



## **Storage and Infrastructure Work Package 9**

### **System Design Document**

**Version 0.4, 2/01/2007**

## **Design for Primary and Secondary Storage**

**Manager:** Russell Keil

**Prepared by:** Richard Spindler

#### **Document Control:**

<b>Date</b>	<b>Action</b>	<b>Author</b>	<b>Contact #</b>
01/12/2006	Original Paper	Richard Spindler	9905 9560
12/12/2006	Minor Edits	Russell Keil	9905 4782
02/01/2007	SRB/Formatting	Russell Keil	9905 4782

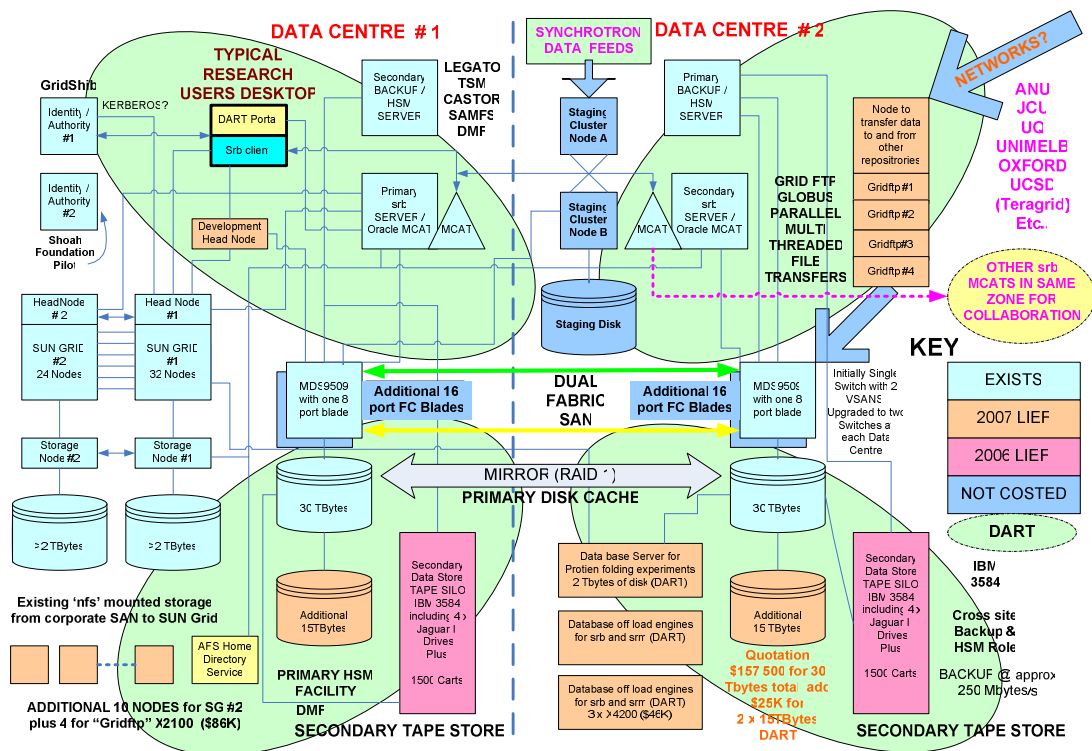


# Executive Summary

This document contains a design brief for the storage architecture to deliver the primary and secondary storage requirements for the DART project.

The goal is to design a storage infrastructure, which will be provisioned so the data can be in multiple locations via middleware such as SRB, and detail policies associated with the storage of that data.

The DART storage is to be delivered via funding from a number of sources. Below is a high-level block diagram of proposed dual data centre solution, DART software components are shown within the 'green' ellipses; embracing Identity Management, Security, Gridftp, Storage Resource Broker (SRB) and Hierarchical Storage Management (HSM).



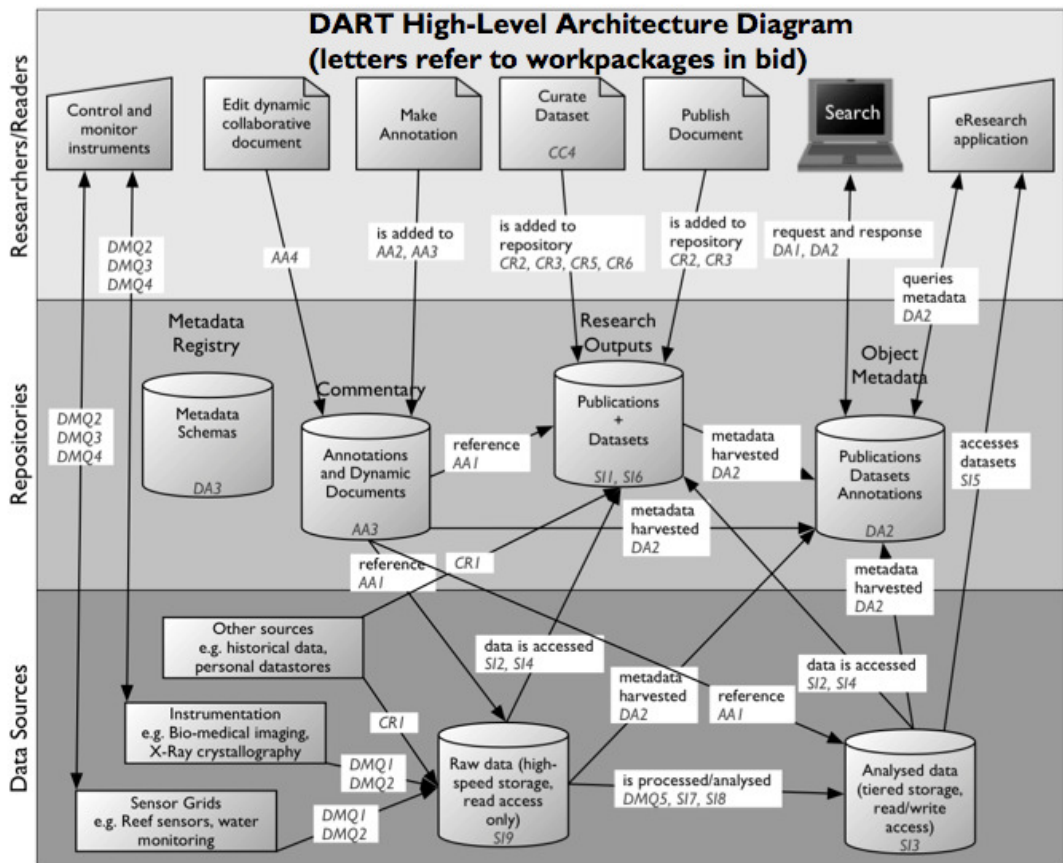
# Table of Contents:

