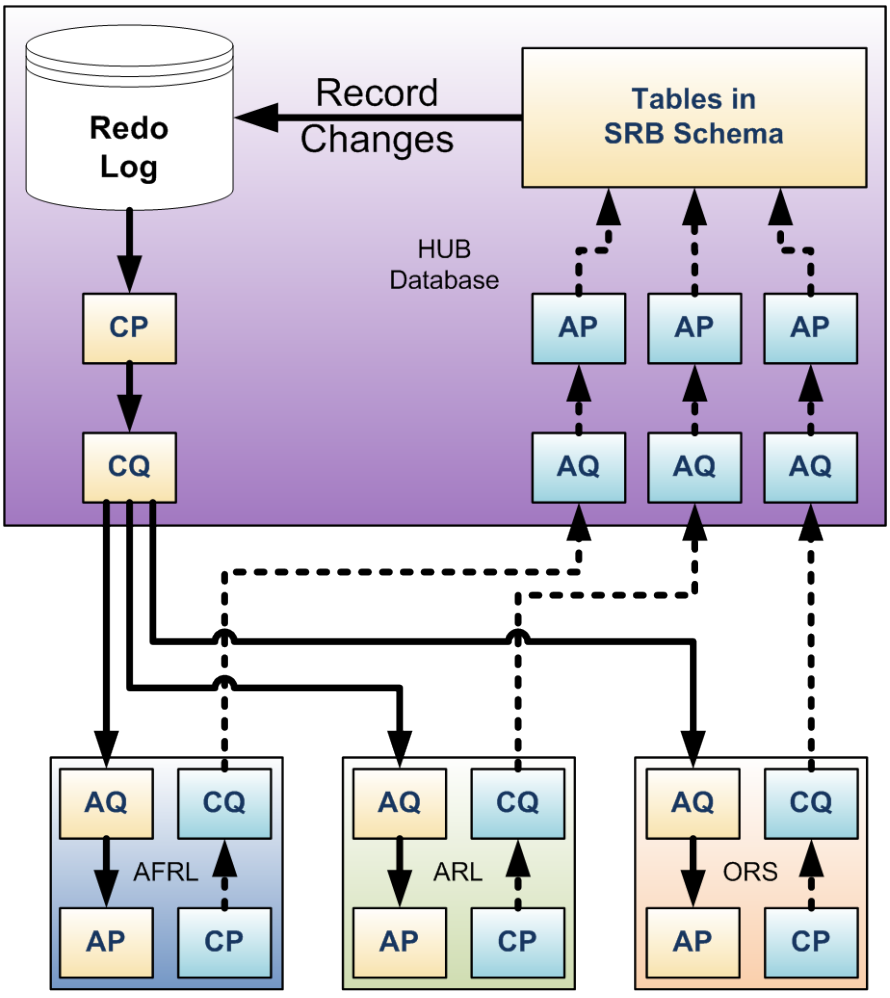


Figure 12-2. Oracle Streams Hub-Spoke Configuration

It is important to note that this is useful only if it is being used purely for the purpose of managing the streams environment. That is, there is no data being created in the hub database directly from any application. The entire setup is cleaner and easier to manage. Also, adding a

new site would only require synchronizing with the one hub database rather than all the other site databases in the environment. A side effect of this is that the overall process of adding a new site is quicker. Conversely, a disadvantage of this configuration is that the hub becomes a single point of failure for the replication infrastructure. Another disadvantage includes the need for an additional set of database servers, licenses and the implicated support infrastructure. Also, streams configuration would be more complex due to the necessity of maintaining LCR tags.

The recommended approach to replication is to implement an n-way topology among sites. The advantage of this topology is that all sites replicate to each other and therefore have all the metadata available to them locally. The disadvantage is that it consumes more bandwidth than may be available. A sample topology using three sites is depicted in the following diagram.

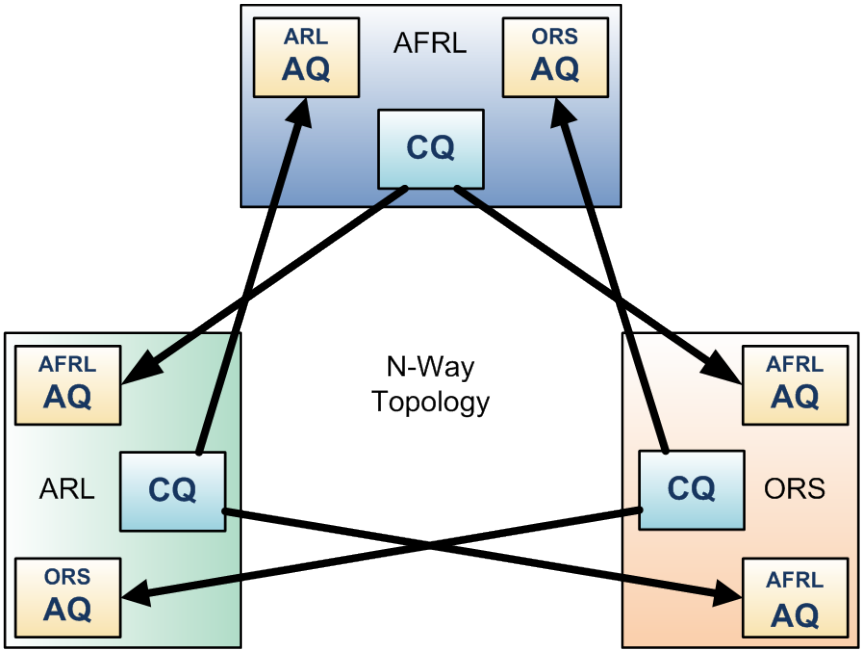


Figure 12-3. Oracle Streams N-Way Configuration Single Capture

N-way replication supports both single and multiple capture/apply queues. The figure above illustrates a single capture queue configuration. In this configuration, the single capture queue serves to hold all the LCRs to be propagated to all sites. The disadvantage of this configuration is that if the streaming to one site is interrupted, there exists a potential for the capture queue to fill-up. At this point no more LCRs can propagate to the other sites as the queue has no capacity to hold additional LCRs. The alternative is to use multiple capture queues for each propagation stream as shown in the diagram below.

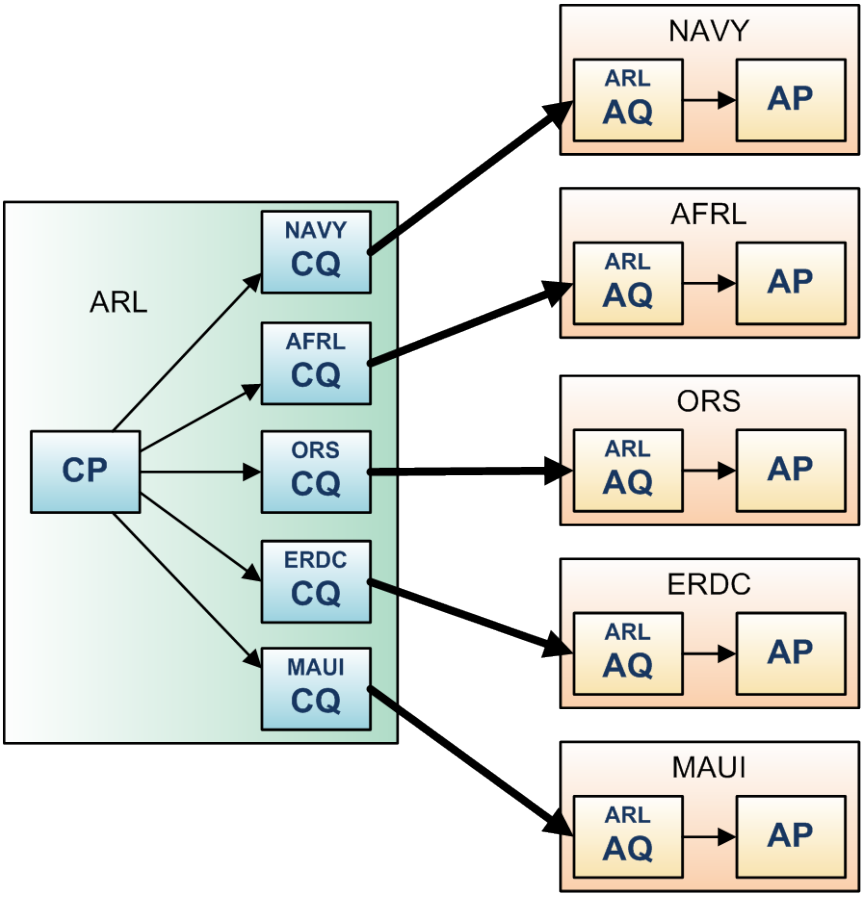


Figure 12-4. Oracle Streams N-Way Configuration Multiple Capture

While one disadvantage of multiple capture queues is the added setup and management complexity, the addition of a new site is much more efficient and has less impact on the streaming environment. That is, since in the single capture queue configuration the queue must be used to hold LCRs that have yet to be replicated to the new site, all replication to the other sites must cease until the new database catches up. This issue does not exist in the multiple capture queue configuration.

An alternative approach is the pair-wise topology. Although this can be useful for backup purposes, it is not a valid approach for Replicated Mode.

12.5 Streams Initialization Parameters

The following parameters affect Oracle Streams Replication performance. Performance tuning is an ongoing process and based on application performance level, these parameters will likely go through changes.

Table 12-2. Streams 11g Initialization Parameters

Parameter	Recommended	Settings
STREAMS_POOL_SIZE	0 or > 100MB	
JOB_QUEUE_PROCESSES	1000	Set the value to maximum number of jobs that would ever be run concurrently on a system PLUS add hoc request jobs. Refer Metalink note 578831.1 on tuning
SGA_TARGET	12 GB	Specifies the total size of all SGA components, sized automatically.
UNDO_RETENTION	900	Specifies (in seconds) the amount of committed undo information to retain in the database. For a database running one or more capture processes, make sure this parameter is set to specify an adequate undo retention period. If you are running one or more capture processes and you are unsure about the proper setting, then try setting this parameter to at least 3600. Also make sure the undo tablespace has enough space to accomodate the UNDO_RETENTION setting.
AQ_TM_PROCESSES	<don't specify>	AQ_TM_PROCESSES controls time monitoring on queue messages and controls processing of messages with delay and expiration properties specified. You do not need to specify a value for this parameter because Oracle Database automatically determines the number of processes and autotunes them, as necessary. Therefore, Oracle highly recommends that you leave the AQ_TM_PROCESSES parameter unspecified and let the system autotune. Note: If you want to disable the Queue Monitor Coordinator, then you must set AQ_TM_PROCESSES to 0 in your parameter file. Oracle strongly recommends that you do NOT set AQ_TM_PROCESSES to 0. If you are using Oracle Streams, then setting this parameter to zero (which Oracle Database respects no matter what) can cause serious problems.
COMPATIBLE	11.2.0.1	Set compatible to the version of Oracle that you are using in order to use all replication features of that version of the database
GLOBAL_NAMES	TRUE	Set the value to TRUE for replication.

Other tips for Oracle 11g Streams:

- Database wide supplemental logging imposes a significant overhead and may affect performance. This should therefore be avoided.
- Capture Parallelism = 1 is the recommended setting and is the default.
- Increase the SDU in a Wide Area Network for Better Network Performance In addition, the SEND_BUF_SIZE and RECV_BUF_SIZE parameters in the listener.ora and tnsnames.ora files increase the performance of propagation on your system. These parameters increase the size of the buffer used to send or receive the propagated messages. These parameters should only be increased after careful analysis of their overall impact on system performance. For more details, refer to Oracle Document ID 780733.1.
- Apply parameters:
  parallelism = 4
  _dynamic_stmts = 'Y'
  _hash_table_size = 1000000
  disable_on_error = 'N'
- If possible, decrease transaction sizes to less than 1000 LCRs. Large or long transactions will affect Streams. These may result in Queue spill or Apply spill. As outlined, most of the areas which can cause issues relate to large and long running transactions which may be associated with Queue spill and Apply spill. Queue spill is more onerous than apply spill.

12.6 Oracle Streams Replication Conflict Resolution

Replication conflicts can occur in a replication environment that permits concurrent updates to the same data at multiple sites. There are a number of conflicts and resolution is presented below. An important prerequisite for many automated conflict resolution scenarios is that presence of a TABLE_TIMESTAMP column in each MCAT database table, which gets updated whenever an update is made to a table row.

12.6.1 Update Conflicts

An update conflict occurs when the replication of an update to a row conflicts with another update to the same row. Update conflicts can happen when two transactions originating from different sites update the same row at nearly the same time.

Update Conflicts are resolved in the primary error handler (i.e., update_dml_error in error_handler.sql) and utilize the TABLE_TIMESTAMP column in every table to determine whether the incoming LCR or the existing row in the target database is more recent. If the incoming LCR's TABLE_TIMESTAMP is determined to be more recent, not all columns in the row are necessarily updated. Only the columns contained in the LCR are being updated. This allows users to change different columns in the same row at the same time and still have all of the columns committed.

Email alerts are sent to the DBA group only when an error occurs in the error handler.

