



Storage Resource Broker (SRB<sup>®</sup>) for Life Sciences  
Advancing Enterprise Collaboration and Regulatory Compliance

---

# White Paper

## Table of Contents

<i>Executive Briefing</i> .....	3
<i>Introduction</i> .....	5
Rise of the Needs.....	5
Current Solutions.....	5
SRB Background.....	6
<i>SRB Architecture</i> .....	7
SRB Design .....	7
Terminology.....	7
Components .....	8
Metadata .....	10
<i>SRB Features</i> .....	12
Highlights.....	12
Virtualization.....	12
Services.....	13
Security.....	13
<i>Enterprise Integration</i> .....	15
Data Collaboration and Management Suite .....	15
Performance.....	15
Automation .....	17
Additional Drivers.....	18
Tickets .....	18
Reports .....	18
<i>Conclusion</i> .....	19
<i>About Us</i> .....	19

## Executive Briefing

Biotech/pharmaceutical companies and biomedical/healthcare institutions face unique challenges in multi-location, cross-enterprise collaborations and regulatory compliance in addition to data growth and security assurance.

The industry needs a data solution that is robust and scalable to accommodate projected data growth; that supports industry-standard encryption and access control; that maintains accessibility and traceability of legacy records; and that enables colleagues around the world to share high-value data in collaborations, partnerships and outsourcing arrangements.

### **SRB: a proven solution to the challenges**

The SRB data grid implementation connects and unifies various data storage devices in a single representation, using advanced management tools and enterprise integration interfaces, thus enabling transparent access and easy manipulation of huge sums of data in a variety of storage devices, old and new, wherever located. Collectively, SRB and those devices form a “federation”, which embodies scalability, diversity and geographical separability. Along with advanced collaborative features, SRB enables dynamic, long-distance collaborations.

Regulatory Affairs is a critical facet of pharmaceutical development, and SRB’s features directly address concerns surrounding data integrity, authenticity, confidentiality, traceability and longevity. With direct access to a wide variety of storage devices, SRB answers the need for long-term data accessibility for companies in pharmaceuticals and health care who still maintain voluminous historical data in legacy storage devices.

Proven in demanding environments since 1995, SRB has become the most widely adopted data federation software in America’s research communities. The system currently manages more than 120 terabytes of data in 30+ million files at over 100 sites including NASA, Stanford and Harvard.

In an ongoing massive collaboration project, SRB helps scientists and medical doctors in leading universities and national laboratories gain access to MRI scan images in one gigantic collection, which are in fact acquired and stored all over the country, all with appropriate controls for patient confidentiality and privacy in compliance with governmental regulations.

### **SRB enables collaboration and boosts productivity cost-effectively**

SRB provides transparent access to virtually limitless data of heterogeneous origin and location in a collaborative environment. Improved communication and data availability translate into higher productivity and accelerated development. In some of the most challenging integrations, users testify that the success of their projects would not be possible without SRB.

Depending on the existing IT infrastructure, companies may find savings variable yet significant. SRB contributes to a company’s bottom line by:

- ❖ providing easy access to, and rapid integration of, old and collaborator’s data;
- ❖ cutting down idle time and improving logistics;
- ❖ extending useful life of legacy systems;
- ❖ saving the cost and effort to upgrade and migrate data otherwise necessary;
- ❖ optimizing storage resource use by priority data handling thus reducing needs to upgrade; and...
- ❖ facilitating integration by abstracting data from storage device type and location.

SRB also reduces the risk of data preservation as the original data can remain unaltered in its storage device while still well represented and accessible to new systems and tools.

For the same reason, SRB minimizes risk for data in installation and upgrading. It is a safe solution to deploy and can be rapidly integrated with existing systems.

### **SRB boasts high performance, rich functionalities and wide interoperability**

SRB has a client-server architecture with multiple mechanisms to boost performance with patented technologies. Various storage devices are connected through SRB Agents over network to an SRB server.

Data from the member devices is analyzed and a metadata repository called MCAT is established to enable a variety of data services including discovery and association. Clients connect to SRB in several different supported interfaces, e.g., through a web browser or just the Windows Network Neighborhood.

Advanced data management features are implemented, including data replication/optimization, custom attributes and search, advanced security and access control, parallel and bulk operations, and collaborative functionalities.

SRB works with virtually all breeds of computer and storage systems, which include—

- ❖ Operating and file systems: AIX (IBM), HP-UX (HP), Irix (SGI), Linux, Mac OS X (Apple), SunOS and Solaris (Sun), Windows (Microsoft), etc.
- ❖ Relational databases: DB2, Oracle, Sybase, Microsoft SQL Server, Postgres, etc.
- ❖ Archive systems: EMC Centera, HPSS, DiskXtender 2000, DiskXtender UX (formerly Unitree), SAM-FS, AMASS, all SCSI-based tape libraries, etc.
- ❖ Internet/Grid Servers: FTP, HTTP (Web pages), GridFTP, etc.