1	Introduction.....	5
2	Customer Needs Analysis .....	7
2.1	Primary Storage.....	7
2.2	Secondary Storage.....	7
2.3	Technical Requirement .....	8
2.4	Performance Metrics .....	9
3	Proposed Architectural Design.....	11
3.1	Storage Resource Broker.....	11
3.2	Metadata Catalogue (MCAT) .....	12
3.3	Federation.....	13
3.4	SRB Zones: .....	15
3.5	Virtualisation.....	16
3.6	HSM (Hierarchical Storage Management) .....	16
3.6.1	DMF.....	17
3.6.2	SAMFS .....	18
3.6.3	CASTOR (CERN Advanced STORage manager) .....	18
3.7	Architecture Issues .....	20
3.7.1	Policy Considerations.....	20
4	Service Definition Statement .....	21
5	Recommendations .....	22
6	Terms of Reference.....	25
6.1	Glossary .....	25
6.2	References .....	25
7	Design Signoff .....	<b>Error! Bookmark not defined.</b>
8	Appendix A .....	26
8.1	Detailed technical documentation. ....	26

# 1 Introduction

The purpose of this document is to provide a design brief for implementing the primary and secondary storage requirements of DART.

The DART project is an ambitious proof-of-concept project to develop tools to support the new collaborative research infrastructure of the future. The project aims to enable researchers and reviewers to access original and analysed data, collaborate around the creation of research outputs, stored publications, plus add content, annotations and notes. It will also look at the collection of large datasets, including the remote control and automated data collection. [1].



More specifically a very large storage infrastructure will be required to store outputs from the various instruments and sensors in the identified research communities. This infrastructure will need to be built using a hierarchical storage management approach comprising fast online and relatively slower near-line storage systems. This work package will look at the engineering and configuration requirements for such a system, and implement a proof of concept using Monash's SAN and HSM infrastructure. The full production hardware will be sought via a separate LIEF grant in which Monash is a participant, and future NCRIS funding. [2].

## **2 Customer Needs Analysis**

As outlined in the executive summary and introduction, the customer needs are for a primary and secondary storage design. There is also a requirement for an analysis of the policies associated with access to such storage. Finally there is a need for a pilot instance of this design for testing purposes.

### **2.1 Primary Storage**

The primary storage as defined by DART is the storage of unprocessed or unannotated (raw) data such as the output of sensor equipment. Typically this would be implemented in the form of a disk cache sufficiently close to the data source and large enough to capture all the data generated in a sequence of related experiments. An example of this could be data travelling from an instrumentation system to a file system on a compute grid for further computation. Primary data can be both static (held on a DVD or CD ROM) and dynamic (available in real time). This data is likely to be initially more restricted (only accessible by the specific research teams) pending publication of scientific papers.

### **2.2 Secondary Storage**

The secondary storage as defined by DART is the storage of processed or annotated data, such as instrument data that has been processed by a computational grid. It would be anticipated that this data would most likely find itself being ingested into a storage resource broker (SRB) facility with the potential to be replicated at more than one storage repository in various geographic locations.

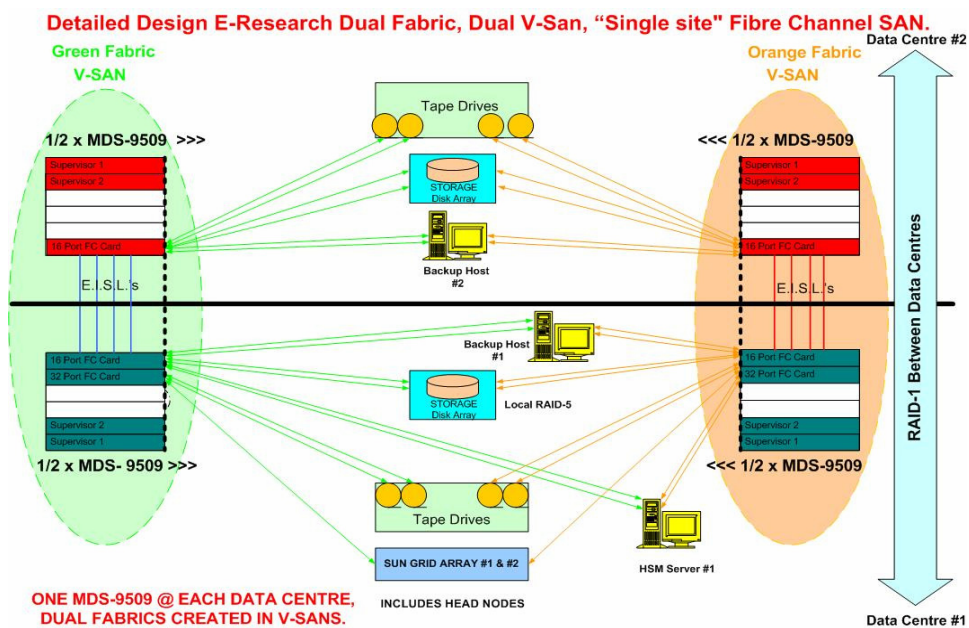
An example of this could be data that has been processed on a compute grid being annotated and stored for retrieval. It could also be data coming directly from an instrument that requires no computation, only annotation before storage. In some cases, the instrument may have already provided annotation of the data as part of the machine metadata. Data used in simulations of experiments or the creation of virtual instruments would fall into this category. It is likely that data in secondary storage will be held for a longer period and available for wider community collaboration.