Update Conflict Scenarios

The following tables demonstrate how the update conflict resolution works in a multi-site environment. In all tables the values in the cell abide by the following format:

localname [n:origname]: *localname* refers to the object name on the current site; *n* refers to the site upon which the object was originally created; origname refers to the original name of the object as created on Site n

Table 12-3. Update Conflict Example 1: No conflict – most common replication situation

Time	Action	Site 1	Site 2	Site 3
1	Create object xy on Site 1	xy [1:xy]		
2	Propagate 1:xy to Site 2	xy [1:xy]		
3	Propagate 1:xy to Site 3	xy [1:xy]		
4	Apply 1:xy on Site 2; no conflict; no name change	xy [1:xy]	xy [1:xy]	
5	Apply 1:xy on Site 3; no conflict; no name change	xy [1:xy]	xy [1:xy]	xy [1:xy]
6	Any attempt to create xy on Site 1, 2, or 3 is rejected since xy exists on all sites	xy [1:xy]	xy [1:xy]	xy [1:xy]

Table 12-4. Update Conflict Example 2: "Insert" conflict – applies to object names

Time	Action	Site 1	Site 2	Site 3
1	Create object xy on Site 1	xy [1:xy]		
2	Create object xy on Site 2	xy [1:xy]	xy [2:xy]	
3	Propagate 1:xy to Site 2	xy [1:xy]	xy [2:xy]	
4	Propagate 1:xy to Site 3	xy [1:xy]	xy [2:xy]	
5	Propagate 2:xy to Site 1	xy [1:xy]	xy [2:xy]	
6	Propagate 2:xy to Site 3	xy [1:xy]	xy [2:xy]	
7	Apply 1:xy on Site 2; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name, and send email to owner of 2:xy re name change	xy [1:xy]	xy [1:xy] xy_201 [2:xy]	
8	Apply 1:xy on Site 3; no conflict; no name change	xy [1:xy]	xy [1:xy] xy_201 [2:xy]	xy [1:xy]
9	Apply 2:xy on Site 1; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]	xy [1:xy]
10	Apply 2:xy on Site 3; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name; at this point the conflicts have been resolved on all sites, no data has been lost, and object references are consistent among all sites	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]

Table 12-5. Update Conflict Example 3: "Insert" conflict; variation of Example 2

Time	Action	Site 1	Site 2	Site 3
1	Create object xy on Site 1	xy [1:xy]		
2	Create object xy on Site 2	xy [1:xy]	xy [2:xy]	
3	Propagate 1:xy to Site 2	xy [1:xy]	xy [2:xy]	
4	Propagate 1:xy to Site 3	xy [1:xy]	xy [2:xy]	
5	Propagate 2:xy to Site 1	xy [1:xy]	xy [2:xy]	
6	Propagate 2:xy to Site 3	xy [1:xy]	xy [2:xy]	
7	Apply 2:xy on Site 1; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name	xy [1:xy] xy_201 [2:xy]	xy [2:xy]	
8	Apply 2:xy on Site 3; no conflict; no name change	xy [1:xy] xy_201 [2:xy]	xy [2:xy]	xy [2:xy]
9	Apply 1:xy on Site 2; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name, and send email to owner of 2:xy re name change	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]	xy [2:xy]
10	Apply 1:xy on Site 3; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name; at this point the conflicts have been resolved on all sites, no data has been lost, and object references are consistent among all sites	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]

Table 12-6. Update Conflict Example 4: Multiple "insert"conflict[1]

Time	Action	Site 1	Site 2	Site 3
1	Create object xy on Site 1	xy [1:xy]		
2	Create object xy on Site 2	xy [1:xy]	xy [2:xy]	
3	Create object xy on Site 3	xy [1:xy]	xy [2:xy]	xy [3:xy]
4	Propagate 1:xy to Site 2	xy [1:xy]	xy [2:xy]	xy [3:xy]
5	Propagate 1:xy to Site 3	xy [1:xy]	xy [2:xy]	xy [3:xy]
6	Propagate 2:xy to Site 1	xy [1:xy]	xy [2:xy]	xy [3:xy]
7	Propagate 2:xy to Site 3	xy [1:xy]	xy [2:xy]	xy [3:xy]
8	Propagate 3:xy to Site 1	xy [1:xy]	xy [2:xy]	xy [3:xy]
9	Propagate 3:xy to Site 2	xy [1:xy]	xy [2:xy]	xy [3:xy]
10	Apply 1:xy on Site 2; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name, and send email to owner of 2:xy re name change	xy [1:xy]	xy [1:xy] xy_201 [2:xy]	xy [3:xy]
11	Apply 1:xy on Site 3; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name, and send email to owner of 3:xy re name change	xy [1:xy]	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_301 [3:xy]
12	Apply 2:xy on Site 1; conflict occurs, so rename the onject with the higher data_id using conflict handler appending data_id to name	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_301 [3:xy]
13	Apply 2:xy on Site 3; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy] xy_301 [3:xy]
14	Apply 3:xy on Site 1; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name	xy [1:xy] xy_201 [2:xy] xy_301 [3:xy]	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy] xy_301 [3:xy]
15	Apply 3:xy on Site 2; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name	xy [1:xy] xy_201 [2:xy] xy_301 [3:xy]	xy [1:xy] xy_201 [2:xy] xy_301 [3:xy]	xy [1:xy] xy_201 [2:xy] xy_301 [3:xy]

[1] This is extremely rare unless 2 or more sites have been disconnected from the network, an update operation was performed on the same objects, and the sites were subsequently reconnected.

Table 12-7. Update Conflict Example 5: Conflict associated with update

Time	Action	Site 1	Site 2	Site 3
1	Site 1 loses network connectivity			
2	Create object xy on Site 1	xy [1:xy]		
3	Propagate 1:xy creation to Site 2 (blocked)	xy [1:xy]		
4	Propagate 1:xy creation to Site 3 (blocked)	xy [1:xy]		
8	Create object xy on Site 2	xy [1:xy]	xy [2:xy]	
9	Propagate 2:xy creation to Site 1 (blocked)	xy [1:xy]	xy [2:xy]	
10	Propagate 2:xy creation to Site 3	xy [1:xy]	xy [2:xy]	
5	Update xy on Site 2	xy [1:xy]	xy [2:xy]	
6	Propagate updated 2:xy to Site 1 (blocked)	xy [1:xy]	xy [2:xy]	
7	Propagate updated 2:xy to Site 3	xy [1:xy]	xy [2:xy]	
11	Apply 2:xy creation on Site 3; no conflict; no name change	xy [1:xy]	xy [2:xy]	xy [2:xy]
12	Site 1 regains network connectivity	xy [1:xy]	xy [2:xy]	xy [2:xy]
13	Propagation of 1:xy creation to Sites 2 & 3 is unblocked	xy [1:xy]	xy [2:xy]	xy [2:xy]
14	Apply 1:xy creation on Site 2; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name	xy [1:xy]	xy [1:xy] xy_201 [2:xy]	xy [2:xy]
15	Apply 1:xy creation on Site 3; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name	xy [1:xy]	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]
16	Apply 2:xy creation on Site 1; conflict occurs, so rename the object with the higher data_id using conflict handler appending data_id to name	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]
17	Propagation of 2:xy update to Site 1 is unblocked	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]
18	Apply 2:xy update on Site 1; the data name xy from Site 2 no longer matches the data name on Site 1 because the replica on the sites have been renamed; the solution for this is not to match data name, but rather data id	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]	xy [1:xy] xy_201 [2:xy]

12.6.2 Uniqueness Conflicts

A uniqueness conflict occurs when the replication of a row attempts to violate entity integrity, such as a PRIMARY KEY or UNIQUE constraint. For example, consider what happens when two transactions originate from two different sites, each inserting a row into a respective table

with the same unique constraint key value. In this case, replication of the transactions causes a uniqueness conflict.

The application data model and primary key definition is designed to avoid conflict from primary key constraint violations. This is accomplished by using different unique key pools for different sites (see section 12.3 Counters). The uniqueness conflicts from unique constraint violations are resolved using a combination of primary and secondary error handlers (i.e., unique_dml_error and dml_error2 in error_handler.sql). These error handlers generally operate by the principle that they rename the conflicting column with the larger unique key to a value like "<column_value>_X_<unique_key>".

Email alerts are sent to the successfully renamed object owners and the DBA group; on error email alerts are sent to the DBA group.

12.6.3 Delete Conflicts

A delete conflict occurs when two transactions originate from different sites, with one transaction deleting a row and another transaction updating or deleting the same row, because in this case the row does not exist to be either updated or deleted.

Delete conflicts cannot be resolved automatically.

The secondary error handler generates an email alert that is sent to the DBA group.

12.6.4 Foreign Key Conflicts

A foreign key conflict occurs when the apply process applies a row LCR containing a change to a row that violates a foreign key constraint.

Foreign key conflicts cannot always be resolved automatically. But the secondary error handler (i.e., dml_error2 in error_handler.sql) retries failed operations at a later time when the cause of the conflict may have been resolved – maybe by having received an additional LCR that adds a missing row and hence avoids the conflict.

If the foreign key conflict remains in the error queue for longer than the configurable grace period (i.e., mail_failure_grace_period in error_handler.sql), an email alert is sent to the DBA group.

12.6.5 Conflict Examples

The following table gives some examples of the possible conflicts and shows how they are being avoided. Note that the table is by no means comprehensive and that the error handlers are capable of resolving many more conflicts than indicated.

Table 12-8. Conflict and Resolution Examples.

Conflicts	Recommended Solution	Probability of Occurrence during Operations	Solution Type	Notes
DATA OBJECTS, COLLECTIONS in MCAT_DATA_INFO and MCAT_COLL_INFO				
Data_id	Use unique ID pools for each site.	Extremely low	Design	Referential integrity compromised if occurs.
Same data_name and collection_id	Append '_X_data_id' to data_name with larger data_id to avoid conflict. Send email to the owners.	Low	Primary and secondary error handler	Data Objects or Collections will have a '_X_data_id' appended to their name.
Update conflicts	Always use latest timestamp to avoid conflicts.	High	Primary error handler	Latest update will erase earlier updates.
REPLICAS in MCAT_DATA_REPLICA				
Data_id	Use unique ID pools for each site.	Extremely low	Design	Referential integrity compromised if occurs.
Data_id, replication_id	Use of additive (if different rsrc_id and multiple '1's keep one '1', change the other one to '2') to avoid conflicts.	Medium	Primary error handler	None.
Update conflicts	Always use latest timestamp to avoid conflicts.	High	Primary error handler	Latest update will erase earlier updates.
USERS, GROUPS, DOMAINS, LOCATIONS in MCAT_USER_ALIAS				
User_id	Use unique ID pools for each site.	Extremely low	Design	Referential integrity compromised if occurs.
Same user_name and domain_name	Append '_X_user_id' to user name with larger user_id to avoid conflict. Send email to the users.	Low	Primary error handler	Users, groups, domains, or locations will have a '_X_user_id' appended to their name.
Update conflicts	Always use latest timestamp to avoid conflicts.	Low	Primary error handler	Latest update will erase earlier updates.
SCHEMES, COLUMNS in MCAT_SCHEME_INFO and MCAT_COLUMN_INFO				
scheme_id, column_id	Use unique ID pools for each site.	Extremely low	Design	Referential integrity compromised if occurs.
same column_name and scheme_name	Scheme and column operations need to be handled via organizational change management procedures in order to avoid conflicts.	Extremely low	Procedure	As scheme and column conflicts usually involve DDL conflicts, they cannot be handled automatically.
PHYSICAL RESOURCES, LOGICAL RESOURCES, CLUSTER RESOURCES in MCAT_RSRC_INFO				
Resource_id	Use unique ID pools for each site.	Extremely low	Design	Referential integrity compromised if occurs.
Same resource_name	Append '_X_rsrc_id' to resource name with larger rsrc_id to avoid conflict. Send email to DBA group.	Low	Primary and secondary error handler	Resources will have a '_X_rsrc_id' appended to their name.

DATA TYPES, EXTENSIONS in MCAT_DATA_TYPE and MCAT_DATA_TYPE_EXT				
Data_type_id	Data type additions need to be handled via organizational change management procedures in order to avoid conflicts.	Low	Procedure	Unique data type ID pools do not exist for every site.
Same data_type_name	Data type additions or name changes need to be handled via organizational change management procedures in order to avoid conflicts.	Low	Procedure	Primary and secondary error handlers are not implemented for data_type_name conflicts.
Duplicate extensions	Data type extension additions or name changes need to be handled via organizational change management procedures in order to avoid conflicts.	Low	Procedure	Primary and secondary error handlers are not implemented to handle primary key conflicts on the mcat_data_type_ext table.
ACCESS CONTROL LIST in MCAT_DATA_ACCS, MCAT_RSRC_ACCS, MCAT_USER_ACCS, MCAT_CONFIG_ACCS and MCAT_COLUMN_ACCS				
Same object, same user_id, different access_id	Always use latest timestamp to avoid conflict.	Medium	Primary error handler	Latest update will erase earlier updates.

13 CENTER-WIDE FILE SYSTEM

Along with the Utility Servers, each DSRC has a high-speed parallel Panasas file system called the center-wide file system (CWFS). The CWFS is accessible directly from the HPC systems and from the Utility Servers. SRB manages the archive as well as the SRB-enrolled files in the CWFS.

It is important to ensure that the SLM solution interoperates with the CWFS so that data can be moved seamlessly between the archive and the CWFS in both directions. The CWFS is directly attached to a dedicated server or cluster under SLM control. This scenario is discussed in the following section.

13.1 CWFS Directly Attached Under SLM Control

In this scenario, shown in Figure 13-1 below, the CWFS is implemented as a standalone system or cluster into each DSRC environment. The SLM directly manages the data on the CWFS. This standalone system has front-end servers provide data access to the user and I/O to the storage.

The CWFS provides a Data Management Application Programming Interface (DMAPI) or similar event stream interface for use by the SLM in updating the SLM metadata about file system objects.