SRB leads in interoperability and interface support. A multitude of interfaces are available, from GUI clients to the Software Development Kit (SDK). Nirvana works closely with leading systems integrators and developers to bring timely and complete solutions to complex environments.

SRB is an enabling technology without peer. It is the only current, full-function, market-ready vehicle for accomplishing enterprise-wide and cross-enterprise collaborative integration and data preservation. In biomedical research, pharmaceutical development and health care, dynamic collaborations, regulatory requirements and sheer volumes of data demand just such a solution.

## Introduction

This paper investigates a solution for accessing and managing heterogeneous data, especially for the purpose of collaborative development and data preservation in biomedical research, pharmaceutical development and health care industries.

### Rise of the Needs

*“Suddenly information technology was so essential that we realized we were an information company more than we are a pill company. Because it’s the software—all the research, networking, marketing—that’s important in that pill.” --Jan Leschly, CEO, SmithKline Beecham(now GlaxoSmithKline), 2000*

Modern genomics and proteomics have ushered in an era of unprecedented data growth. It is not uncommon for even small companies to generate a large amount of data. For example, Perlegen, with fewer than 100 people, was reported to generate over 2 terabytes of data per week. Meanwhile, digital production and conversion of clinical data have become mainstream as acquisition and analysis technologies mature and gain prevalence. More data will be produced in the next three years than that from all previous time of human history, according to a recent study at UC-Berkeley. Exposed to both trends in the activities of R&D and clinical studies, the Life Science industries face pressing needs to expand data storage capabilities and upgrade data management tools.

The complexity of this task is further compounded by the requirements of regulatory compliance with government agencies such as the FDA. Regulations such as 21 CFR Part 11 and HIPAA were published not long ago, with companies still chasing the deadlines. In addition to integrity, authenticity and security, historical data have to be preserved and maintained with the assurance of full accessibility and traceability. In many cases, patent prosecution and litigation also require long-term data preservation and audit ability. Controlled and tracked access to heterogeneous data including those residing in legacy databases is essential.

Yet another dimension of this challenge is the collaborative nature of biomedical research and pharmaceutical development. Typically an established pharmaceutical company has dozens of locations around the world. M&A (merge and acquisition) activities further add to the dynamics. Not only do the researchers and clinicians have to collaborate within the organization, in most cases they also have to work cross-enterprise with people in other companies, research institutions, hospitals and governments. Effective and efficient sharing of heterogeneous data in amorphous formats is critical in such collaborations. The access has to accommodate multiple users working concurrently with respective access privileges in a collaborative environment. The storage has to guarantee reliability and integrity while maximizing resource utilization. To connect multiple systems and pool all disparate types of data in an organization is rated as one of today’s biggest challenges by industry executives.

Today’s biomedical industries need a new set of data solutions -- scalable enough for data growth, versatile enough to work with a variety of systems, old and new, from multiple vendors, and smart enough to connect those systems seamlessly for collaborations even when they are continents apart.

### Current Solutions

*“Over the last 8-10 years this increased need for online storage strategies has been answered partially with various “home-grown” systems with a patchwork of storage solutions cobbled together with time-stamping middleware, and an ad-hoc security scheme which is often simply an extension of existing network policy. The FDA has been patient with the evolution of such systems, but the agency is clearly trending toward a higher level of enforcement around data products.” -- Dr. Michael Breggar, Deloitte Consulting, 2003*

Three groups of vendors are active in this space. The first group, traditional IT powerhouses have been offering special packages to Life Science customers designed to fit the needs of increasing capacity and enhanced security. They generally leave the fulfillment of more specific needs in regulatory compliance and collaboration to system integrators, the second group. In many cases, it is the third group, smaller companies with clear focus, that pioneer in offering solutions with newer technologies, open standards, and better regulatory compliance. Many of the third group team up with members of the first group to offer full solution packages with their software running on name brand hardware.

Rather than forsaking the existing systems, Life Science companies tend to hire the second group of integrators to patch up systems. This approach is value-justified until the integration cost runs high, often as much as ten times the hardware cost. Companies seem to be caught in a dilemma of comparable evils—to migrate: the upfront cost of new systems is high and sometimes it is not an option for historical data in legacy systems; to patch: the cost can build up even higher and the result can be an unwieldy mosaic mess. Managing heterogeneous existing systems has a strong correlation to overall IT expenditure in a recent survey of 156 CFOs by Booz Allen.

Quality solutions tailored for collaboration among geographically dispersed parties are even more scarce, because integrating heterogeneous data sources remotely remains a major technical challenge. The utopia, as described by an industry leader, is to provide combined virtual services of geographically scattered computing resources to users just as scattered. For most companies, it remains a utopia.

Data grid, which addresses connectivity to both legacy systems and remote resources, seems to be the direction of evolution. Already, many in the academic research community have been harnessing the power of data grids. The leading solution among that group is the SRB data grid implementation.