It should be noted that data being held in the SRB is available to multiple readers searchable using a metadata catalogue for future annotation, some attention may need to be focused on version control, and the synchronization of replicas. If the annotation software does not support the checking out of files it may be necessary to add a CVS layer using "Subversion" or a similar product to ensure coherency of the data sets.

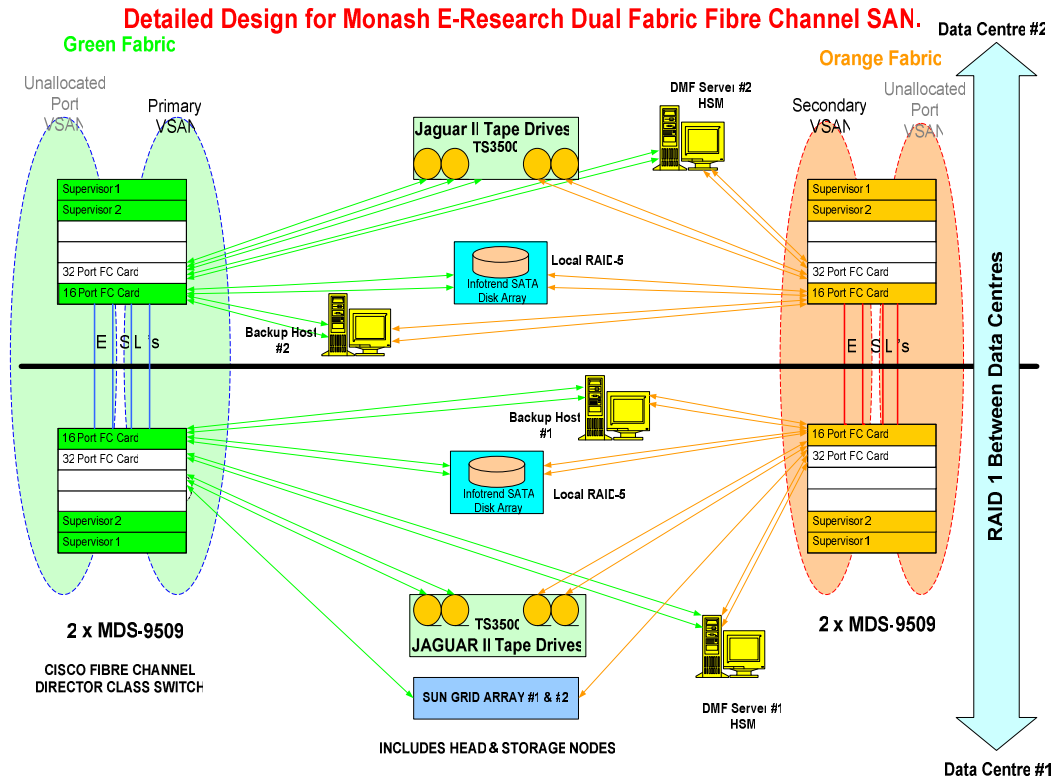
## 2.3 Technical Requirement

The ARC grant for strategic Infrastructure to was structured to provision around 40TBytes of SATA disk storage and about 80TBytes of near-line tape storage; this was to be via a \$300K grant with a \$150K contribution from Monash, providing a total of around \$500K. Monash being the Lead institution and grant recipient of the Data acquisition, accessibility, annotation, e-Research technology (DART) project, has agreed to match in kind infrastructure to the tune of \$532K to facilitate the discharge of the individual work packages assigned to the university.

The DART project will leverage this infrastructure through the SI9 work package to provide primary and secondary storage to the other work units. This will be a pilot that will also look at the HSM components underneath SRB to manage data on tiered storage.



Initial Design involved two MDS9509's running two Virtual Sans' (or fabrics) the plan being to progressively upgrade the design to a higher resilience model involving 4 fabrics in the future



This is the next stage in resilience of the solution involving duplication of the fundamental Fibre Channel switches at each of two data centres. The model can tolerate any single director class switch failing as all components are connected by dual paths one into each fabric. Again the mirroring of data between the two centres makes this a very highly available solution for the delivery of a quality production service.