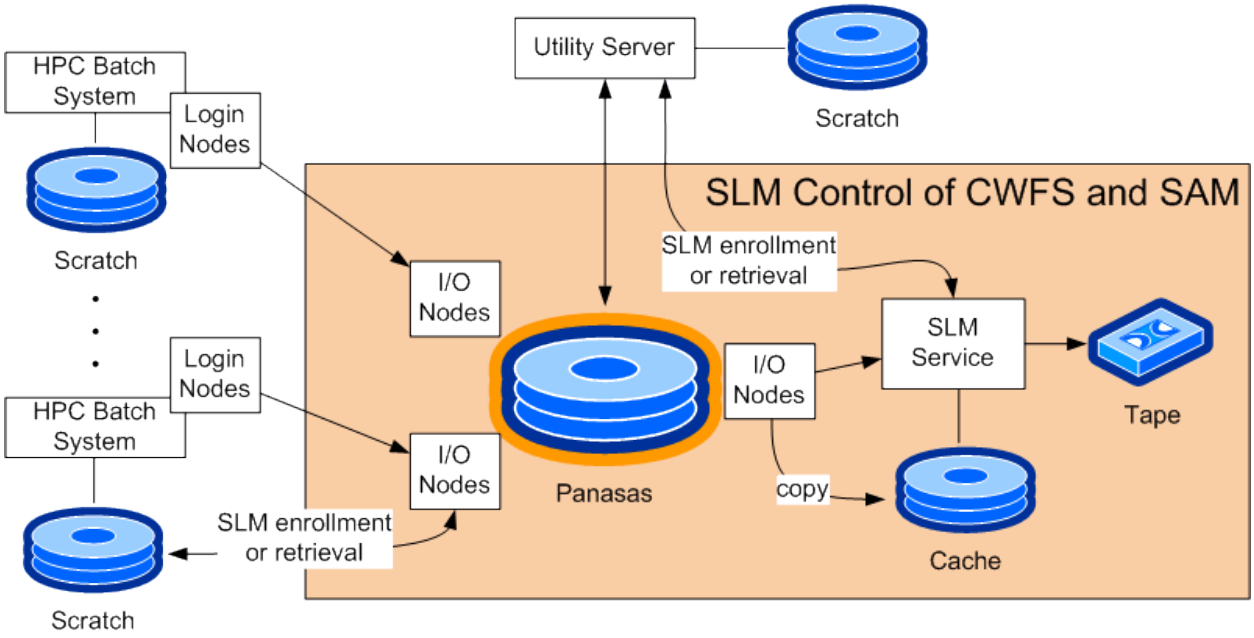


Figure 13-1. CWFS Directly Attached Under SLM Control

Using the DMAPI or similar interface, the SRB Sync Daemon can automatically register all CWFS files and directories with SRB in real-time mode. SRB maintains files on the CWFS for a configurable period of time (e.g., 30 days) and then automatically deletes them from the CWFS. Only user-selected files will be archived to the SAM-QFS resource(s).

Consequently, all files are removed from the CWFS after the specified period of retention. Users might not expect this behavior when they access the CWFS directly, thus circumventing SRB access. Files can be brought back from the archive into the CWFS using an Sreplicate command.

13.2 Pros and Cons of the Selected Design Scenario

The selected design has the benefit that the CWFS interoperates with the archive. The CWFS user can utilize SRB as a metadata store; archive; and local or cross-site query mechanisms. Additionally, the SRB ILM Daemon can be used to auto-archive any files on the CWFS that are enrolled into SRB and that are older than the configurable period.

SRB Agents will run on all the Login Nodes and Utility Servers where an SLM-controlled CWFS is mounted. The CWFS is configured as an SRB Cluster Resource on each of those SRB Agents. In this way the CWFS has multiple data paths accessible from every server.

Furthermore, it needs to be understood by the users that the access control mechanism in the local CWFS and the SRB Global Namespace are not directly synchronized. If access permissions in the CWFS are granted using group owners (GIDs), those group ownerships are not automatically translated into group permissions inside SRB. As long as the data is accessed through the local file system, the group permissions will work. But after the file is purged from the CWFS, groups will no longer have access to the file through SRB.

13.3 Data Flow between CWFS and Archive

Files and directories are stored from the CWFS into SRB automatically. Sput can be used to physically transfer data into the CWFS or the SAM-QFS archive even if the CWFS is not mounted locally.

Files or directories can be retrieved from SRB into the CWFS using Scommands such as Sget or Sreplicate. Sget will export the data outside of the SRB Federation with the local UNIX user performing the writing of the files. After export, those files are then outside of SRB's control until an SRB Sync Daemon registers them back into SRB. In the case of Sreplicate the files remain under SRB control and are actually written by the SRB Agent's system process. SRB 2010 introduced a mechanism where the ownership of files created in local file systems are changed to the current SRB user. In this fashion the current UNIX user will be able to access those replicated files.

If users prefer to put/archive files directly from the HPC Login Nodes into the archive, they can do so, bypassing the CWFS. Thus, users can put files into SRB using either the CWFS or the SAM-QFS archive resources.

13.4 Center-wide File System (Interface Specification, API)

In the Center-Wide File System (CWFS) design it was stressed that a DMAPI or similar interface was required for a seamless CWFS integration with SLM. For a successful near real-time synchronization of the CWFS with SRB, the following events and attributes are required:

Events:

1) file/dir created
2) file/dir attributes changed
3) file closed
4) file/dir renamed
5) file/dir deleted

Attributes:

1) Inode number
2) Inode generation number
3) Full path and file/dir names (if possible; if not possible Parent Inode number and Parent Inode generation number are required and CWFS needs to provide an ioctl for a fast name lookup by inode/gen; in the case of file/dir rename, both old and new paths and names (or inode/generation numbers) are needed)
4) UNIX mode including file/dir type (regular file, directory, link, etc.)
5) File size in bytes
6) Checksum and algorithm identifier (optional)
7) Most recent access time – file “access time”: create file, on every file close; directory “access time”: create directory, opendir directory.
8) Most recent change time – for file “change time”: create file, on file close only when file has changed, chmod file, chown file; for directory “change time”: create file in directory, delete file from directory, create subdir, chmod directory, chown directory
9) Most recent modification time – for file “modify time”: create file, on file close only when file has changed; for directory “modify time”: create file in directory, delete file from directory, create subdir.
10) creation time (optional but very useful)
11) UID
12) GID
13) Date and time of the event

The DMAPI or similar interface should ideally be available as a POSIX C language binding. Alternatively, the events can be written to a log file and then read-out and cleared by the SRB Sync Daemon in real-time mode.

14 KMS

14.1 Architecture Overview

Each primary HPC center, the DR site and the two classified sites will deploy a KMS cluster consisting of two Key Management Appliances (KMAs). Wherever doing so enhances overall

availability, the two KMAs will be physically separated in a way that minimizes the risk of their simultaneous failure.

The two KMAs will share a common Management Network for replication purposes. Management of the KMS cluster configuration via the KMAs is via the KMS Management GUI which runs on a Management Network-attached Solaris workstation.

Tape drives connect to the KMAs via a separate, non-routable KMS Service Network. Key requests are serviced via this network.

Should a single KMA be lost, ongoing KMS replication enables the surviving KMA to continue to process requests from tape drives in the cluster. Should both KMAs be lost simultaneously, assuming the other archive infrastructure has survived, recovery would be based on KMS backups.

It is possible to incorporate one or more remote KMAs into a cluster or unite all 7 sites in a single cluster to establish off-site replicas of the KMS configuration and reduce dependency on local backups. However, it is expected that creating and sending regular local backup files to the DR site via DREN will achieve the same purpose and eliminate a point of coordination across Center boundaries.

The architecture discussed above is reflected graphically in the following diagrams.

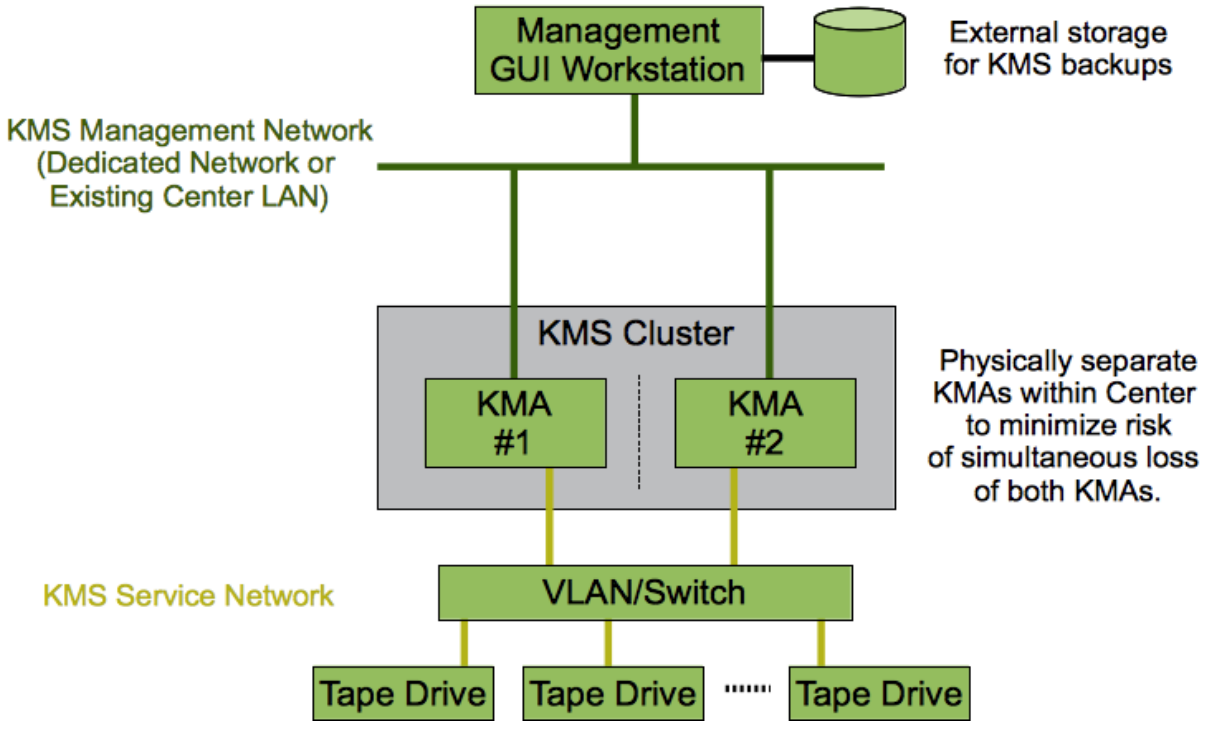


Figure 14-1. Dual-KMA KMS Cluster.

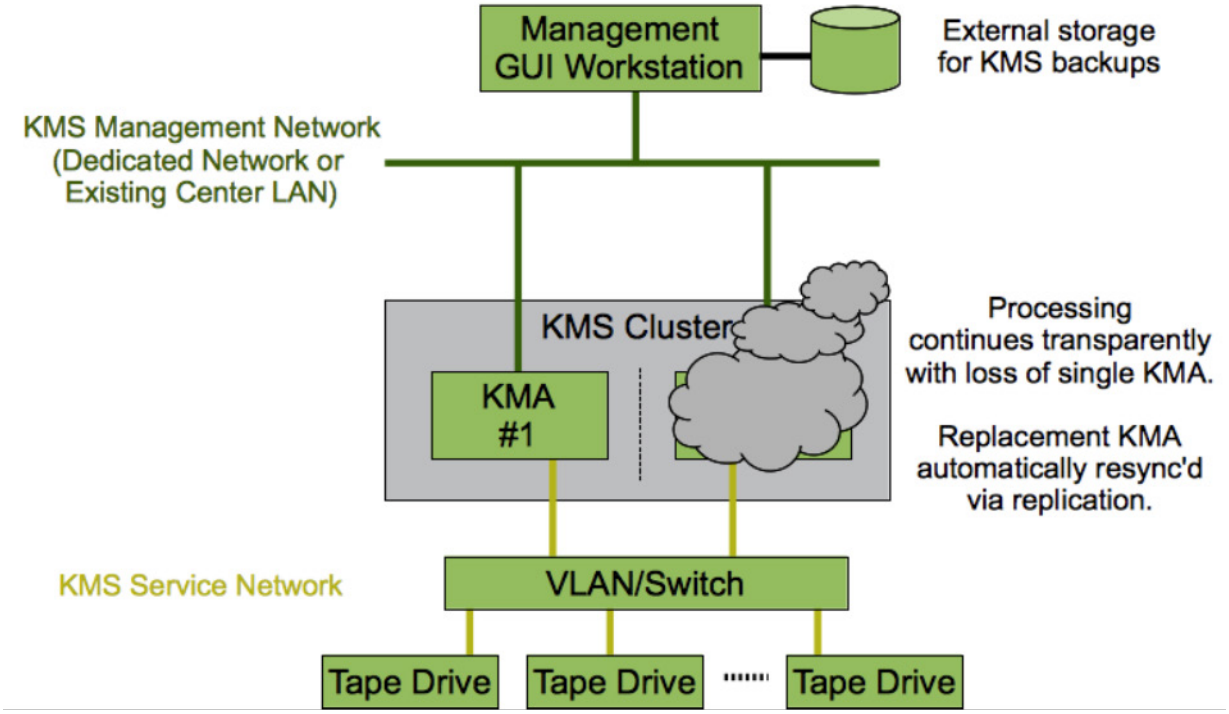


Figure 14-2. Loss of Single KMA on a Dual-KMA Cluster

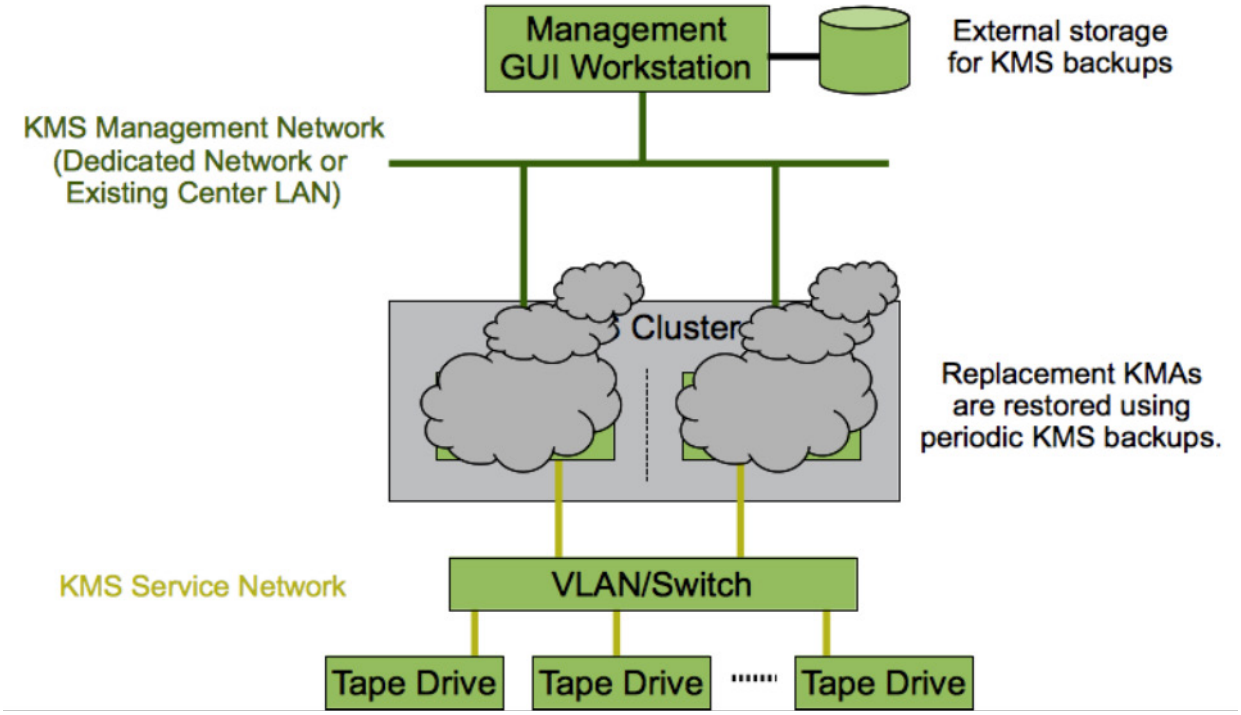


Figure 14-3. Loss of Both KMAs on a Dual-KMA Cluster

14.2 IP Networking Requirements

14.2.1 Physical Networks

The following physical networks will be used for KMS traffic.