### SRB Background

*“Data grids provide the mechanisms needed to manage distributed data, the tools that simplify automation of data management processes, and the logical name spaces needed to assemble collections. The Storage Resource Broker data grid implementation is an example that has been successfully applied to a wide variety of scientific disciplines, over many years of evolution and refinement, for management of massive collections.”*  
—Dr. Reagan Moore, San Diego Supercomputer Center, 2004

SRB is a data grid implementation that is easily and quickly deployed, operates in most compute and data environments, and provides, through data abstraction and metadata, the capabilities required to speed collaborative innovation in Life Sciences. Various types of data scattered in multiple geographic locations can be accessed in one representation from any location to provide true workflow integration and “view of the whole picture”. Collaborations are enabled and empowered to tackle “grand-scale” problems from a whole new level and perspective. By connecting to a wide variety of storage devices, and potentially collaborating with other SRB systems, SRB can accommodate virtually limitless amount of data. Nirvana’s SRB is also intended to satisfy regulatory requirements in data integrity, authenticity, confidentiality, traceability and longevity in the original data storage.

SRB traces its origin to an academic effort at the San Diego Supercomputer Center. Dr. Reagan Moore developed the first version of SRB in 1995, two years before the first major grid computing project, Distributed.net. Some of this SRB experience is described in papers entitled “The SDSC Storage Resource Broker,” and “SRB Case Studies”, both of which are available on request.

Since 1995, SRB has been proven in the nation’s most advanced supercomputing centers, as the basis for data grids, digital media libraries, and persistent archives. It has been implemented at the heart of over 100 data federations in some of the world’s most demanding compute environments, to manage digitized & scanned content and images in biomedical research and geographical informatics. In a conservative estimate, at least 120TB data in 30 million files in the 100+ data federations is currently managed by the SRB technology.

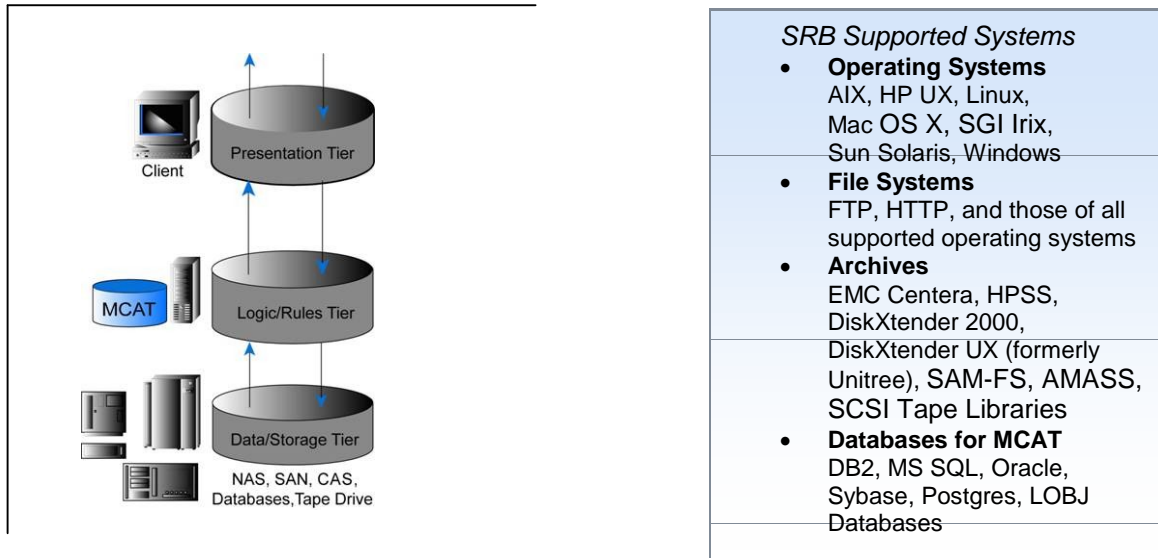
In the newest commercial version, SRB is buttressed with industrial-strength features, such as advanced security, bulk operations, parallel data streaming and packing, automated policy-based enterprise-wide data management, and a multitude of interfaces adapting to most enterprise environments. The following chapters describe the design and features of SRB in detail.

## SRB Architecture

### SRB Design

SRB offers a simple user presentation, in effect a “Collection hierarchy” – an extension of a standard directory hierarchy – of files residing in heterogeneous data sources. The SRB design can be described conceptually as three major layers: the SRB Clients (Presentation Tier), the Metadata Catalog or MCAT (Logic Tier), and the SRB Agents with drivers to the storage systems (Data Tier). This concept is illustrated as follows:

Figure1: SRB Design



### Terminology

#### Data Objects

Every piece of data managed or accessed through SRB is represented as a Data Object. Examples of Data Objects are: images, metadata files, databases, spreadsheets, database queries, URLs, or others. Data Objects can physically reside anywhere within an SRB Federation including file systems, tape drives, tape libraries, relational databases, or archives. A Data Object is not necessarily the same as a file, although it can be. Furthermore, Data Objects do not necessarily have a one-to-one relationship to the underlying data. A Data Object can in fact point to several (replicated) pieces of data (a one-to-many relationship). Therefore, a Data Object can be viewed as a logical entity, whereas a file refers to a physical entity. There are numerous attributes automatically associated with each Data Object in SRB: name, data type, size, physical path, creation timestamp, modification timestamp, last access timestamp, and other custom attributes, etc.