## 2.4 Performance Metrics

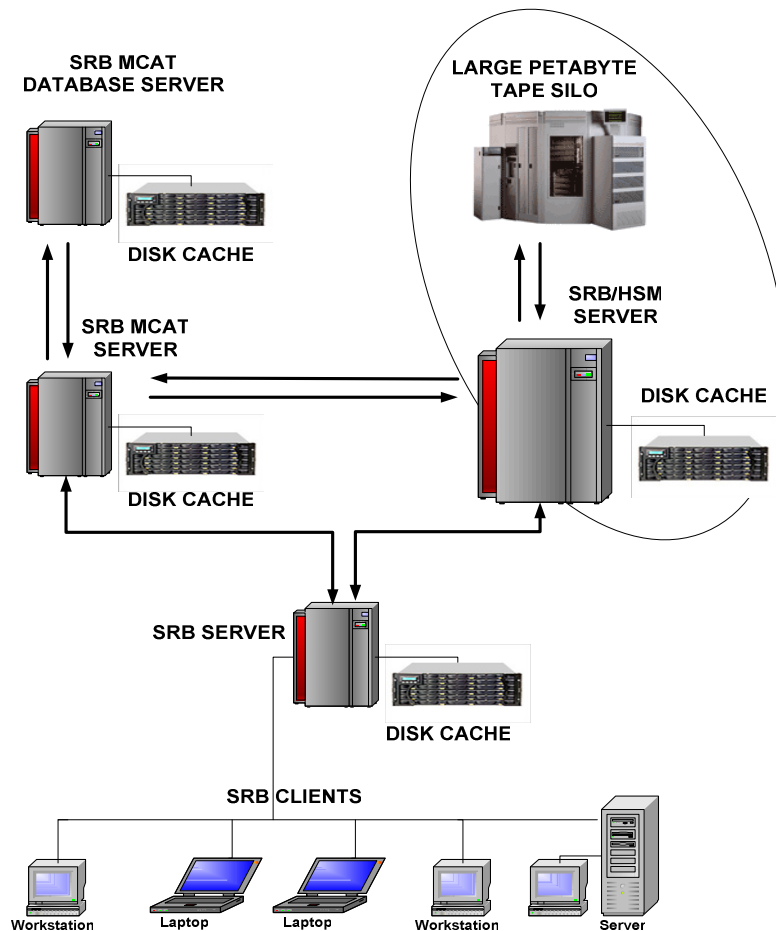
The choice of hardware infrastructure proposed is capable of reading and writing data in the order of 200 Mbytes/s for the disk controllers, the tape drives can read and write at a sustained work rate of around 75 Mbytes/s per drive. The network being 1 Gigabit Ethernet can move data at around 100 Mbytes/s, which can be improved using trunking. What tends to be the bottle neck in most implementations is the application software, it is seldom written to handle “double buffering” for the data movement and although a sequential write may be able to operate at 200 Mbytes/s peak during a block transfer, unless another write command is issued before the current one completes the average transfer rate falls away quite fast.

Whilst the authors are quite confident that the proposed design will meet the DART demonstrator requirements from a performance perspective, it may be necessary to engineer some additional threading (parallelism) into the final production release if larger data flows are being anticipated.

### 3 Proposed Architectural Design

The proposed architectural design uses SRB to federate storage. As discussed earlier SRB federates storage and insulates the user from the underlying storage. Thus SRB can be thought of as an abstraction layer that sits above the institutional storage layer, permitting institutions to architect different hardware and mass storage HSM policies. However we will look at various HSM systems and make recommendations.

The diagram below shows a possible arrangement of SRB servers and a data store underneath the infrastructure. SRB supports multiple types of storage, a large Data Store below could equally be a DMF, SAMFS or CASTOR based HSM system. The data store can also be different at different instances, as the intercommunication between data stores is handled by SRB.



**Typical SRB Architecture with large data store attached**

#### 3.1 Storage Resource Broker

Quoting from the SRB documentation. [3].

As the name implies the Storage Resource Broker (SRB), brokers storage resources. It provides access, via a uniform API, to various types of data storage across local and wide-area networks, and maintains meta-data (data about the data) about each stored object (files). SRB, in conjunction with MCAT, provides a means for accessing data objects and resources through querying their attributes instead of knowing their physical names and/or locations.

The SDSC Storage Resource Broker (SRB) provides the abstraction mechanisms needed to implement data grids, digital libraries, and persistent archives for data sharing, data publication, and data preservation.

Many people, using only a subset of the features, find that using the SRB as global file system is its most compelling function. Users of multiple distributed computing systems find it to be an essential tool to easily and quickly access files from various locations. With the SRB's parallel I/O capabilities, the SRB will transfer files at least as quickly as any other mechanism, and usually faster.