Table 14-1. KMS Network Traffic Patterns

Proposed Physical Network to be Used	New or Existing Network	Summary of New Ethernet Ports Deployed
KMA Management VLAN	New	1 per local center KMA (2 total) 1 per Management Workstation
Service Network	Existing at some centers New at some centers	1 per encrypting tape drive 1 or 2 (IEEE 802.1AX-2008 link aggregated) per local center KMA

Networks above identified as “new” may be implemented with dedicated network hardware or as a new VLAN.

14.2.2 Traffic and Physical Networks

Under normal operating circumstances, the elements of the DoD HPC KMS architecture communicate over the above physical networks as described below.

Table 14-2. KMS Network Traffic Patterns

Traffic	Hardware Entity 1	Hardware Entity 2	Proposed Physical Network to be Used	Ports for Inter-site Traffic
Certificate Authority (Management)	KMA & Management GUI	KMA	Center LAN	TCP/3331 (bi-directional)
Certificate Service (Management)	KMAs & Management GUI	KMA	Center LAN	TCP/3332 (bi-directional)
Certificate Authority (Service)	Tape Drives	KMA	Service Network	No inter-center traffic
Certificate Service (Service)	Tape Drives	KMA	Service Network	No inter-center traffic
KMS Management	Management GUI	KMA	Center LAN	TCP/3333 (bi-directional)
Agent	Tape Drives	KMA	Service Network	None
KMA Discovery (Management)	KMAs & Management GUI	KMA	Center LAN	No inter-site traffic
KMA Discovery (Service)	Tape Drives	KMA	Service Network	No inter-site traffic
KMA Replication	KMA	KMA	Center LAN	TCP/3336 (bi-directional)

14.2.3 KMS Networking Terminology

The Oracle StorageTek KMS architecture defines two network concepts: a Management Network and a Service Network.

The Service Network is the network at a center over which tape drives communicate with the KMS cluster. This network is typically private and non-routable. The table above employs the term Service Network in the same way KMS documentation does.

In the KMS documentation, other KMS traffic is said to traverse the “Management” Network. For clarity's sake in the sections above, this network is identified in the tables above as the “Center LAN”.

14.2.4 Console Network

Initial KMA setup and certain technical support functions require console access to the appliance. Console access may be achieved via direct attachment of a center-provided keyboard and monitor.

If no keyboard and monitor are available for this purpose, centers may choose to configure the HTTPS-based remote console CLI, either on the primary center LAN or a dedicated Console Management network where one exists. One connection per KMA is required in this case.

Table 14-3. KMS Console Connections.

Center	Entity(-ies)	Connection to Console Management LAN
Centers and DR SIte	Primary KMAs ELOM (aka NET MGT)	1x new copper 10/100 per appliance (2x total per center)

14.3 KMS Physical Configuration

The KMS deployment at each center and the DR site requires the following hardware.

- Two KMAs.
- A Solaris (or Windows) KMS management workstation having monitor, keyboard and mouse. The workstation should have removable storage or network connectivity to a remote site to allow for transport of backups to a remote site.
- An edge switch that connects the encryption-enabled tape drives locally at the library(-ies), connected via uplink to a VLAN shared with the site's KMAs, forming the private Service Network.
- A new or existing switch or VLAN that connects the KMAs to each other and to the KMS management workstation, forming the Management Network.
- Encryption-enabled tape drives.

The following sections detail the physical requirements for the KMAs. Note that the KMA platform is undergoing a refresh of the underlying hardware. Detail is not yet available. The numbers cited below are for the currently shipping X2200-based KMA.

14.3.1 Rack Requirements

The rack requirements in the table below are per KMA.

Table 14-4. KMS Rack Requirements.

Attribute	Value
Dimensions	1.7”H (1RU) x 16.8”W x 25”D
Weight	24.7 lbs

14.3.2 Power Requirements

The power requirements in the table below are per KMA.

Table 14-5. KMS Power Requirements.

Attribute	Value
AC Power	100-120/200-240 VAC @ 50-60 Hz
Max Power Consumption	450 W
Max Heat Output	1416 BTU/hr

14.3.3 Network Attachment

This table summarizes from a different perspective the information presented in section 14.2 above.

Table 14-6. KMS Network Attachment.

Ethernet Port Identifier	Function	Network Connected
LAN 0	Management Network	Center LAN
LAN 1	Appliance Remote Console	If desired, Console Management Network (if one exists) or Center LAN
LAN 2	Service Network for tape drives	Service Network
LAN 3	Service Network aggregation for tape drives	Service Network

14.3.4 Backup Power (UPS)

The KMS is designed to be resilient to power failures. Should all KMAs in a cluster be taken offline due to a power failure, key requests (whether for old or new keys) would not be serviced, but once power is restored, the KMAs are designed to resume operation without administrator intervention.

14.3.5 Hardware Configuration Drawing

To be elaborated at time of acquisition.

14.4 KMA, Management Station and Tape Drive Pre-Configuration

Certain parameters need to be specified for each entity prior to incorporation into the cluster.

14.4.1 KMA Pre-Configuration

The following template captures appliance configuration needed prior to KMS cluster integration.

Table 14-7. KMA Configuration Template.

Attribute	Value	Notes
Tech Support account on KMAs	Disable	
DHCP / Static IP Addresses / Subnet Mask / IPv6 Addresses (if any) on Management Network	Center specified	
Gateway on Management Network	Center specified	Must be specified if needed to allow KMAs to see each other or management workstation
DHCP / Static IP Address / Subnet Mask / IPv6 Address (if any) on Service Network	Center specified	The Service network for KMAs is a private non-routable network
Gateway on Service Network	None	
DNS configuration	Center specified	
KMA identifier within KMS cluster	Center specified (Example scheme: KMA-ARL-01, KMA-ARL-02)	

14.4.2 Management Station Pre-Configuration

Each center will require at least one KMS management workstation running the Oracle Crypto KMS Manager GUI 2.x software.

The application is available for Solaris and Windows. Java 1.5 or later is required.

14.4.3 Tape Drive Pre-Configuration

The following template captures the information needed for tape drives before they can connect to the KMS cluster and be “enrolled”. This information must be specified for each tape drive via the Virtual Operator Panel (VOP) software.

Table 14-8. Tape Drive Configuration Template.

Attribute	Value	Notes
DHCP / Static IP Address / Subnet Mask on Service Network	Center specified	The Service network is a private non-routable network between KMAs and tape drives.
Gateway on Service Network	None	

14.5 KMS Configuration

Configuring the KMAs together into a functioning cluster that manages a center's keys requires that information in this section be specified per cluster.

14.5.1 KMS Users by Role

Initial configuration of the KMS requires that the following cluster-wide users with the following (pre-defined) roles be configured.

14.5.1.1 Key Split Quorum Users

Operations on the KMS cluster that may pose a security risk, such as adding a new KMA to the cluster, require a quorum of a pre-defined set of users to provide their quorum passphrase. The set of quorum user names and passphrases are separate from the user names and passphrases used for day-to-day administration of the KMS cluster.

The total number of users in the group and the required number of users for quorum are expected to be subject to center security policies. A configuration template is provided for center planning purposes.

Table 14-9. KMS User Management Template.

Attribute	Value	Notes
Number of quorum users	Center determined	Larger numbers make it easier to muster a quorum to take both valid and malicious action.
Names of quorum users	Center determined	
Required quorum for security-sensitive operations	Center determined	Larger quorums are more secure, but harder to muster.

14.5.1.2 Security Officer Users

The security officer role allows management of the KMS Cluster's security settings, users, centers and key transfer partners. Example tasks include:

- Initiate, along with a key split quorum, the addition of a KMA to a cluster.
- Reset a KMA to factory defaults and optionally zeroize it.
- Lock a KMA to prevent its use for key operations. Unlocking additionally requires a quorum.
- View the list of key split quorum users. Modifying the list additionally requires a key split quorum.
- Manage KMS users and roles.
- Initiate core security backups.
- Modify cluster security parameters such as audit log sizes, login attempt limits, passphrase length, inactivity timeout.
- Configure KMAs for SNMP and NTP.
- Configure, along with a key split quorum, any key transfer partners.

To ensure continuity of KMS operations, it is recommended that each center's KMS cluster have at least two users with the Security Officer role.

A configuration template is provided for center planning purposes.

Table 14-10. KMS Security Officer Template.

Attribute	Value	Notes
Users with Security Officer role	Center determined	At least two are recommended.

14.5.1.3 Operator Users

Users with the operator role manage tape drives, media and keys within the KMS cluster. Example tasks include:

- Define a new tape drive to the cluster.
- Remove a tape drive from the cluster.
- Destroy key material that is post-operational.
- Import and export keys among key transfer partners.

Certain operations are only possible by a user having the Operator role. It is required that at least one user be assigned this role.

A configuration template is provided for center planning purposes.

Table 14-11. KMS Operator User Template.

Attribute	Value	Notes
Users with Operator role	Center determined	At least one user with this role is required in order to perform the entire range of KMS operations.

14.5.1.4 Backup Operator Users

Users with the Backup Operator role perform KMS backups. Example tasks include:

- Creating backups.
- Confirming the destruction of backups.

Certain operations are only possible by a user having the Operator role. It is recommended that at least one user be assigned this role.

A configuration template is provided for center planning purposes.

Table 14-12. KMS Backup Operator User Template.

Attribute	Value	Notes
Users with Backup Operator role	Center determined	At least one user with this role is required in order to perform the entire range of KMS operations.

14.5.1.5 Compliance Officer Users

Users with the Compliance Officer role manage key policies and key groups and determines which tape drives and transfer partners can use which key groups. Example tasks include:

- Define key management policies (encryption period and cryptoperiod).

- Create key groups from which keys are assigned to tape drives.

Certain operations are only possible by a user having the Compliance Officer role. It is required that at least one user be assigned this role.

A configuration template is provided for center planning purposes.

Table 14-13. KMS Compliance Officer User Template.

Attribute	Value	Notes
Users with Compliance Officer role	Center determined	At least one user with this role is required in order to perform the entire range of KMS operations.

14.5.1.6 Auditor Users

Users with the Auditor role can view information about the KMS cluster. They cannot change the KMS configuration. Example tasks include:

- List security parameters.
- View audit logs.
- View network, SNMP, NTP configuration.

Anything that can be done with the Auditor role can be done by users in other roles. It is not required to have a user in this role.

A configuration template is provided for center planning purposes.

Table 14-14. KMS Auditor Template.

Attribute	Value	Notes
Users with Auditor role	Center determined	None are required.

14.5.2 KMAs

New KMA's have to be defined to the cluster as described in this section before they can be joined to the cluster. The following information has to be specified.

14.5.2.1 First KMA in the KMS Cluster

When a cluster is created with the first KMA, the following information is specified.

Table 14-15. Required First KMA Information.

Attribute	Value	Notes
KMA Identifier	Center determined	See naming scheme specified in section 14.4.1
Key Split Credentials	Center determined	See key split quorum configuration specified in section 14.5.1.1
Initial Security Officer	Center determined	See list of security officers specified in section 14.5.1.2

Attribute	Value	Notes
Autonomous Unlock Setting	Disable	Determines whether a KMS automatically begins operation after power up without an explicit unlock by the quorum users. Disabling removes risk of key compromise if KMA is physically removed at the expense of convenience on KMA power up.
Key Pool Size	1000	Each KMA pre-generates and maintains a pool of keys. These pre-operational keys must be backed up or replicated before a KMA will provide them to a tape drive for use in protecting data. This helps to ensure that a key will never be permanently lost, even in disaster scenarios. A smaller key pool size prevents unnecessary initial database (and backup) size, but requires frequent backups or a reliable network to ensure that activation-ready keys are always available. Conversely, a large key pool size is more tolerant of infrequent backups or unreliable network connections between KMAs, but the large number of pre-generated keys causes the database (and backups) to be quite large.
NTP Server	Center determined	KMAs in a cluster must keep their clocks synchronized. Specify an NTP server if one is available on the center network. Otherwise, manually specify the date and time to which the KMA clock should be set.

14.5.2.2 Subsequent KMAs

The following information is required to be defined in the cluster before adding additional KMAs to an established cluster.

Table 14-16. Required Subsequent KMA Information.

Attribute	Value	Notes
KMA Identifier and Passphrase	Center determined	See naming scheme specified in section 14.4.1. Administrators define these two pieces of information to the established cluster prior to a KMA attempting to join the cluster. When the new KMA does attempt to join, these credentials must match.

14.5.3 Sites

KMS maintains the notion of a “site” to allow key requests to be serviced from the “nearest” KMA in a wide area cluster.