#### Collections

An SRB Collection is (much like a folder or directory in a file system) an object that contains other Collections or Data Objects. A Collection is used to organize Data Objects into a logical hierarchy that is easily accessible and understood by both user and administrator. For example, a Collection named "Project X" can be used to store all the Data Objects related to project X, independent of where that data is physically stored or how it is physically structured. This logical Collection view is maintained virtually through SRB so that users do not need to keep track of the physical locations of their data. Collections in SRB possess the same attributes as Data Objects.

### Global Namespace

The entire logical hierarchy comprising all Collections, Sub-Collections, and Data Objects is called a Global Namespace, also referred to as a *Logical Namespace*. By definition, a Global Namespace can span multiple heterogeneous and distributed storage systems and data centers.

### Storage Resources

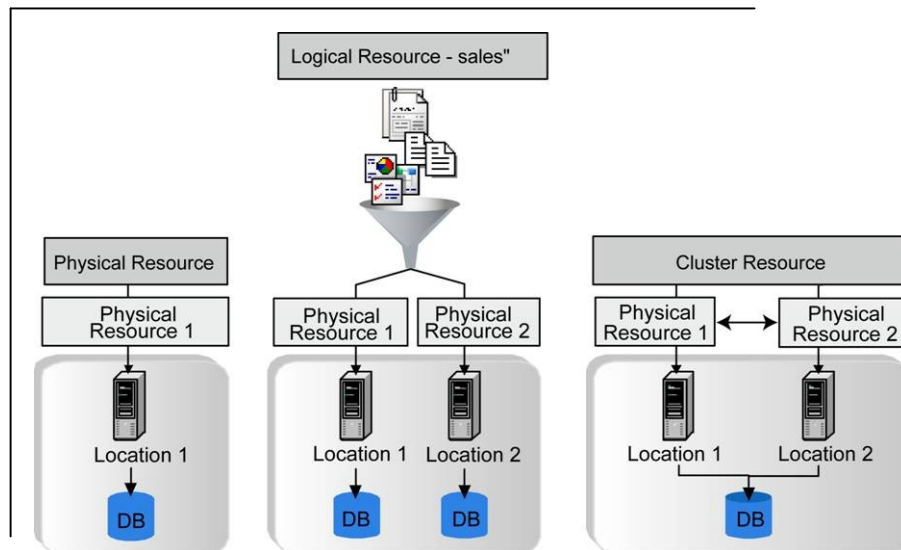
Every piece of data must ultimately be on a physical storage system. In SRB, the mapping between those storage systems and the Data Objects is done using Resources. There are three types of Resources in any SRB Federation: *Physical Resources*, *Logical Resources*, and *Cluster Resources*.

**Physical Resources** represent abstractions of places where Data Objects are physically stored. Such “places” include file systems on UNIX, Linux, or Windows; relational databases like Oracle, Sybase, or DB2; tape drives and libraries; Content Addressable Storage (CAS) systems; Web or FTP servers etc.

**Logical Resources** are used to group one or more Physical Resources together, making it transparent to the SRB user where data is physically stored. Logical Resources can be useful for many reasons: (a) data can be transparently stored to several underlying resources simultaneously (replication), (b) Resources can be switched out or added, (c) load balancing can be managed among several Resources, and (d) SRB Containers can be used as a mechanism to automatically archive and stage the data in the correct archival or cache resource — and all this can be accomplished transparently to SRB users.

**Cluster Resources** enable the efficient management of clustered file systems and tape media that can be accessed through several tape drives (e.g. in a tape library). To any SRB Client a Cluster Resource shows up as a Physical Resource. Behind the scenes, the Cluster Resource might in fact have many Physical Resources attached. When data is read or written on the Cluster Resource, the Cluster will iterate through its attached Physical Resources until it finds a working Resource or the right tape in a drive that is not busy. Cluster Resources provide SRB with an elegant automatic fail-over from one server to the next if those servers are seeing the same data.