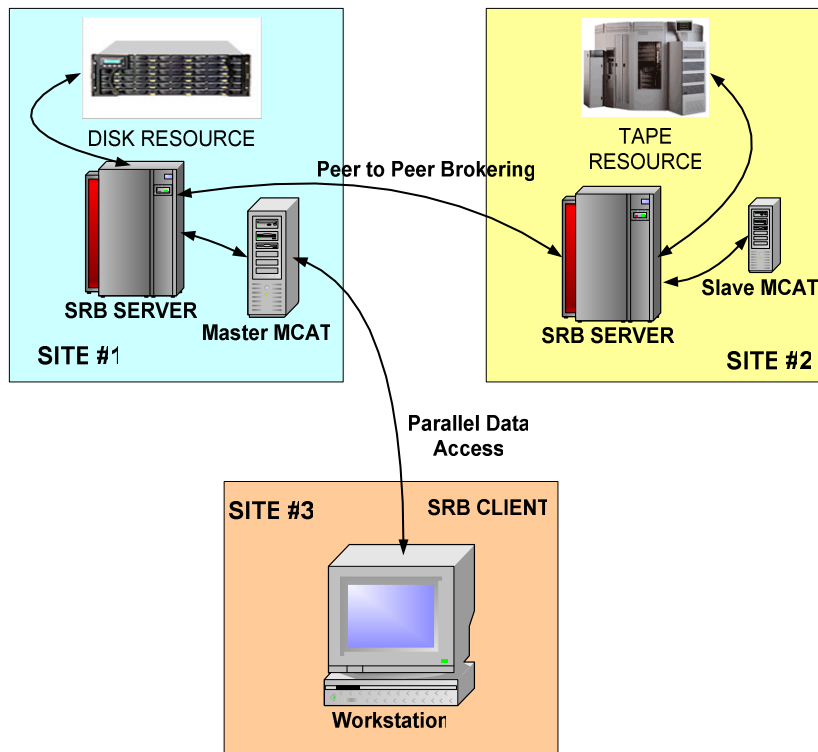
### **3.2 Metadata Catalogue (MCAT)**

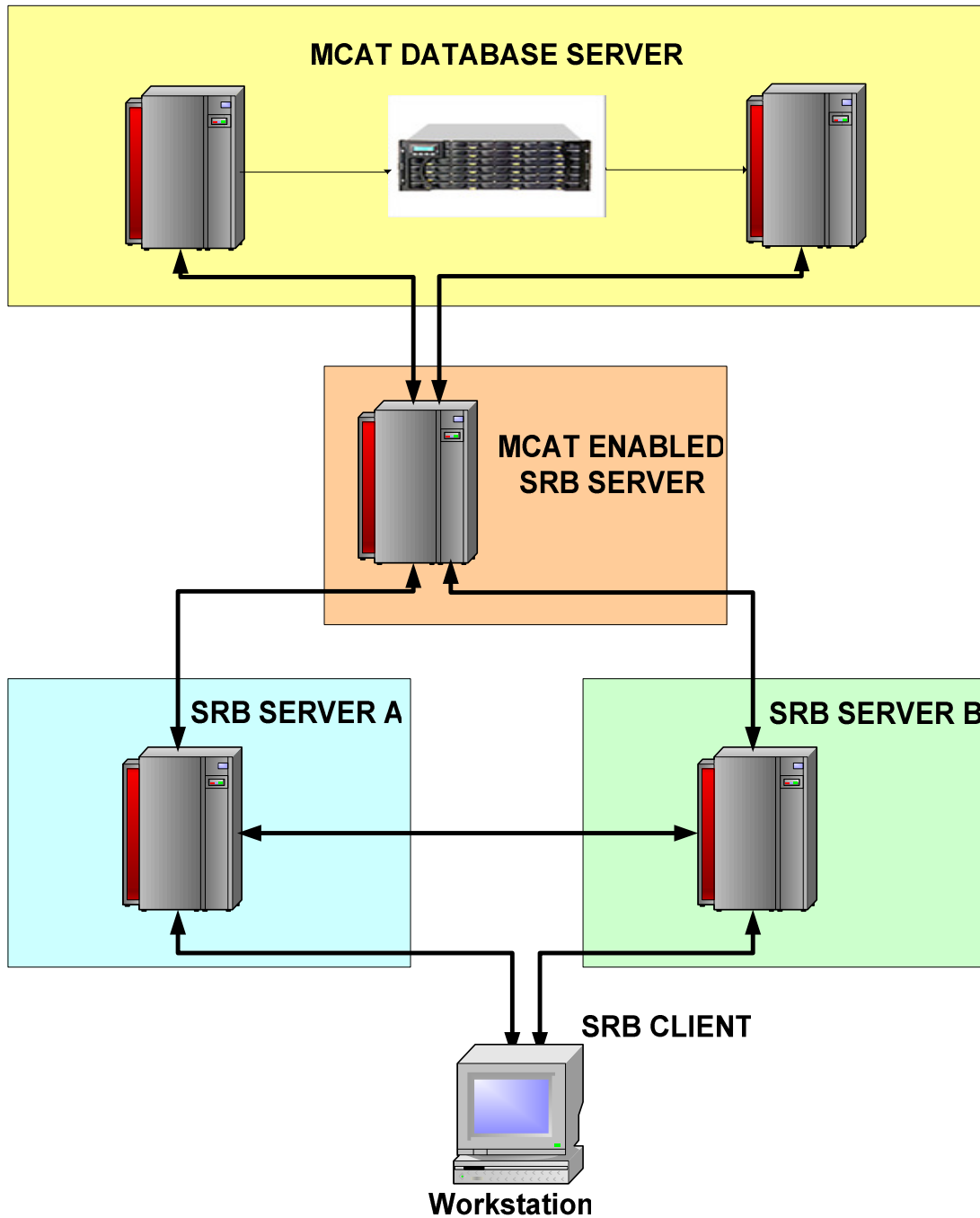
Quoting the SRB documentation [3]:

MCAT, or Meta data Catalog, is a meta data repository system implemented at SDSC to provide a mechanism for storing and querying system-level and domain-dependent meta data using a uniform interface. MCAT provides a resource and data object discovery mechanism that can be effectively used to identify and discover resources and data objects of interest using a combination of their characteristic attributes instead of their physical names and/or locations. In general the MCAT stores data about Users, Data Objects, Collections, Resources, Locations and Methods. It carries access control information and provides an audit trail of access to data sets.

### 3.3 Federation.

SRB has the capability to federate between different instances. These instances can be local or remote, for example between institutions. In the case of institutions, policies and agreements need to be made between the institutions in order for data to be exchanged. The diagram below shows an example of multiple SRB servers federated, and a single SRB client that is able to search data across the federated instances.





## Overview of SRB Components

Quoting the SRB documentation [4]. The motivations for a federated MCAT system is:

Improve MCAT WAN performance. In world-wide networks, the network latency would cause significant SRB performance degradation. For example, U.S. the East/West coast latency for a simple query is often 1–2 seconds. Many SRB operations require multiple MCAT interactions, compounding the delays.

Local control. Some SRB sites want to share resources and collections, yet maintain more local control over those resources, data objects, and collections. Rather than one SRB system managed by one administrator, they needed two (or more) cooperating SRB system managed locally, as primarily a security and authorization issue.

Scalability of the MCAT. For heavily loaded MCAT, distributing the load to multiple servers, MCATs, and DBMS 's will avoid bottlenecks and improved overall performance, particularly as the systems are scaled up.

No single point of failure. If site A and site B have a single MCAT at site B, then B's MCAT and server must be up and accessible for site A users to access data objects, even if those data objects are on a resource at site A. With a Zone a A and B, operations are more independent and locally controlled.