In each HPC KMS cluster, only one site will be defined. The example shows a list of sites for AFRL's KMS cluster.

Table 14-17. KMS Sites.

HPC Center	KMS Site Name	Notes
AFRL	AFRL	Used with cluster entities (KMAs and tape drives) physically located at AFRL.

14.5.4 Tape Drives

New tape drives have to be defined to the cluster as described in this section before they can be enrolled into the cluster. The following information needs to be specified.

Table 14-18. Tape Drive Information.

Attribute	Value	Notes
Tape Drive Identifier and Passphrase	Center determined (Example tape drive naming scheme: T10KC-MHPCC-001, T10KC-MHPCC-002, …)	Administrators define these two pieces of information to the established cluster prior to a attempting to enroll a tape drive the cluster. When the new tape drive does attempt to join, it must provide these credentials.
Tape Drive Site	KMS site name corresponding to HPC center where tape drive is located	See section 14.5.3

14.5.5 Security Parameters Configuration

Adjustment of security parameters such as audit log retention parameters, login attempt limits, passphrase policy, FIPS-compliance mode are specified in this section.

Table 14-19. KMS Security Parameters.

Attribute	Value	Notes
Short Term Retention Audit Log Size Limit	Center determined	Measured in entries. (min/default/max) = (1,000/10,000/1,000,000)
Short Term Retention Audit Log Lifetime	Center determined	Measured in days. (min/default/max) = (7/7/25,185)
Medium Term Retention Audit Log Size Limit	Center determined	Measured in entries. (min/default/max) = (1,000/100,000/1,000,000)
Medium Term Retention Audit Log Lifetime	Center determined	Measured in days. (min/default/max) = (7/90/24,855)
Long Term Retention Audit Log Size Limit	Center determined	Measured in entries. (min/default/max) = (1,000/1,000,000/1,000,000)
Long Term Retention Audit Log Lifetime	Center determined	Measured in days. (min/default/max) = (7/7/24,855)
Login Attempt Limit	Center determined	Failed login attempts before an entity is disabled. (min/default/max) = (1/5/100)
Passphrase Minimum Length	Center determined	(min/default/max) = (8/8/64)
Management Session Inactivity Timeout	Center determined	(min/default/max) = (0/15/60)

Attribute	Value	Notes
FIPS Mode Only	On	Whether key transfer imports and exports are restricted to KMS 2.1's (and later) FIPS format.

14.5.6 Key Policy and Group Configuration

Tape drives are associated with key groups, which are governed by a key policy. Key policies prescribe key type and key lifecycle.

When a new piece of media is loaded into an encrypting drive, it is assigned a key which is added to the key group.

It is expected that a single key policy, implemented by a single key group will suffice for each center's needs. Here is a configuration template for a key policy.

Table 14-20. Key Policy Configuration Template.

Attribute	Value	Notes
Key Policy ID	Center determined (Example: “SAM”)	
Key Policy Description	Center determined (Example: “Default SAM key policy.”)	
Key Type	Center determined (Example: AES-256)	
Encryption Period	Center determined (Example: 7 days)	Time period for which a newly assigned key can be used to encrypt data.
Cryptoperiod	Center determined (Example: 2 years)	Time period for which a newly assigned key can be used to decrypt (but not encrypt) data. The relevant NIST standard does allow the key to decrypt beyond this period.
Allow Export From	Center determined (Example: Yes)	
Allow Import To	Center determined (Example: Yes)	

Here is a configuration template for a key group.

Table 14-21. Key Group Configuration Template.

Attribute	Value	Notes
Key Group ID	Center determined (Example: “Default”)	
Key Group Description	Center determined (Example: “Default SAM key policy.”)	

Key Policy	Use defined center key policy(-ies)	

To complete configuration, all tape drives in the center environment would be assigned to the single key group.

Note that should centers require more complex configurations, multiple key groups may be created and tape drives may be assigned to multiple key groups.

14.6 SNMP Configuration

KMAs can be configured to send SNMP Informs to center SNMP Managers.

Here is a configuration template for defining SNMP managers.

Table 14-22. SNMP Manager Configuration Template.

Attribute	Value	Notes
SNMP Manager ID	Center determined	KMS internal identifier for SNMP manager
SNMP Manager Description	Center determined	KMS internal description for SNMP manager
Network Address	Center determined	
Enabled	Generally, yes	Whether sending information to Manager is enabled
User name	Center determined	SNMPv3 user name
Protocol Version	v3	Required in HPC environments

14.7 Key Transfer Configuration

Pairs of KMS clusters can be configured to allow export and exchange of keys. However, because the exchange of media across centers is not current practice nor expected to be future practice, configuration of KMS clusters for key transfer to other clusters is not being recommended. By establishing multi-site clusters, the intra-cluster replication in KMS results in wide area replication of keys which provides a measure of protection against failure of all KMAs at a single site.

14.8 KMS Maintenance

14.8.1 Application Updates/Patches

The KMS Administration Guide addresses KMS upgrades. Here are some key considerations taken from the guide for KMS version 2.2.

1. KMA software upgrades are signed by Oracle and verified before they are applied.
2. A key split quorum is required after upgrade to validate the upgrade.
3. Each KMA can and should be upgraded in series to maintain availability of the cluster.

14.8.2 Backups/Restores

Two types of backups exist in KMS:

- Core Security backups

The primary element of the Core Security component is the Root Key Material. It is key material that is generated when a Cluster is initialized. The Root Key Material protects the Master Key. The Master Key is a symmetric key that protects the Data Unit (tape media) Keys stored on the KMA. Core Security is protected with a key split scheme that requires a quorum of users defined in the Key Split Credentials to provide their user names and passphrases to unwrap the Root Key Material.

- Database backups

These backups contain the actual keys from the KMS cluster that protect data on tape media.

Database backups protect against a situation where all the KMAs in a KMS Cluster are lost. As long as one KMA from a cluster is available, the cluster will continue to function. After the failure of a single KMA, a new KMA can be added to regain lost redundancy via replication from surviving KMAs. Backups are not needed. However, if *all* KMAs in a cluster are lost, however, a backup must be used to restore the key material contained in the cluster.

14.8.2.1 Proposed Backup Scheme

Core Security backups should be taken immediately after establishment of a KMS cluster and any subsequent change to the quorum users for a cluster.

Database backups should be taken periodically. To select a frequency, note that each KMA maintains a continually replenished (10 at a time) pool of pre-generated keys, at least 1000 in quantity. Upon their creation, these keys are *replicated* to peer KMAs before they can be put into service, providing a first layer of key protection. If the KMA that originated the key fails, its surviving peer can provide the key. Backups protect against failure of *all* KMAs in a cluster and need only be frequent enough to ensure that the all pre-generated keys are backed up prior to being put into use.

It is expected that daily KMS backups would be more than sufficient to protect the keys at the expected consumption rate of new keys. It is likely that weekly or monthly backups would suffice for most environments.

Database backups are initiated from the KMS management workstation, producing backup files that are saved to media via the management workstation. For both core security and database backups, it is recommended that backup instances are saved to both local file systems (for example, a USB stick) and off-site file systems (for example, a remote NFS share) in a center-approved manner.

For sizing purposes, one observed instance of a 500,000-key KMS backup consumed about 220MB of storage.

14.8.2.2 KMS Backup Security Notes

A database backup consists of two files -- the backup itself and a backup key file needed to unlock the backup. In addition to the two backup files, a core security backup must also be periodically created, and must be used to perform a restore. The backup file is encrypted using a key generated specifically for that backup. The key is placed in the backup key file, and is encrypted using a public key generated by the system, called the backup public key. The corresponding private key, called the backup private key, is contained in the core security backup file. The backup private key is encrypted using a master key. The core security backup file also contains the master key encrypted using the quorum information. When a restore is performed, the quorum information is used to decrypt the master key. Next, the master key is used to decrypt the backup private key, also from the core security backup file. The backup private key is used to decrypt the backup key file. Then the key from the backup key file is used to decrypt the backup. Without the core security backup file and the backup key file, the backup cannot be decrypted.

When a system is restored, the quorum information from the backup file becomes the quorum for the restored system.

A backup file and its corresponding backup key file must be used together. Any core security backup file can be used with any backup file/backup key file pair. When key split credentials are modified, a new core security backup file should be created. Old core security backup files should be destroyed, since the quorum that was in effect when the old file was created can still be used to do a restore, even of a backup that is made after the quorum is changed. This is intentional. It allows, for example, an old backup to be restored with a recent core security backup file in the event where the quorum from the time of the backup is unavailable. However, it does create a risk in that an old core security backup file can be used to restore a recent backup.

15 REFERENCES

1. RFC 4556, “Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)”, June 2006
2. HPCMP PKINIT Users Guide, dated March 17, 2009.
3. RFC 3961, “Encryption and Checksum Specifications for Kerberos 5”, February 2005.
4. RFC 4120, “The Kerberos Network Authentication Service (V5)”, June 2005.
5. “SUN StorageTek KMS 2.0 Technical Overview,”
6. Nirvana SRB Administration Guide, 2012 R3 (V.4.3.00)
7. Nirvana SRB Installation Guide, 2012 R3 (V.4.3.00)
8. Nirvana SRB Software Development Kit, 2012 R3 (V.4.3.00)
9. Nirvana SRB User Guide, 2012 R3 (V.4.3.00)

APPENDIX A - SOFTWARE BASELINE – APRIL 19, 2012

Table A-1. Software Baseline.

Application	Version	Documentation
Solaris	10 Update 10	
Nirvana SRB 2012 R3	4.3.00	
Oracle Database for Solaris Operating System (x86-64)	11g Release 2 (11.2.0.3.0)	
Oracle Database Grid Infrastructure for Solaris Operating System (x86-64)	11g Release 2 (11.2.0.3.0)	
Oracle De-install Utility for Solaris Operating System (x86-64)	11.2.0.3.0	
Sun One Studio	12 (C and C++ 5.9)	Certified Compilers for OCCI [Oracle Metalink ID 437957.1]
Java Development Kit (JDK)	6 Update 31	
SAM-QFS	5.2	
libcurl	7.22.0	
libical	0.46	
libiconv	1.13.1	
libz	1.2.5	
HPCMP Kerberos	Rel. 2011-12-25	
OpenSSH	1.2.7	
OpenSSL	0.9.8q	
ILOM (firmware)	ILOM 3.0.6.10	http://www.oracle.com/technetwork/systems/patches/firmware/release-history-jsp-138416.html#X4170
Emulex (firmware)	2.00a3 (U3D2.00A3)	

APPENDIX B - BILL OF MATERIALS

Table B-1. Bill of Materials (Standard Build).

Assembly Level					Vendor	Mdl or Part	Description	Assembly Qty		
1	2	3	4					GA Supplied	LM Supple-mental	Total
X						Standard Node	MCAT use, top-to-bottom nodes 1 & 2; Database use, top-to-bottom node 3	3		3
	X				Oracle	X4270-S1-AA	Sun Fire X4270 x64 Server: 2.5-inch HDD base chassis package including motherboard, no DVD, 1 x PSU, redundant fans and service processor for factory integration. RoHS-6.	1		1
		X			Oracle	5894A	Solaris 10 pre-installation for Sun Fire X4270 server. For factory integration only.	1		1
		X			Oracle	XSR-JUMP-1MC13	Single Jumper Cable 1 meter (C13 plug).	2		2
		X			Oracle	5870A	4GB Memory Kit DDR3-1333 registered ECC DIMMS (1x4GB) for Sun Fire and X4270. RoHS-6. Factory Integration Only.	16	-4	12
		X			Oracle	RA-SS2CF-300G10K	300GB 10K RPM 2.5" SAS hard disk drive with Marlin bracket. RoHS-6.	2	4	6
		X			Oracle	SG-PCIESAS-R-INT-Z	Sun StorageTek 8-port internal SAS RAID Host Bus Adapter with RAID 0,1, 1E, 10, 5, 5EE, 50, 6, 60 support, 256 MB of onboard memory with 72 hour battery backed write cache. RoHS 6. XATO.	1		1
		X			Oracle	SG-XPCIE2FC-EM8-Z	Sun StorageTek PCI-E Enterprise 8Gb FC host bus adapter, Dual Port, Emulex, includes standard and low profile brackets. RoHS 6 compliant.	2		2
		X			Oracle	X4446A-Z	Sun x4 PCI Express Quad Gigabit Ethernet UTP low profile adapter, low profile bracket on board, standard bracket included. RoHS-6 compliant, IntelOEM card	1		1
		X			Oracle	5861A	1 Intel Xeon Model Number X5570 Quad-Core (2.93GHz/95W) Processor without Heatsink for Sun Fire X4270 Servers. RoHS-6. For Factory Integration Only.	2		2
		X			Oracle	6328A	Redundant hot-swappable AC 1050W power supply unit for Sun Fire X4270 Server. RoHS-5.	1		1