Figure 2: SRB Logical and Physical Resources

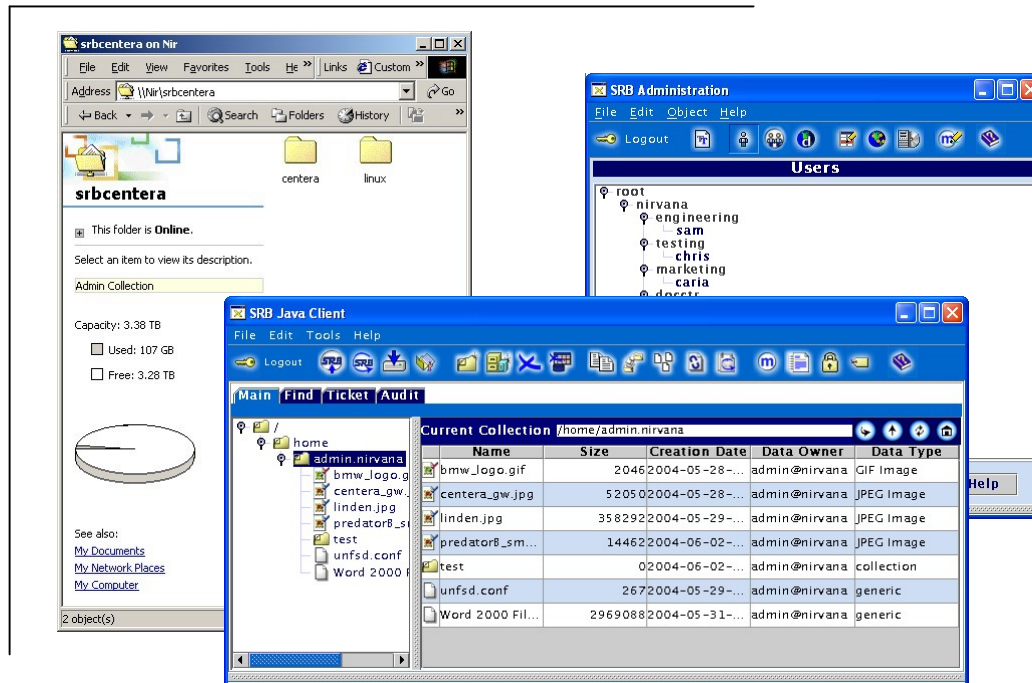


### Components

SRB is implemented as a distributed software layer using a client-server architecture. The combination of all SRB Servers, Clients, and storage systems is called a **Federation**. Every Federation must have a central master server connected to a Metadata Catalog (MCAT). SRB Agents can be installed optionally on other servers besides the master server. The client interfaces provide easy administration, navigation, and access

to all the data within the Federation. The SRB configuration contains an MCAT and a number of client interfaces.

Figure 3: SRB User Interfaces



## MCAT

The Metadata Catalog (MCAT) includes attributes required for the implementation of the Logical Namespace and the mapping of Data Objects to Storage Resources within the Federation. The MCAT is used to determine where a given Data Object is physically located and which drivers handle and access the data of interest. Furthermore the MCAT contains all file attributes, access control lists, storage resource information, user data and other metadata. MCAT also handles queries.

## Agents / Locations

Optionally, the SRB Federation maintained in MCAT can be expanded to span multiple servers within a department, the entire enterprise, or even beyond the enterprise and across administrative domains. This is achieved by installing additional software called **SRB Agents** on the respective servers. These Agents can either reside on an existing or (for performance reasons) on a dedicated server. An Agent provides a full range of services such as data handling, load balancing, data discovery, legacy system integration, and proxy services.

Each SRB Agent listens to a unique IP address and port number combination (a **Net Address**) on the server machine for requests from other Agents or SRB Clients. The combination of a unique port number and IP address (or hostname) is called an **SRB Location**. Therefore, several SRB Locations (or unique port number/IP address combinations) can coexist on a single server machine if they are monitoring different port numbers or IP addresses. SRB Locations are hierarchically organized.

SRB installations typically start on a single server with one or two storage systems attached. Storage such as EMC Centera can serve as an online archive in such an environment. The installation can then easily be expanded through the addition of SRB Agents.

## Drivers

The MCAT or an SRB Agent may contain any number of drivers to interface with various storage systems. SRB is configured with drivers to EMC Centera and a number of file systems. Additional drivers are separately available and new drivers are continually under development. As data grows in volume and complexity (due to reorganization, collaboration, acquisition, etc.), SRB enables continuing data access and greatly simplifies migration to new storage formats and media technologies.

## Client Interfaces

SRB has the following interfaces ready for shipping:

- CIFS Gateway for Windows and Mac OS X
- NFS Gateway for UNIX/Linux
- Java Admin
- Java Client
- Web Client
- Command Line (Scommands, Acommands, Mcommands)
- Client APIs

The CIFS Gateway for Windows essentially creates a Windows network share using a Samba server with an SRB Module in the backend. Applications running under Windows will be able to access data as if it were on their local disk. For applications running on UNIX/Linux the NFS Gateway will provide the same functionality as the CIFS Gateway on Windows.

The entire administration of SRB can be performed through a central console (Java Admin) and allows for such tasks as new Storage Resources creation, access permission change, user management, Metadata Scheme customization, and others.