One or more “slave” MCAT’s can be added into an SRB Wide–Area–Network instance to reduce latency. SRB function calls that perform reads from the MCAT and not updates do not need to reference the Master, but can query the local slave MCAT.

Ref: SRB user manual release 3.4.0

### **3.4 SRB Zones:**

A Zone is defined as a federation of SRB resources/servers controlled by a single MCAT. Each zone has full control of its administrative domain and can operate independently from all other zones. A Federation of MCAT zones allows the sharing of SRB resources/servers between all zones in the federation. Users in one zone with the appropriate authentication and permissions can access data and resources in any other zone within the federation. The security model used ensures that a security compromise in one zone will not significantly impact the security of another zone in the federation.

For further information consult the following SDSC links:

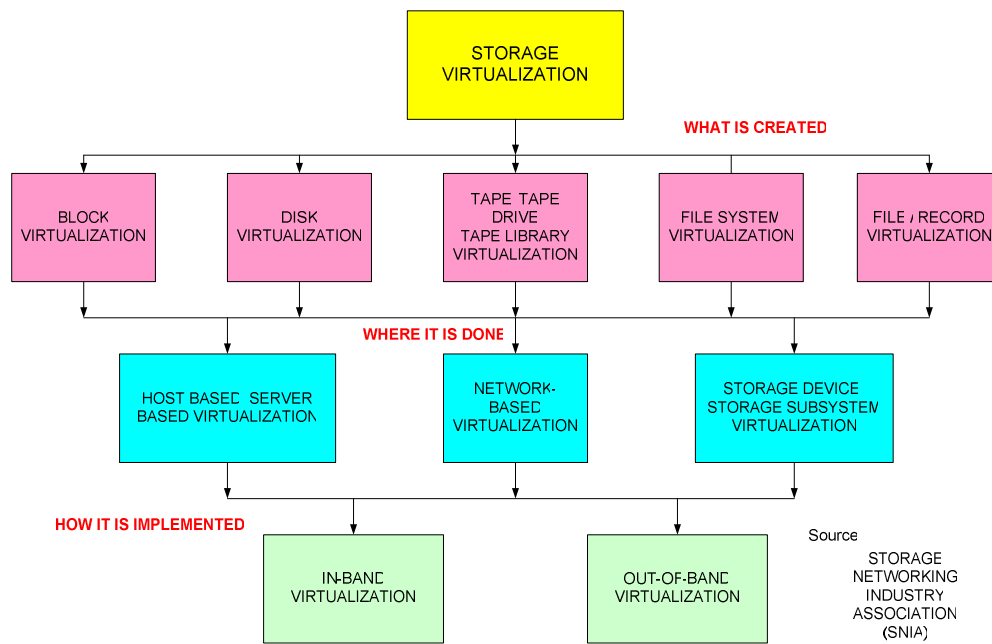
[http://www.sdsc.edu/srb/index.php/Fed\\_MCAT](http://www.sdsc.edu/srb/index.php/Fed_MCAT)

<http://www.sdsc.edu/srb/index.php/Zones>

[http://www.sdsc.edu/srb/index.php/Secure\\_Compressed\\_Data](http://www.sdsc.edu/srb/index.php/Secure_Compressed_Data)

### 3.5 Virtualisation.

As discussed SRB provides a way of virtualising storage and management, including HSM. Below is a diagram explaining the concept of storage virtualisation.



As with all forms of virtualisation there is overhead. Therefore SRB should be used primarily as secondary storage, except in the case of primary storage where the instrument provides annotated date, and that data requires no further reduction.

### 3.6 HSM (Hierarchical Storage Management)

HSM is a data storage system that automatically moves data between high-cost and low-cost storage media. HSM systems exist because high-speed storage devices, such as hard disk drives, are more expensive (per byte stored) than slower devices, such as optical discs and magnetic tape drives. While it would be ideal to have all data available on high-speed devices all the time, this is prohibitively expensive for many organizations. Instead, HSM systems store the bulk of the enterprise's data on slower devices, and then copy data to faster disk drives when needed. In effect, HSM turns the fast disk drives into caches for the slower mass storage devices. The HSM system monitors the way data is used and makes best guesses as to which data can safely be moved to slower devices and which data should stay on the hard disks. [4].

As stated before, SRB insulates us from the storage and HSM, providing a level of virtualisation. So providing the HSM software is supported by SRB it should interoperate with the institutions existing HSM system. If a HSM system is not already present at the institution, then things like current infrastructure may influence the decision of which HSM product to use. Ideally to reduce costs, if staff are available then open source alternatives like CASTOR can be considered.

### **3.6.1 DMF**

Quoting DMF documentation [5]:

DMF transparently migrates files from online storage to near-line storage based on user-defined criteria such as time of last access. Without DMF, the only choice most sites have is to either maintain a huge pool of inactive data on expensive disk storage or manually archive that data to tape, where it is difficult or impossible to access.

With DMF, a nearly infinite data store can be cost-effectively maintained and managed without sacrificing accessibility. All data is transparently recalled to primary storage when accessed, so users never need to know where the data resides. Many SGI customers are already using DMF to manage over a petabyte of data with no end in sight.