Assembly Level					Vendor	Mdl or Part	Description	Assembly Qty		
1	2	3	4					GA Supplied	LM Supple-mental	Total
		X			Oracle	6331A	Drive bay filler panel for Sun Fire X4270 server. RoHS-6. XATO.	14	-4	10
		X			Oracle	X6326A	Slide rail kit for Sun Fire X4140/X4170/X4270/X44 40/ X4150/X4450 server. Only fits in Sun Rack 938, Sun Rack 1038, Sun Rack 1042, or racks that have front to rear rail spacing between 610mm and 915mm (about 24" to about 36"). RoHS-6. X-Option.	1		1
		X			Oracle	5899A	CPU Heatsink for Sun Fire X4270 server. For Factory Installation Only. RoHS-6.	2		2
		X			Oracle	8325A	DVD+/-RW SATA-based drive for Sun Fire X4170/X4270 x64 servers. RoHS-5. X-Option.	1		1
		X			Oracle	5879A	Memory filler panel for Sun Fire X4270 server. RoHS-6. XATO.	2		2
		X			Oracle	X6000A	Cryptographic Accelerator		1	1
X						Additional Disk Node	Database use, top-to-bottom node 4	1		1
	X				Oracle	X4270-S1-AA	Sun Fire X4270 x64 Server: 2.5-inch HDD base chassis package including motherboard, no DVD, 1 x PSU, redundant fans and service processor for factory integration. RoHS-6.	1		1
		X			Oracle	5894A	Solaris 10 pre-installation for Sun Fire X4270 server. For factory integration only.	1		1
		X			Oracle	XSR-JUMP-1MC13	Single Jumper Cable 1 meter (C13 plug).	2		2
		X			Oracle	5870A	4GB Memory Kit DDR3-1333 registered ECC DIMMS (1x4GB) for Sun Fire and X4270. RoHS-6. Factory Integration Only.	16	-4	12
		X			Oracle	RA-SS2CF-300G10K	300GB 10K RPM 2.5" SAS hard disk drive with Marlin bracket. RoHS-6.	2	6	8
		X			Oracle	SG-PCIESAS-R-INT-Z	Sun StorageTek 8-port internal SAS RAID Host Bus Adapter with RAID 0,1, 1E, 10, 5, 5EE, 50, 6, 60 support, 256 MB of onboard memory with 72 hour battery backed write cache. RoHS 6. XATO.	1		1
		X			Oracle	SG-XPCIE2FC-EM8-Z	Sun StorageTek PCI-E Enterprise 8Gb FC host bus adapter, Dual Port, Emulex, includes standard and low profile brackets. RoHS 6 compliant.	2		2

Assembly Level					Vendor	Mdl or Part	Description	Assembly Qty		
1	2	3	4					GA Supplied	LM Supple-mental	Total
		X			Oracle	X4446A-Z	Sun x4 PCI Express Quad Gigabit Ethernet UTP low profile adapter, low profile bracket on board, standard bracket included. RoHS-6 compliant, IntelOEM card	1		1
		X			Oracle	5861A	1 Intel Xeon Model Number X5570 Quad-Core (2.93GHz/95W) Processor without Heatsink for Sun Fire X4270 Servers. RoHS-6. For Factory Integration Only.	2		2
		X			Oracle	6328A	Redundant hot-swappable AC 1050W power supply unit for Sun Fire X4270 Server. RoHS-5.	1		1
		X			Oracle	6331A	Drive bay filler panel for Sun Fire X4270 server. RoHS-6. XATO.	14	-6	8
		X			Oracle	X6326A	Slide rail kit for Sun Fire X4140/X4170/X4270/X44 40/ X4150/X4450 server. Only fits in Sun Rack 938, Sun Rack 1038, Sun Rack 1042, or racks that have front to rear rail spacing between 610mm and 915mm (about 24" to about 36"). RoHS-6. X-Option.	1		1
		X			Oracle	5899A	CPU Heatsink for Sun Fire X4270 server. For Factory Installation Only. RoHS-6.	2		2
		X			Oracle	8325A	DVD+/-RW SATA-based drive for Sun Fire X4170/X4270 x64 servers. RoHS-5. X-Option.	1		1
		X			Oracle	5879A	Memory filler panel for Sun Fire X4270 server. RoHS-6. XATO.	2		2
		X			Oracle	X6000A	Cryptographic Accelerator		1	1
X						RAID Disk Array	Database use	1		1
	X				Oracle	TA6180R11A2-0	RoHS-5, Sun Storage 6180 array with 4GB cache and 8 * FC host ports, Rack-Ready Controller Tray – Diskless Chassis, 0GB, 0 drives; Includes: 2 * 2GB-cache memory FC RAID Controller cards, 2 * redundant AC power supplies and cooling fans, 2 * FC ports for expansion trays and 8 * 8 Gb/s host ports with shortwave SFPs, 2 * 5M fibre optic cables, 2 * 6M ethernet cables and management software, 3 yr on-site warranty included (For factory integration only)	1		1

Assembly Level					Vendor	Mdl or Part	Description	Assembly Qty		
1	2	3	4					GA Supplied	LM Supple-mental	Total
		X			Oracle	TC-FC1CF-300G15KZ	RoHS-6, Sun StorageTek (tm) 6140 array / CSM200, 300GB 15Krpm FC-AL drive	16	8	24
	X				Oracle	XTCCSM2R01A0J 4800Z	RoHS 5, Sun StorageTek (tm) CSM200, Rack Ready Expansion tray, 4800GB, 16 x 300GB 15Krpm 4GB FC-AL Drives, 2xI/O Modules, 2 x redundant AC power supplies and cooling fans, 2 x FC ports for expansions, 4 x shortwave SFPs with x 2 LC-LC FC cables. (Standard Configuration).	2		2
	X				Oracle	XTCCSM2R01A0J 1500Z	RoHS-5,Sun StorageTek (tm)CSM200, Rack-Ready Expansion Tray, 1500GB, 5 * 300GB 15Krpm 4Gb FC-AL Drives, 2 * I/O Modules, 2 * redundant AC power supplies & cooling fans, 2 * FC ports for expansions, 4 shortwave SFPs with 2 * LC-LC FC cables.	1		1
		X			Oracle	XTC-FC1CF-300G15KZ	RoHS-6, Sun StorageTek (tm) 6140 array / CSM200, 300GB 15Krpm FC-AL drive	3		3
	X				Oracle	XTCCSM2-RK-19UZ	RoHS 6, Sun Modular Storage : StorageTek (tm) 6140 / CSM200, Rack Rail Kit for standard 19-inch system cabinets and racks including the Sun Rack 900/1000 racks.	4		4
	X				Oracle	XSR-JUMP-1MC13	Single Jumper Cable 1 meter (C13 plug)	8		8
					Oracle		Internal tray-to-tray FC optical LC-LC cables	8		8
X						Network Switch	Private-side networks		2	2
	X				Cisco	3750X			1	1
		X			Cisco	WS-C3750X-24T-L	Catalyst 3750X 24-port Data LAN Base		1	1
		X			Cisco	S375XVK9T-12253SE	CAT 3750X IOS UNVESAL WITH WEB BASE DEV MGR		1	1
		X			Cisco	C3KX-PWR-350WAC/2	Catalyst 3K-X 350W AC Secondary Power Supply		1	1
		X			Cisco	CAB-STACK-50CM	Cisco StackWise 50 cm Stacking Cable		1	1
		X			Cisco	C3KX-PWR-350WAC	Catalyst 3K-X 350W AC Power Supply		1	1
		X			Cisco	C3KX-NM-BLANK	Catalyst 3K-X Network Module Blank		1	1
X						Network Switch	Public-side networks		2	2
	X				Cisco	4900M			1	1

Assembly Level					Vendor	Mdl or Part	Description	Assembly Qty		
1	2	3	4					GA Supplied	LM Supple-mental	Total
		X			Cisco	WS-C4900M	Base system with 8X2 ports and 2 half slots		1	1
		X			Cisco	S49MIPBK9-12246SG	Cisco CAT4900M IOS IP BASE SSH		1	1
		X			Cisco	WS-X4920-GB-RJ45	20-port 10/100/1000 RJ45		1	1
		X			Cisco	PWR-C49M-1000AC	4900M AC power supply, 1000W		1	1
		X			Cisco	PWR-C49M-1000AC/2	Redundant AC PS for 4900M		1	1
		X			Cisco	4900M-X2-CVR	X2 cover on 4900M		1	1
		X			Cisco		10GE cross-connect cable		1	1
		X			Cisco		X2 10GE SR		1	1
X						Equipment Rack	Equipment rack & infrastructure		1	1
	X				APC	AR3300	NetShelter SX 42U 600mmx1200mm 19-inch EIA equipment rack		1	1
		X			APC	AR8425A	1U Horizontal Cable Organizer		5	5
		X			APC	AR8442	Vertical Cable Organizer		2	2
		X			APC	AR8136BLK	1U 19-inch LCD Blanking Panel		1	1
		X			APC		Blanking Panels, 1U		22	22
	X					KVM system				
		X			Avocent	MPU4032DAC-001	32-port 4-path digital KVM switch dual power supply		1	1
			X		Avocent	ECS19UWRUSB-001	19-inch LCD pull-out tray keyboard, LCD monitor, trackball mouse		1	1
			X		Avocent	MPUIQ-VMC	Server Interface Module, Virtual & CAC, VGA & USB connection		4	4
			X		Avocent	MPUIQ-SRL	Serial Interface Module		10	10
			X		Avocent	UPD-AM	Power supply for 3 MPUIQ-SRL		4	4
	X					PDU system			2	2
		X			Avocent	PM3005V-404	PM3000 OU Vertical 3-phase, 60A, 208VAC PDU, IEC 60309 460C9W 4-wire plug		1	1
		X			Avocent	PMHD-THS	Combo Tempreture Hummidity Sensor		1	1
						Other Cables				
X						Various	CAT-6 Cu RJ45 Ethernet cables various lengths		58	58

Assembly Level					Vendor	Mdl or Part	Description	Assembly Qty		
1	2	3	4					GA Supplied	LM Supple-mental	Total
X						Various	50-micron LC-LC fibre optic cables, server nodes to RAID array controller		4	4
X						Various	IEC C13 female connector / IEC 60320 Sheet E plug 3-wire power jumper		26	26
X						Various	IEC 60320-1 C7 / IEC 60320 Sheet E plug 3-wire power jumper		4	4

Table B-2. Bill of Materials (ARL Build)

Assembly Level					Vendor	Mdl or Part	Description	Assembly Qty		
1	2	3	4					GA Supplied	LM Supple-mental	Total
X						Standard Node	MCAT use, top-to-bottom nodes 1 & 2; Database use, top-to-bottom node 3	3		3
	X				Oracle	X4270-S1-AA	Sun Fire X4270 x64 Server: 2.5-inch HDD base chassis package including motherboard, no DVD, 1 x PSU, redundant fans and service processor for factory integration. RoHS-6.	1		1
		X			Oracle	5894A	Solaris 10 pre-installation for Sun Fire X4270 server. For factory integration only.	1		1
		X			Oracle	XSR-JUMP-1MC13	Single Jumper Cable 1 meter (C13 plug).	2		2
		X			Oracle	5870A	4GB Memory Kit DDR3-1333 registered ECC DIMMS (1x4GB) for Sun Fire and X4270. RoHS-6. Factory Integration Only.	16	-4	12
		X			Oracle	RA-SS2CF-300G10K	300GB 10K RPM 2.5" SAS hard disk drive with Marlin bracket. RoHS-6.	2	6	8
		X			Oracle	SG-PCIESAS-R-INT-Z	Sun StorageTek 8-port internal SAS RAID Host Bus Adapter with RAID 0,1, 1E, 10, 5, 5EE, 50, 6, 60 support, 256 MB of onboard memory with 72 hour battery backed write cache. RoHS 6. XATO.	1		1
		X			Oracle	SG-XPCIE2FC-EM8-Z	Sun StorageTek PCI-E Enterprise 8Gb FC host bus adapter, Dual Port, Emulex, includes standard and low profile brackets. RoHS 6 compliant.	2		2
		X			Oracle	X4446A-Z	Sun x4 PCI Express Quad Gigabit Ethernet UTP low profile adapter, low profile bracket on board, standard bracket included. RoHS-6 compliant, IntelOEM card	1		1
		X			Oracle	5861A	1 Intel Xeon Model Number X5570 Quad-Core (2.93GHz/95W) Processor without Heatsink for Sun Fire X4270 Servers. RoHS-6. For Factory Integration Only.	2		2
		X			Oracle	6328A	Redundant hot-swappable AC 1050W power supply unit for Sun Fire X4270 Server. RoHS-5.	1		1
		X			Oracle	6331A	Drive bay filler panel for Sun Fire X4270 server. RoHS-6. XATO.	14	-6	8

		X			Oracle	X6326A	Slide rail kit for Sun Fire X4140/X4170/X4270/X44 40/ X4150/X4450 server. Only fits in Sun Rack 938, Sun Rack 1038, Sun Rack 1042, or racks that have front to rear rail spacing between 610mm and 915mm (about 24" to about 36"). RoHS-6. X-Option.	1		1
		X			Oracle	5899A	CPU Heatsink for Sun Fire X4270 server. For Factory Installation Only. RoHS-6.	2		2
		X			Oracle	8325A	DVD+/-RW SATA-based drive for Sun Fire X4170/X4270 x64 servers. RoHS-5. X-Option.	1		1
		X			Oracle	5879A	Memory filler panel for Sun Fire X4270 server. RoHS-6. XATO.	2		2
		X			Oracle	X6000A	Cryptographic Accelerator		1	1
X						Additional Disk Node	Database use, top-to-bottom node 4	1		1
	X				Oracle	X4270-S1-AA	Sun Fire X4270 x64 Server: 2.5-inch HDD base chassis package including motherboard, no DVD, 1 x PSU, redundant fans and service processor for factory integration. RoHS-6.	1		1
		X			Oracle	5894A	Solaris 10 pre-installation for Sun Fire X4270 server. For factory integration only.	1		1
		X			Oracle	XSR-JUMP-1MC13	Single Jumper Cable 1 meter (C13 plug).	2		2
		X			Oracle	5870A	4GB Memory Kit DDR3-1333 registered ECC DIMMS (1x4GB) for Sun Fire and X4270. RoHS-6. Factory Integration Only.	16	-4	12
		X			Oracle	RA-SS2CF-300G10K	300GB 10K RPM 2.5" SAS hard disk drive with Marlin bracket. RoHS-6.	2	8	10
		X			Oracle	SG-PCIESAS-R-INT-Z	Sun StorageTek 8-port internal SAS RAID Host Bus Adapter with RAID 0,1, 1E, 10, 5, 5EE, 50, 6, 60 support, 256 MB of onboard memory with 72 hour battery backed write cache. RoHS 6. XATO.	1		1
		X			Oracle	SG-XPCIE2FC-EM8-Z	Sun StorageTek PCI-E Enterprise 8Gb FC host bus adapter, Dual Port, Emulex, includes standard and low profile brackets. RoHS 6 compliant.	2		2
		X			Oracle	X4446A-Z	Sun x4 PCI Express Quad Gigabit Ethernet UTP low profile adapter, low profile bracket on board, standard bracket included. RoHS-6 compliant, IntelOEM card	1		1
		X			Oracle	5861A	1 Intel Xeon Model Number X5570 Quad-Core (2.93GHz/95W) Processor without Heatsink for Sun Fire X4270 Servers.	2		2