A user may also interface with SRB through a Java Client that can be installed on any of the supported platforms. The GUI looks familiar to anyone who has used a file manager to copy and move files and folders. Compared to the gateways it has additional SRB-specific functionality like data migration, data discovery, and audit trails.

The SRB Web Client runs on any system that supports Internet Explorer or Netscape Navigator browsers. The SRB Web Client is therefore accessible from anywhere on the internet and does not require any client-side software installation.

The Command Line Interface includes Scommands, Acommands, and Mcommands. The Scommands are created to enable automation of certain tasks with scripts. The Scommands essentially extend the functionality of local UNIX commands (like 'ls', 'cd', 'pwd', etc.) into the Global Namespace and are especially useful for terminal sessions or scripting. Command-line-based administration is available through the Acommands. They are used for administrative tasks that require bulk operations or scripting. The Mcommands are utilities used for the maintenance of MCAT and are usually only necessary during the installation or upgrade process of MCAT.

## Metadata

The MCAT at the heart of an SRB Federation keeps track of several different metadata attributes. They are generally grouped into system-level and user-level attributes.

**System-level attributes** are automatically maintained by MCAT and can mostly be queried, displayed, or used for data management. System-level attributes are maintained on all SRB Objects including Data Objects, Collections, Storage Resources, Locations, Users, Groups, and Domains. Examples for system-level attributes are file path, Data Object size, resource type, etc.

**User-level attributes** are created and maintained by applications or users and can be attached to both Data Objects and Collections. They also are useful for query, display, and data management purposes. User-level attributes are access controlled on a per-attribute-basis so that SRB Administrators can create access policies, for example, "Engineering can modify the revision attribute, Analysts can read it, and the Public can not even see it". Other examples for user-level attributes are doctors, patients, diseases, hospitals, etc.

Data discovery is made infinitely easier through SRB's metadata attributes. Additionally, Collection metadata can ease discovery of Data Collections with attributes such as author or project name. SRB keeps access control lists in every user-level metadata attribute ensuring a high level of security. Different access levels like 'read' or 'write' are supported, guaranteeing that metadata is only viewed and modified by authorized users.

### *Metadata Schemes*

To make SRB management easier, SRB Administrators can group different metadata attributes into SRB Metadata Schemes. Metadata Schemes provide "attribute groupings" that may be specific to certain types of Data Objects or specific user groups within the organization. One example of this is a Metadata Scheme called 'Documentation' that contains the 'Author Name', 'Creation Date', 'Description', and 'Title'.

## SRB Features

### Highlights

An entry-level SRB system includes the following features:

- 20TB maximum capacity MCAT
- 3 SRB Users (super + 2 custom)
- 1 type of specialty storage driver (e.g., EMC Centera driver)
- Disk driver for UNIX/ Linux/ Windows file systems
- Windows (CIFS) and UNIX/ Linux (NFS) gateways
- Java Admin
- Java Client
- Mcommands (for MCAT maintenance)

SRB provides Windows, UNIX and Linux gateways to a variety of storage systems, enables unified access control for data and metadata, and features an integrated query engine for secure data discovery based on attributes.

Virtually every component of the configuration is upgradeable, therefore offering a scalable path to very large installations (multi-petabyte) with a wide variety of networked storage and server configuration options. Users can add SRB Agents, upgrade MCAT and server capacities, and also add optional Policy/HSM (Hierarchical Storage Management) and synchronization daemons, as required. Also available are Web Client, Scommands (command-line client), Acommands (command-line administration), the SRB Software Development Kit with APIs in C and Java, SRB Container technology for reducing latencies and complexities, additional reporting mechanisms and MCAT reports, Ticketing for time- and access-count-restricted access, and high performance bulk operations.

### Virtualization

When using SRB, applications, clients, and administrators immediately benefit from the following levels of abstraction, which ease administration, data discovery, and data access:

#### *Information Repository Virtualization*

The information repository (or MCAT) where the Federation's metadata is stored is transparently accessed and managed by SRB. This allows for a seamless exchange of the underlying information repository over time. Currently there is support in SRB for a number of different relational databases (see *SRB Supported Systems* above). The entire information repository can be exported and imported using XML. SRB dynamically generates SQL to perform various operations on the MCAT.

#### *Storage Virtualization*

The technologies for storing data change rapidly every few years. The SRB provides a virtual, unified representation to its clients regardless which particular storage technologies underlie the environment. For SRB Clients it is no longer necessary to remember a server's hostname, IP address, operating system, access protocol, or even whether a system is an archival storage system or a cache system. SRB hides those complexities and alleviates data storage complications by defining a common set of operations for manipulating data through its drivers.

#### *Data Virtualization*

Data within the SRB Federation is virtualized through a mapping mechanism from the physical path to a logical name within SRB. Every Data Object in SRB has its unique Global Identifier that is independent from the physical data path. The SRB takes the approach of associating any metadata like access control lists, audit trails, checksums, or user-level metadata with the logical name (or Global Identifier). This enables the SRB to function across heterogeneous and distributed storage systems.