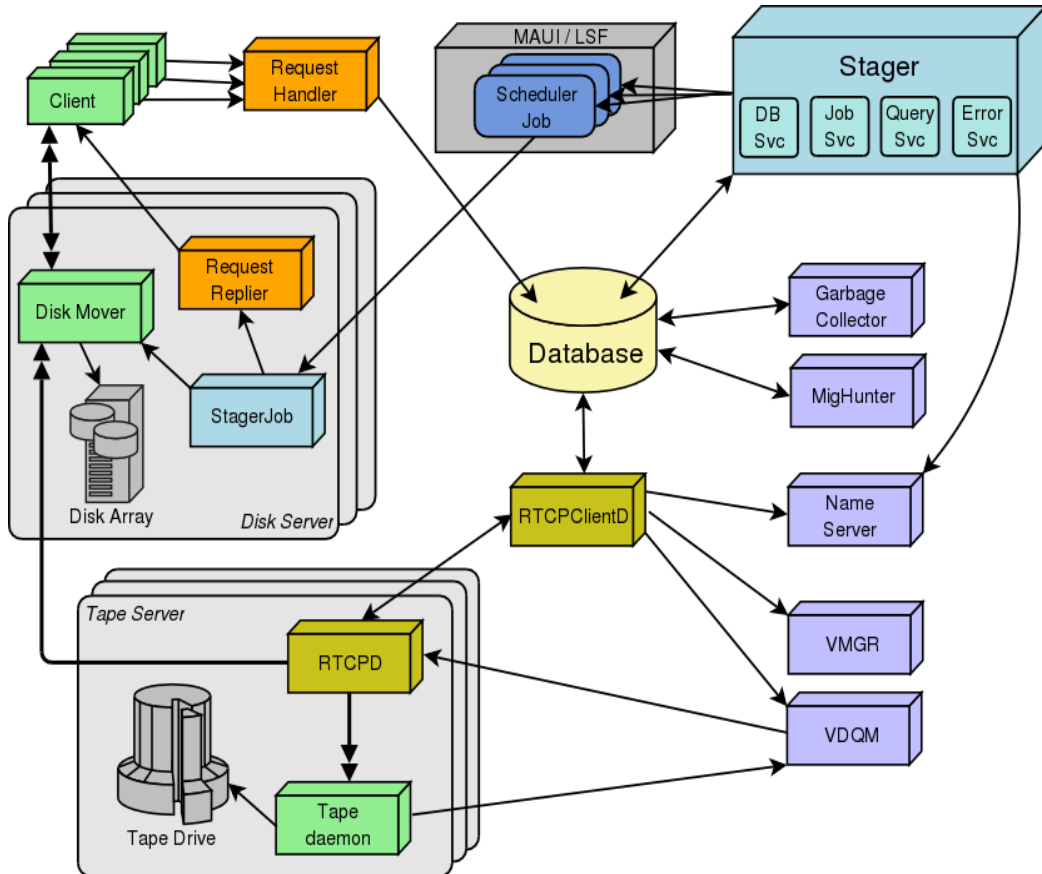
Hosted on either IRIX or Linux systems, DMF can also be used in conjunction with SGI InfiniteStorage NAS and SAN solutions to provide transparent data lifecycle management services across Windows®, Mac OS® X, Linux® and Unix® environments to transparently access data on multiple tiers of storage.

### 3.6.2 SAMFS

Quoting SAM-FS documentation [6]:

Sun StorageTek SAM-FS software (SAM-FS) provides data classification, centralized meta-data management, policy based data placement, protection, migration, long-term retention, and recovery to help organizations effectively manage and utilize data according to business requirements. SAM enables users to reduce the cost of storing vast data repositories by providing a powerful, easily managed, cost-effective way to access, retain, and protect business data over its entire lifecycle. This self-protecting file system offers continuous backup and fast recovery features to help enhance productivity and improve resource utilization.

### 3.6.3 CASTOR (CERN Advanced STORAGE manager)



Quoting CASTOR documentation [7]:

CASTOR2 possesses a modular design with a central database for information handling. The main functionality can be grouped in 5 modules, briefly introduced as it follows:

- The **Stager** has the primary role of a disk pool manager whose functions are to allocate space on disk to store a file, to maintain a catalogue of all the files in its disk pools and to clear out old or least recently used files in these pools when more free space is required. A disk pool is simply a collection of file systems – from one to many.
- The principal role of the **Name Server** is to implement a hierarchical view of the CASTOR name space so that it appears that the files are in directories. Names are made up of components (between //) and for each component, file access permissions, file size and access times are stored. The name server also remembers the file location on tertiary storage if the file has been migrated from the disk pool in order to make space for more current files. Files may be segmented or be made up of more than one contiguous chunk of tape media. This allows the use of the full capacity of tape volumes and permits file sizes to be bigger than the physical limit of a single tape volume. Additionally, it provides the ability to create directories and files, change ownership etc as specified in the posix standard.
- **Tapes**
- The **database** plays a central role in the CASTOR2 design. Its database schema is very complex and contains about sixty tables and several procedures and triggers. The functionality of the database is to store the actual status information of ongoing processes to have stateless components. Relation entity Diagram
- The **Client** allows you to interact with the server in order to get the basic functionality. You can get a file which is stored in the disk server or tape server, using RFIO, ROOT, GRIDFTP or XROOTD (coming soon). You also can check the status of a file or update it, as well as put a file. The client can be CASTOR client (command line mode or API) or SRM
- The **Storage Resource Management SRM** is a middleware component for managing shared storage resources on the GRID (enlace a grid). It provides dynamic space allocation on a file management and uniforms the access to heterogeneous storage elements, CASTOR is one of them.

## **3.7 Architecture Issues**

The final security authentication mechanism within the DART project has not been decided, but at the moment GSI is the preferred authentication mechanism as it is supported by many components of the DART architecture such as Globus and SRB. As such SRB needs to be able to store encrypted data, but not encrypt it. SRB needs to provide access controls so that access to data is limited to those users allowed to access that data. At the moment it is being designed to allow GSI to determine appropriate authentication and encryption.

Within the DART project there are many institutions with different hardware and operating environments, by using open source software it is hoped support will be available for as many platforms as possible.

Storage middleware such as SRB helps hide the difference in underlying storage and HSM systems between institutions.