							RoHS-6. For Factory Integration Only.			
		X			Oracle	6328A	Redundant hot-swappable AC 1050W power supply unit for Sun Fire X4270 Server. RoHS-5.	1		1
		X			Oracle	6331A	Drive bay filler panel for Sun Fire X4270 server. RoHS-6. XATO.	14	-8	6
		X			Oracle	X6326A	Slide rail kit for Sun Fire X4140/X4170/X4270/X44 40/ X4150/X4450 server. Only fits in Sun Rack 938, Sun Rack 1038, Sun Rack 1042, or racks that have front to rear rail spacing between 610mm and 915mm (about 24" to about 36"). RoHS-6. X-Option.	1		1
		X			Oracle	5899A	CPU Heatsink for Sun Fire X4270 server. For Factory Installation Only. RoHS-6.	2		2
		X			Oracle	8325A	DVD+/-RW SATA-based drive for Sun Fire X4170/X4270 x64 servers. RoHS-5. X-Option.	1		1
		X			Oracle	5879A	Memory filler panel for Sun Fire X4270 server. RoHS-6. XATO.	2		2
		X			Oracle	X6000A	Cryptographic Accelerator		1	1
X						RAID Disk Array	Database use	1		1
	X				Oracle	TA6180R11A2-0	RoHS-5, Sun Storage 6180 array with 4GB cache and 8 * FC host ports, Rack-Ready Controller Tray – Diskless Chassis, 0GB, 0 drives; Includes: 2 * 2GB-cache memory FC RAID Controller cards, 2 * redundant AC power supplies and cooling fans, 2 * FC ports for expansion trays and 8 * 8 Gb/s host ports with shortwave SFPs, 2 * 5M fibre optic cables, 2 * 6M ethernet cables and management software, 3 yr on-site warranty included (For factory integration only)	1		1
		X			Oracle	TC-FC1CF-300G15KZ	RoHS-6, Sun StorageTek (tm) 6140 array / CSM200, 300GB 15Krpm FC-AL drive	16	8	24
	X				Oracle	XTCCSM2R01A0J4800Z	RoHS 5, Sun StorageTek (tm) CSM200, Rack Ready Expansion tray, 4800GB, 16 x 300GB 15Krpm 4GB FC-AL Drives, 2xI/O Modules, 2 x redundant AC power supplies and cooling fans, 2 x FC ports for expansions, 4 x shortwave SFPs with x 2 LC-LC FC cables. (Standard Configuration).	2		2
	X				Oracle	XTCCSM2R01A0J1500Z	RoHS-5,Sun StorageTek (tm)CSM200, Rack-Ready Expansion Tray, 1500GB, 5 * 300GB 15Krpm 4Gb FC-AL Drives, 2 * I/O Modules, 2 * redundant AC power	1		1

							supplies & cooling fans, 2 * FC ports for expansions, 4 shortwave SFPs with 2 * LC-LC FC cables.			
		X			Oracle	XTC-FC1CF-300G15KZ	RoHS-6, Sun StorageTek (tm) 6140 array / CSM200, 300GB 15Krpm FC-AL drive	3		3
	X				Oracle	XTCCSM2-RK-19UZ	RoHS 6, Sun Modular Storage : StorageTek (tm) 6140 / CSM200, Rack Rail Kit for standard 19-inch system cabinets and racks including the Sun Rack 900/1000 racks.	4		4
	X				Oracle	XSR-JUMP-1MC13	Single Jumper Cable 1 meter (C13 plug)	8		8
					Oracle		Internal tray-to-tray FC optical LC-LC cables	8		8
X						Network Switch	Private-side networks		2	2
	X				Cisco	3750X			1	1
		X			Cisco	WS-C3750X-24T-L	Catalyst 3750X 24-port Data LAN Base		1	1
		X			Cisco	S375XVK9T-12253SE	CAT 3750X IOS UNVESAL WITH WEB BASE DEV MGR		1	1
		X			Cisco	C3KX-PWR-350WAC/2	Catalyst 3K-X 350W AC Secondary Power Supply		1	1
		X			Cisco	CAB-STACK-50CM	Cisco StackWise 50 cm Stacking Cable		1	1
		X			Cisco	C3KX-PWR-350WAC	Catalyst 3K-X 350W AC Power Supply		1	1
		X			Cisco	C3KX-NM-BLANK	Catalyst 3K-X Network Module Blank		1	1
X						Network Switch	Public-side networks		2	2
	X				Cisco	4900M			1	1
		X			Cisco	WS-C4900M	Base system with 8X2 ports and 2 half slots		1	1
		X			Cisco	S49MIPBK9-12246SG	Cisco CAT4900M IOS IP BASE SSH		1	1
		X			Cisco	WS-X4920-GB-RJ45	20-port 10/100/1000 RJ45		1	1
		X			Cisco	PWR-C49M-1000AC	4900M AC power supply, 1000W		1	1
		X			Cisco	PWR-C49M-1000AC/2	Redundant AC PS for 4900M		1	1
		X			Cisco	4900M-X2-CVR	X2 cover on 4900M		1	1
		X			Cisco		10GE cross-connect cable		1	1
		X			Cisco		X2 10GE SR		1	1
X						Equipment Rack	Equipment rack & infrastructure		1	1
	X				APC	AR3300	NetShelter SX 42U 600mmx1200mm 19-inch EIA equipment rack		1	1
		X			APC	AR8425A	1U Horizontal Cable Organizer		5	5
		X			APC	AR8442	Vertical Cable Organizer		2	2
		X			APC	AR8136BLK	1U 19-inch LCD Blanking Panel		1	1
		X			APC		Blanking Panels, 1U		22	22

	X					KVM system				
		X			Avocent	MPU4032DAC-001	32-port 4-path digital KVM switch dual power supply		1	1
			X		Avocent	ECS19UWRUSB-001	19-inch LCD pull-out tray keyboard, LCD monitor, trackball mouse		1	1
			X		Avocent	MPUIQ-VMC	Server Interface Module, Virtual & CAC, VGA & USB connection		4	4
			X		Avocent	MPUIQ-SRL	Serial Interface Module		10	10
			X		Avocent	UPD-AM	Power supply for 3 MPUIQ-SRL		4	4
	X					PDU system			2	2
		X			Avocent	PM3005V-404	PM3000 OU Vertical 3-phase, 60A, 208VAC PDU, IEC 60309 460C9W 4-wire plug		1	1
		X			Avocent	PMHD-THS	Combo Tempreture Hummidity Sensor		1	1
						Other Cables				
X						Various	CAT-6 Cu RJ45 Ethernet cables various lengths		58	58
X						Various	50-micron LC-LC fibre optic cables, server nodes to RAID array controller		4	4
X						Various	IEC C13 female connector / IEC 60320 Sheet E plug 3-wire power jumper		26	26
X						Various	IEC 60320-1 C7 / IEC 60320 Sheet E plug 3-wire power jumper		4	4

Table B-3. Bill of Materials (ORS Build)

Assembly Level					Vendor	Mdl or Part	Description	Assembly Qty		
1	2	3	4					GA Supplied	LM Supple-mental	Total
X						Standard Node	MCAT use, top-to-bottom nodes 1 & 2; Database use, top-to-bottom node 3	3		3
	X				Oracle	X4270-S1-AA	Sun Fire X4270 x64 Server: 2.5-inch HDD base chassis package including motherboard, no DVD, 1 x PSU, redundant fans and service processor for factory integration. RoHS-6.	1		1
		X			Oracle	5894A	Solaris 10 pre-installation for Sun Fire X4270 server. For factory integration only.	1		1
		X			Oracle	XSR-JUMP-1MC13	Single Jumper Cable 1 meter (C13 plug).	2		2
		X			Oracle	5870A	4GB Memory Kit DDR3-1333 registered ECC DIMMS (1x4GB) for Sun Fire and X4270. RoHS-6. Factory Integration Only.	16	-4	12
		X			Oracle	RA-SS2CF-300G10K	300GB 10K RPM 2.5" SAS hard disk drive with Marlin bracket. RoHS-6.	2	6	8
		X			Oracle	SG-PCIESAS-R-INT-Z	Sun StorageTek 8-port internal SAS RAID Host Bus Adapter with RAID 0,1, 1E, 10, 5, 5EE, 50, 6, 60 support, 256 MB of onboard memory with 72 hour battery backed write cache. RoHS 6. XATO.	1		1
		X			Oracle	SG-XPCIE2FC-EM8-Z	Sun StorageTek PCI-E Enterprise 8Gb FC host bus adapter, Dual Port, Emulex, includes standard and low profile brackets. RoHS 6 compliant.	2		2
		X			Oracle	X4446A-Z	Sun x4 PCI Express Quad Gigabit Ethernet UTP low profile adapter, low profile bracket on board, standard bracket included. RoHS-6 compliant, IntelOEM card	1		1
		X			Oracle	5861A	1 Intel Xeon Model Number X5570 Quad-Core (2.93GHz/95W) Processor without Heatsink for Sun Fire X4270 Servers. RoHS-6. For Factory Integration Only.	2		2
		X			Oracle	6328A	Redundant hot-swappable AC 1050W power supply unit for Sun Fire X4270 Server. RoHS-5.	1		1
		X			Oracle	6331A	Drive bay filler panel for Sun Fire X4270 server. RoHS-6. XATO.	14	-6	8

		X			Oracle	X6326A	Slide rail kit for Sun Fire X4140/X4170/X4270/X44 40/ X4150/X4450 server. Only fits in Sun Rack 938, Sun Rack 1038, Sun Rack 1042, or racks that have front to rear rail spacing between 610mm and 915mm (about 24" to about 36"). RoHS-6. X-Option.	1		1
		X			Oracle	5899A	J Heatsink for Sun Fire X4270 server. For Factory Installation Only. RoHS-6.	2		2
		X			Oracle	8325A	DVD+/-RW SATA-based drive for Sun Fire X4170/X4270 x64 servers. RoHS-5. X-Option.	1		1
		X			Oracle	5879A	Memory filler panel for Sun Fire X4270 server. RoHS-6. XATO.	2		2
		X			Oracle	X6000A	Cryptographic Accelerator		1	1
X						Additional Disk Node	Database use, top-to-bottom node 4	1		1
	X				Oracle	X4270-S1-AA	Sun Fire X4270 x64 Server: 2.5-inch HDD base chassis package including motherboard, no DVD, 1 x PSU, redundant fans and service processor for factory integration. RoHS-6.	1		1
		X			Oracle	5894A	Solaris 10 pre-installation for Sun Fire X4270 server. For factory integration only.	1		1
		X			Oracle	XSR-JUMP-1MC13	Single Jumper Cable 1 meter (C13 plug).	2		2
		X			Oracle	5870A	4GB Memory Kit DDR3-1333 registered ECC DIMMS (1x4GB) for Sun Fire and X4270. RoHS-6. Factory Integration Only.	16	-4	12
		X			Oracle	RA-SS2CF-300G10K	300GB 10K RPM 2.5" SAS hard disk drive with Marlin bracket. RoHS-6.	2	8	10
		X			Oracle	SG-PCIESAS-R-INT-Z	Sun StorageTek 8-port internal SAS RAID Host Bus Adapter with RAID 0,1, 1E, 10, 5, 5EE, 50, 6, 60 support, 256 MB of onboard memory with 72 hour battery backed write cache. RoHS 6. XATO.	1		1
		X			Oracle	SG-XPCIE2FC-EM8-Z	Sun StorageTek PCI-E Enterprise 8Gb FC host bus adapter, Dual Port, Emulex, includes standard and low profile brackets. RoHS 6 compliant.	2		2
		X			Oracle	X4446A-Z	Sun x4 PCI Express Quad Gigabit Ethernet UTP low profile adapter, low profile bracket on board, standard bracket included. RoHS-6 compliant, IntelOEM card	1		1
		X			Oracle	5861A	1 Intel Xeon Model Number X5570 Quad-Core (2.93GHz/95W) Processor without Heatsink for Sun Fire X4270 Servers.	2		2