In the case that an SRB User does not know the logical name of a Data Object, data can be discovered and accessed solely through an object's attributes (like creation date, size, frequency of access, keywords, projects, etc.).

### *Access Virtualization*

The following mechanisms provide for a complete access transparency: 'single-sign-on' and access through one common set of APIs. Clients do not require previous knowledge of a system's APIs or security mechanisms.

### **Services**

The MCAT and the Global Namespace enable the transitioning of services such as backup or migration from a local administrative domain into the enterprise-wide SRB Federation. The following paragraphs give an overview of the most important services implemented in SRB.

### *Data Load*

There are two mechanisms to load data into SRB: registration and ingestion. Registration leaves the data untouched and only registers a pointer to the data in MCAT without transferring any of the data contents. Ingestion is similar to Registration but also transfers the data to an SRB Storage Resource.

### *Data Retrieval*

Data is as easily retrieved from SRB as it is loaded. It is completely transparent where the data is coming from because SRB and TCP/IP handle the routing of the data streams.

### *Replication*

Data replication – the process of creating and maintaining synchronized copies of a Data Object throughout an SRB Federation – has a number of uses and benefits: it enables disaster protection with instantaneous failover from one replica to another one; caching is achieved by allowing clients to access a local replica rather than a remote one; and load balancing works by distributing synchronized replicas across multiple servers.

### *Persistent Migration*

Using the Global Namespace as a logical view of federation-wide data, SRB Clients can migrate data to new Storage Resources without affecting the way users see data organized in their Collections.

### *Backup*

Unlike Replication, a backup is not automatically overwritten when the original data is modified. This protects against accidental or malicious destruction of data. A backup in SRB makes copies of Data Objects or entire Collections in designated backup Collections.

### *Restore*

The restore process recovers deleted or wrongfully modified objects to an earlier point in time.

### *Secure Queries*

A number of different objects can be queried through the MCAT: Resources, configuration information, Users, Groups, Domains, Locations, Data Objects, Collections, Containers, Tickets, Schemes, and Tokens. There are several hundred attributes that allow the creation of very complex SQL-style queries. At no point in time will SRB Clients gain access to any object/information for which they do not have at least 'read' access.

### **Security**

SRB provides authentication schemes to protect data (Challenge Response Mechanism, Grid Security Infrastructure - GSI, Kerberos) enabling Single sign-on throughout the whole organization. SRB's authorization

service maintains Access Control Lists (ACLs) on various SRB Objects: Data Objects, Collections, Resources, and Metadata Attributes. Support is also built in for data encryption (GSI, Kerberos), User management (users, groups, domains) as well as Audit trails.

### *Authentication*

There are a number of built-in mechanisms that ensure the authenticity of an SRB User: challenge response mechanism (password is not sent across the network), Grid Security Infrastructure (using public and private keys), and Kerberos (sophisticated ticket management mechanism to provide single sign-on for multiple networked services). When using the CIFS and NFS Gateways, administrators are free to set up any supported authentication mechanism for their end-users, such as the integration with Microsoft Active Directory (AD) or Lightweight Directory Access Protocol (LDAP).

### *Authorization*

Once users are authenticated to ensure that they really are who they claim to be, SRB performs authorization services on every SRB Object that is requested. This ensures that users gain access only to the appropriate objects. The level of access may vary depending on the Access Control List (ACL).

### *Users / Groups / Domains*

Users within an SRB Federation have roles assigned to them and only one *Super User* has the power to do and see everything. Others can be designated to manage a subset of the Federation and users can be assigned to manage Data Collections or to be data readers.

In order to ease the administration of an SRB Federation, administrators can create domains and groups of users. Access control lists can then be created referencing SRB Domains and Groups.

### *Access Control Lists*

Access is controlled (authorization service) on Data Objects, Collections, Resources, and Metadata Attributes. There are various levels of access for each object ranging from 'null' (no access) to 'all' (full access). A fine-grained security mechanism enables access control on a per-user-level. In order to ease user and access control management, access can also be controlled for entire SRB Groups or Domains.

### *Audit Trails*

Every transaction within an SRB Federation can be audited. An audit trail will typically contain information such as date of transaction, success or error code, user performing transaction, type of transaction, and notes. The audit trails are just like everything else stored inside MCAT and can therefore be easily queried and filtered.