### **3.7.1 Policy Considerations**

Some data involved in research may be subject to intellectual property or corporate ownership issues; as such data needs to be stored in a secure manner. This might result in separately zoned data or separate secure installation of storage components to meet the requirements of the client. This needs to be assessed on a case by case basis.

To co-locate, or mirror data between organizations, agreements need to be made between institutions housing data.

## 4 Service Definition Statement

This service is designed to be a pilot, and as such should not expect a production level of support. None the less the engineering involved in the solution should use production values in its design.

## 5 Recommendations

The recommendations of this work package can be split into software, hardware and policy recommendations.

- As discussed in the introduction the first main software recommendation of this work package is that SRB be used to insulate the users from the storage requirements. This recommendation helps reduce the number of hardware recommendations required as it insulates the user from the hardware by running on a variety of operating systems. As we will discuss, other software such as the HSM software will dictate more which operating system is chosen. Operating systems supported by SRB include Linux, Mac OS X, AIX, Solaris, SunOS, SGI Irix and Windows. Windows cannot be configured with an MCAT server.
- SRB also allows the federation of data, controlling its access from multiple locations. As such the second main software recommendation of this work package is that GSI be used as the authentication mechanism as SRB supports GSI and allows for it to be integrated with other software such as Globus, which also support GSI.
- There are other recommendations that can be made with regards to the installation of SRB. These recommendations are based on how to provide SRB in a production environment to ensure availability and performance.
  - The first recommendation in relation to SRB is that it be installed with virtual hostnames and virtual interfaces. This helps in moving the application from one machine to another in the event of hardware failure. This also helps when the software is installed on operating systems that have virtualisation capabilities, which leads to our second recommendation.

- The second recommendation in relation to SRB is that it be installed on operating systems capable of virtualisation. By using virtualisation multiple SRB instances can be installed on the one machine. This helps where there are multiple SRB servers talking to common or unique MCAT servers. Multiple SRB server instances can result from a need to split load, or isolate data. Using virtual operating system instances reduces the complexity as one working image can be duplicated and only the configuration of the SRB server needs to be changed. It also allows the SRB server to be run on the default port, rather than running multiple instances on the one operating system on different ports. This reduces the administration overhead where multiple SRB servers required. It also reduces the chance of human error associated with trying to stop different instances of the SRB server running on the same machine on different ports.
- The third recommendation in relation to the installation of SRB is that its MCAT database has fail-over capability. This can be achieved in a number of ways, using the SRB master slave MCAT configuration, or using a clustered database as the back, or using products such as Oracle Data Guard.
- The fourth recommendation is that the naming scheme for servers and zones be illustrative of the organization or department to enable easier searching of data collections. For example for a separate SRB instance for a faculty, the host name and zone name should be along the lines of srb.faculty.university.edu.au. Where the SRB instance is shared it should be srb.university.edu.au/faculty. Obviously the share model is preferred, but where ownership of data has legal ramifications and the instance needs to be isolated, the first naming convention would have to be used. This has met with positive feedback from the SRB community we have communicated with.

- The fifth recommendation is in regards to storage virtualisation, both at the storage level and the application level (SRB). With all forms of virtualisation there is overhead. Therefore SRB should be used primarily as secondary storage, except in the case of primary storage where the instrument provides annotated data, and that data requires no further reduction.
- The recommendation in relation to HSM software is that it be one that is supported by SRB. SRB has support for DMF, SAMFS and CASTOR. Each of these HSM products has a similar level of expandability and functionality. The main difference being CASTOR is free. It does however require more support staff as one of the conditions of using the software is that a 0.5 EFT staff be available to help with its development. It also doesn't have a built in management database, and therefore requires an external database and associated database administration support.
- The HSM software chosen will dictate the hardware chosen for tape storage, disk storage and the hardware the operating system underneath the HSM software. As SRB runs on most UNIX operating systems, the HSM software itself is more of a dictating factor over which hardware is used.
- With regards to hardware recommendations, as the use of SRB allows a great deal of flexibility, our main recommendations are in the area of resilience, performance and reliability. Where possible we recommend redundant connections, whether those are network, or SAN connections. Most operating systems and hardware support multi-pathing and trunking of interfaces to improve resilience and performance. This however may dictate to some degree the selection of hardware and operating system if these recommendations are to be adhered to.

## 6 Terms of Reference

### 6.1 Glossary

Acronym	Definition
CASTOR	CERN Advanced STORAge manager.
DMF	Data Migration Facility
GSI	Grid Security Infrastructure
HSM	Hierarchical Storage Management
MCAT	Metadata Catalogue
SAN	Storage Area Network.
SRB	Storage Resource Broker

### 6.2 References

[1] DART Overview presentation.

[http://dart.edu.au/DART\\_Overview.pdf](http://dart.edu.au/DART_Overview.pdf)

[2] SI9 Work Unit Description.

[3] SRB FAQ

[http://www.sdsc.edu/srb/index.php/FAQ#What\\_does\\_the\\_SRB\\_do.3F](http://www.sdsc.edu/srb/index.php/FAQ#What_does_the_SRB_do.3F)

[4] HSM definition

<http://www.webopedia.com/TERM/H/HSM.html>

[5] DMF definition

<http://www.sgi.com/products/storage/tech/dmf.html>

[6] SAM-FS definition

[http://www.sun.com/storagetek/management\\_software/data\\_management/sam-fs/](http://www.sun.com/storagetek/management_software/data_management/sam-fs/)

[7] CASTOR definition

<http://castor.web.cern.ch/castor/>

## **7 Appendix A**

### **7.1 Detailed technical documentation.**

A detailed technical document is being compiled in parallel to this document. This document goes into far more depth about findings associated with testing of components and recommendations made in this brief. It also contains more in depth information discovered from investigation existing installations.