							RoHS-6. For Factory Integration Only.			
		X			Oracle	6328A	Redundant hot-swappable AC 1050W power supply unit for Sun Fire X4270 Server. RoHS-5.	1		1
		X			Oracle	6331A	Drive bay filler panel for Sun Fire X4270 server. RoHS-6. XATO.	14	-8	6
		X			Oracle	X6326A	Slide rail kit for Sun Fire X4140/X4170/X4270/X44 40/ X4150/X4450 server. Only fits in Sun Rack 938, Sun Rack 1038, Sun Rack 1042, or racks that have front to rear rail spacing between 610mm and 915mm (about 24" to about 36"). RoHS-6. X-Option.	1		1
		X			Oracle	5899A	CPU Heatsink for Sun Fire X4270 server. For Factory Installation Only. RoHS-6.	2		2
		X			Oracle	8325A	DVD+/-RW SATA-based drive for Sun Fire X4170/X4270 x64 servers. RoHS-5. X-Option.	1		1
		X			Oracle	5879A	Memory filler panel for Sun Fire X4270 server. RoHS-6. XATO.	2		2
		X			Oracle	X6000A	Cryptographic Accelerator		1	1
X						RAID Disk Array	Database use	1		1
	X				Oracle	TA6180R11A2-0	RoHS-5, Sun Storage 6180 array with 4GB cache and 8 * FC host ports, Rack-Ready Controller Tray – Diskless Chassis, 0GB, 0 drives; Includes: 2 * 2GB-cache memory FC RAID Controller cards, 2 * redundant AC power supplies and cooling fans, 2 * FC ports for expansion trays and 8 * 8 Gb/s host ports with shortwave SFPs, 2 * 5M fibre optic cables, 2 * 6M ethernet cables and management software, 3 yr on-site warranty included (For factory integration only)	1		1
		X			Oracle	TC-FC1CF-300G15KZ	RoHS-6, Sun StorageTek (tm) 6140 array / CSM200, 300GB 15Krpm FC-AL drive	16	8	24
	X				Oracle	XTCCSM2R01A0J4800Z	RoHS 5, Sun StorageTek (tm) CSM200, Rack Ready Expansion tray, 4800GB, 16 x 300GB 15Krpm 4GB FC-AL Drives, 2xI/O Modules, 2 x redundant AC power supplies and cooling fans, 2 x FC ports for expansions, 4 x shortwave SFPs with x 2 LC-LC FC cables. (Standard Configuration).	2		2
	X				Oracle	XTCCSM2R01A0J1500Z	RoHS-5,Sun StorageTek (tm)CSM200, Rack-Ready Expansion Tray, 1500GB, 5 * 300GB 15Krpm 4Gb FC-AL Drives, 2 * I/O Modules, 2 * redundant AC power	1		1

							supplies & cooling fans, 2 * FC ports for expansions, 4 shortwave SFPs with 2 * LC-LC FC cables.			
		X			Oracle	XTC-FC1CF-300G15KZ	RoHS-6, Sun StorageTek (tm) 6140 array / CSM200, 300GB 15Krpm FC-AL drive	3		3
	X				Oracle	XTCCSM2-RK-19UZ	RoHS 6, Sun Modular Storage : StorageTek (tm) 6140 / CSM200, Rack Rail Kit for standard 19-inch system cabinets and racks including the Sun Rack 900/1000 racks.	4		4
	X				Oracle	XSR-JUMP-1MC13	Single Jumper Cable 1 meter (C13 plug)	8		8
					Oracle		Internal tray-to-tray FC optical LC-LC cables	8		8
X						Network Switch	Private-side networks		2	2
	X				Cisco	3750X			1	1
		X			Cisco	WS-C3750X-24T-L	Catalyst 3750X 24-port Data LAN Base		1	1
		X			Cisco	S375XVK9T-12253SE	CAT 3750X IOS UNVESAL WITH WEB BASE DEV MGR		1	1
		X			Cisco	C3KX-PWR-350WAC/2	Catalyst 3K-X 350W AC Secondary Power Supply		1	1
		X			Cisco	CAB-STACK-50CM	Cisco StackWise 50 cm Stacking Cable		1	1
		X			Cisco	C3KX-PWR-350WAC	Catalyst 3K-X 350W AC Power Supply		1	1
		X			Cisco	C3KX-NM-BLANK	Catalyst 3K-X Network Module Blank		1	1
X						Network Switch	Public-side networks		2	2
	X				Cisco	4900M			1	1
		X			Cisco	WS-C4900M	Base system with 8X2 ports and 2 half slots		1	1
		X			Cisco	S49MIPBK9-12246SG	Cisco CAT4900M IOS IP BASE SSH		1	1
		X			Cisco	WS-X4920-GB-RJ45	20-port 10/100/1000 RJ45		1	1
		X			Cisco	PWR-C49M-1000AC	4900M AC power supply, 1000W		1	1
		X			Cisco	PWR-C49M-1000AC/2	Redundant AC PS for 4900M		1	1
		X			Cisco	4900M-X2-CVR	X2 cover on 4900M		1	1
		X			Cisco		10GE cross-connect cable		1	1
		X			Cisco		X2 10GE SR		1	1
X						Equipment Rack	Equipment rack & infrastructure		1	1
	X				APC	AR3300	NetShelter SX 42U 600mmx1200mm 19-inch EIA equipment rack		1	1
		X			APC	AR8425A	1U Horizontal Cable Organizer		5	5
		X			APC	AR8442	Vertical Cable Organizer		2	2
		X			APC	AR8136BLK	1U 19-inch LCD Blanking Panel		1	1
		X			APC		Blanking Panels, 1U		22	22

	X					KVM system				
		X			Avocent	MPU4032DAC-001	32-port 4-path digital KVM switch dual power supply		1	1
			X		Avocent	ECS19UWRUSB-001	19-inch LCD pull-out tray keyboard, LCD monitor, trackball mouse		1	1
			X		Avocent	MPUIQ-VMC	Server Interface Module, Virtual & CAC, VGA & USB connection		4	4
			X		Avocent	MPUIQ-SRL	Serial Interface Module		10	10
			X		Avocent	UPD-AM	Power supply for 3 MPUIQ-SRL		4	4
	X					PDU system			2	2
		X			Avocent	PM3005V-404	PM3000 OU Vertical 3-phase, 60A, 208VAC PDU, IEC 60309 460C9W 4-wire plug		1	1
		X			Avocent	PMHD-THS	Combo Tempreture Hummidity Sensor		1	1
						Other Cables				
X						Various	CAT-6 Cu RJ45 Ethernet cables various lengths		58	58
X						Various	50-micron LC-LC fibre optic cables, server nodes to RAID array controller		4	4
X						Various	IEC C13 female connector / IEC 60320 Sheet E plug 3-wire power jumper		26	26
X						Various	IEC 60320-1 C7 / IEC 60320 Sheet E plug 3-wire power jumper		4	4
X						Fibre Channel Switch			1	1
	X				Oracle	SGXSWBRO5100-8NS	Brocade 5100 40-port auto-sensing FC switch, 24-port activated, 24 SW SFP		1	1
		X			Oracle	SGXSWBRO5100-POD8	Brocade 5100 FC switch license key enabling an additional 8 ports. 8 SFP		1	1
		X			Oracle	SGXSWBRO3X50-RK-Z	Rackmount kit for the Brocade		1	1

APPENDIX C - MIGRATION PLAN

The original design for HPCMP's Storage Lifecycle Management included the concept of a Global Namespace spanning all DSRC sites. Such a Global Namespace allows users to uniquely name and distribute files across all of the DSRC sites and to have unified logical access to all of their files irrespective of the files' physical location. This is accomplished by fully replicating all the MCATs at all of the DSRCs utilizing Oracle Streams. Thus, this type of operation has been called Replicated Mode and it has always been assumed that production would commence in this mode.

Initial HPCMP production usage will not be in Replicated Mode, but instead, each DSRC's MCAT will operate independently with no communication between sites. Operating in this way has been termed Isolated Enclave Mode. What this means is that at the start of SLM production, Oracle Streams will not be activated, each MCAT will be unique, and there will not exist a Global Namespace.

However, the program realizes the potential long-term benefit to a Global Namespace. This appendix outlines ideas related to taking a production system operating in Isolated Enclave Mode and migrating it to fully Replicated Mode. These ideas have been discussed amongst the program's participants and with Oracle but they remain completely untested. Furthermore, no program member has ever performed such a transition. Since it is unknown if or when a migration will take place it makes the most sense to preserve the future capability of migrating to fully Replicated Mode.

Preserving a Migration Capability

The following necessary initial conditions have been identified at the onset of Isolated Enclave Mode production deployment so that a transition to Replicated Mode can be completed. These conditions are necessary but may not be sufficient.

- Users, classifications, and schemes are initially synchronized between sites.
- A central authority has to manage users, groups, domains, locations, classifications, schemes, and columns using SRB connections to each site;
    - Users, Groups, and Schemes must be centrally managed so that identical objects have the same IDs across sites;
    - Domains should be managed centrally so that sites do not add the same sub-domain with different IDs;
    - Locations are internally treated like users, so despite each site managing its own servers (i.e., Locations in SRB terminology) they need to be managed centrally as they are tied to the user IDs, which MUST be managed centrally to avoid duplication;

- Classifications must be centrally managed so that different sites don't associate different meanings with the same classification bit masks;

- There are two types of data: global and local. Global data is common among all sites by definition and hence must have the same data IDs globally (e.g., the /archive Collection). Local data is specific to a single site and hence must have globally unique data IDs (e.g., the /arl/others/scheder/test.txt file). Global data IDs need to be matched across all sites using a central authority. Local data IDs need to utilize the established offsets (per site, per node).
- Resource IDs need to utilize the established offsets (per site).

In conclusion, the databases among all sites must start identical with the exception of data ID and resource ID offsets.

The following table illustrates the possible local and global objects in Isolated Enclave Mode. Global objects need to be created using the –preferred_id argument in the respective registerXXX Acommands. The –preferred_id argument allows for specifying a globally unique ID managed by a central authority. All global objects should be created using the –preferred_id argument.

For example, to create a global user, the following syntax could be utilized:

```
registerUser –preferred_id 12345 scheder@HPCMP.HPC.MIL KERBEROS_AUTH "" staff
```

Table 1-1: Global vs. Local SRB Objects

Object	ID	Local	Global
User	USER_ID		X
Group	USER_ID		X
Domain	USER_ID		X
Location	USER_ID		X
Resource	RSRC_ID	X	
Scheme	SCHEME_ID		X
Column	SCHEME_ID		X
Collection	DATA_ID	X	X
Data Object	DATA_ID	X	X

Migration Plan from Isolated Enclave Mode to Replicated Mode

The steps and the accompanying figures illustrate the anticipated path to successfully transition the production HPCMP SLM environment from Isolated Enclave Mode to fully Replicated Mode. The migration utilizes for the MCATs, a hub and spoke architecture as a stepping-stone to n-way replication (Fig. C-1).

1. SLM production with each DSRC operating as its own unique identity. Six databases are shown for representative purposes only.
2. One of the DSRC MCATs is identified as the Hub Database. The remaining five databases export their information with a known System Change Number (SCN)

designated SCN1A to SCN5A. This exported information is then imported into the Hub Database.

3. Oracle Streams is activated using the exported SCN1A-SCN5A numbers from step 2 allowing the Hub Database to receive all updates.
4. The SLM functionality via SRB is stopped, the Oracle Streams finish replication, and SCN1B-SCN5B numbers of each of the databases are identified to allow for the final migration work. During this time, the SLM system is not available to the user community.
5. The Hub Database is exported and then imported into each of the other databases with a known SCN1C-SCN5C number.
6. Oracle Streams are activated again utilizing the required SCN values. This results in a full n-way replication of the MCAT databases allowing the SLM system to be reactivated.

N-Way Replication Versus Hub and Spoke

During the research into this migration plan Oracle personnel cautioned about deploying an n-way replication with 5 or 6 different locations. Specifically, the complexities about a production n-way replication with more than 3 locations are substantial and the benefits of such a deployment need to be weighed against these extra complications. Specifically with 6 locations, each database will have 5 captures and 5 propagations. In contrast, for the hub and spoke system, although it introduces some latency it is easier to manage since there is less overall data movement. Also, hub and spoke allows for easier addition and removal of sites. The hub and spoke does introduce a single point of failure but having redundant hardware for the hub may still be easier to manage than the n-way system.

Further work needs to be performed before a successful migration can be undertaken. First, a clearer understanding of the costs and benefits must be obtained. Then secondly, a list of associated changes to support the migration needs be created. For example, Stream setup and conflict resolution scripts will need to be redesigned during the migration from Isolated Enclave Mode to Replicated Mode regardless of whether n-way or hub and spoke are selected. However, the items listed above regarding the preservation of a migration capability apply equally to both configurations.

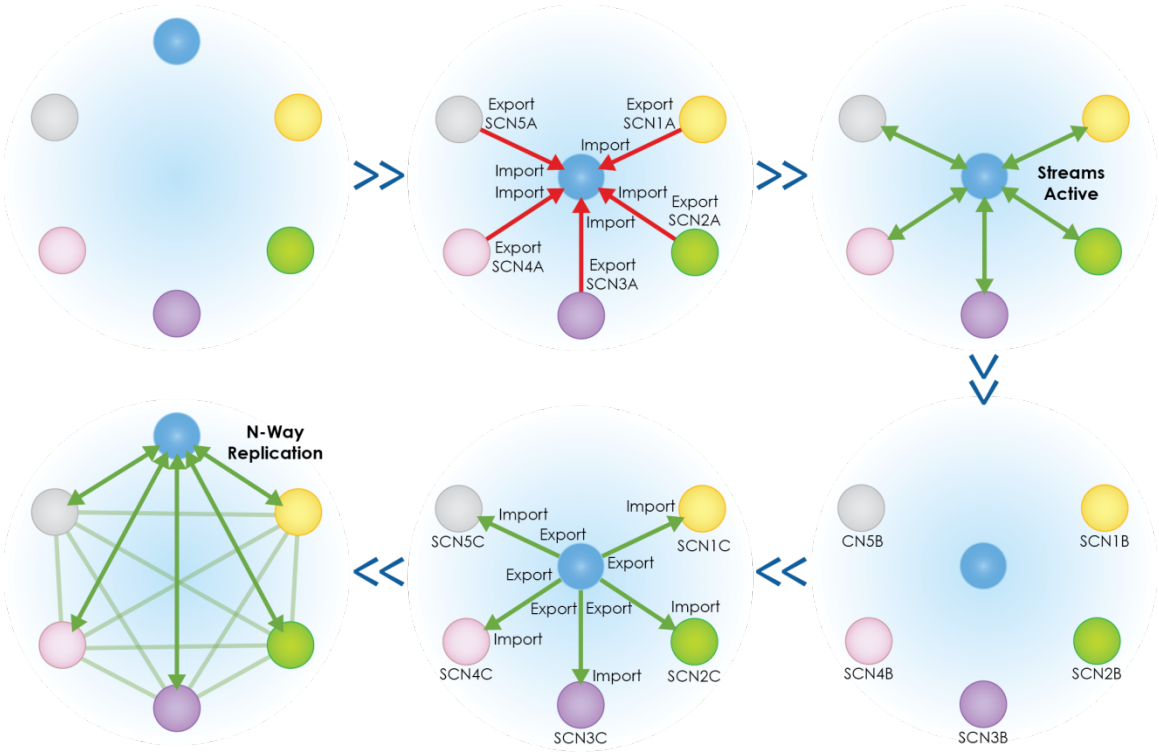


Figure C-1. Migration from Isolated Enclave Mode to Replicated Mode using an n-way model with the hub and spoke model as an intermediate stepping stone

P.O. BOX 85608 SAN DIEGO, CA 92186-5608 (858) 455-3000