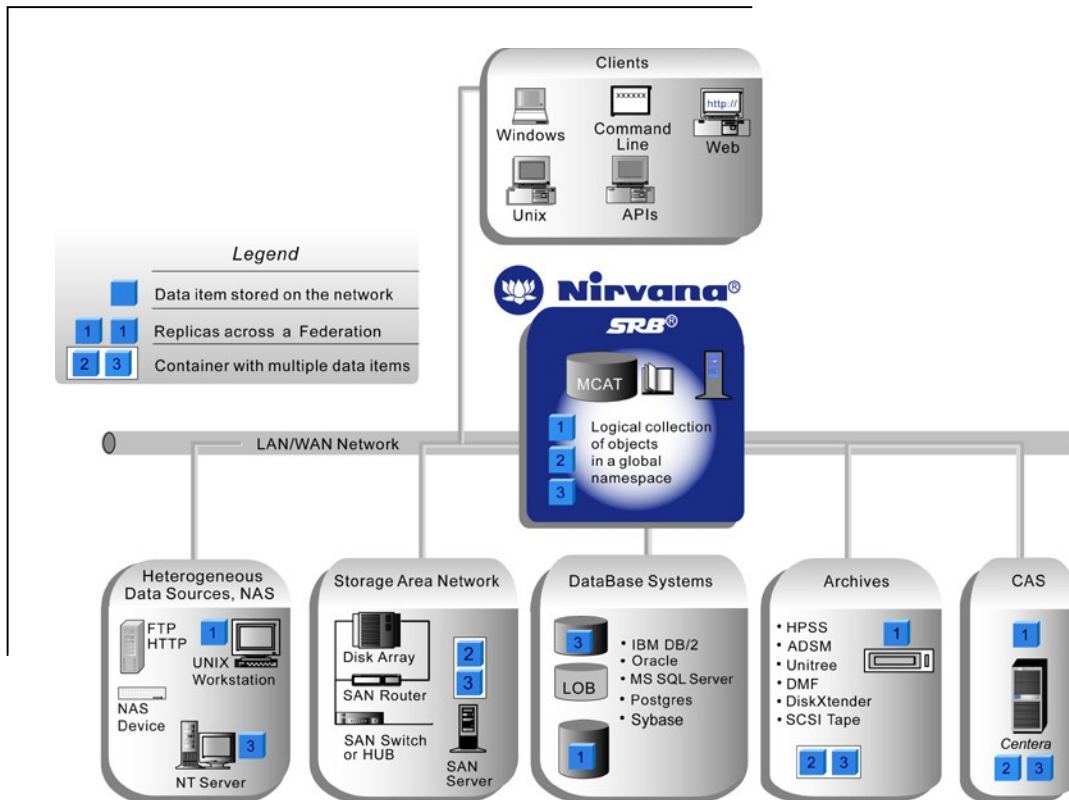
### *Parallel I/O*

The SRB communication protocol is a thin protocol implemented on top of TCP/IP. This protocol supports parallel communications using multiple sockets and a single static port initiated by an SRB Client. This greatly enhances the utilization of bandwidth, especially when sending larger files across Wide Area Networks (WANs). SRB's parallel I/O mechanism can be configured to use a single static port through a firewall since the request is initiated by the client. Because the SRB Server (or rather its administrator) has a better understanding of the server's bandwidth availability and speed of the attached storage system, the server controls the number of parallel I/O streams. In SRB's parallel I/O mechanism not only Data Objects are transferred using the parallel mechanism but also any other data (like a large query result) is transported in parallel.

## Enterprise Integration

Full integration of SRB into an enterprise information system gives rise to an entire enterprise-wide data grid. Along the expansion path, capacity can be increased, more resource types and drivers can be added, additional servers (or Locations) can be attached, data management processes can be automated, and performance can be scaled.

Figure 4: SRB Overview



### Data Collaboration and Management Suite

SRB is an advanced data grid implementation that enables simplified information management in ad hoc federations of users. Not only does it enable cross-platform access to data, but it also facilitates the discovery of the information residing on very different, usually incompatible, platforms going across administrative domains. SRB is a comprehensive solution for managing all the information in an organization. This includes table databases, LOBJ (Large Object) databases, file systems, tape drives, and archives. To allow viewing of table databases together with typical documents (spreadsheets, letters, forms), SRB can display customizable reports dynamically generated from database queries.

### Performance

Numerous performance advantages include prefetch for Containers, support for a network of distributed caches, support for multiple archive resources in Containers, and Collection-based (hierarchical) management of distributed digital objects. Parallel I/O and bulk operations are also supported by SRB.

### Latency Minimization

SRB features many different mechanisms to reduce latencies that occur during any data transfer:

- (a) *Streaming*...is the continuous transfer and receipt of data without any noticeable lag time.
- (b) *Caching*...keeps local copies of frequently accessed data on fast storage systems or memory.
- (c) *Staging*...temporarily writes Data Objects stored on archival storage media to disk for faster access.
- (d) *Data Containers*...transparently aggregate multiple Data Objects into one large file that can be stored and transferred more efficiently than multiple smaller files.
- (e) *Input/Output Containers*...allow for a remote execution of multiple commands in large batches.
- (f) *Replication*...keeps multiple synchronized copies of Data Objects at different sites.

Furthermore, SRB can pre-spawn processes on the server. The administrator can determine the size of the 'SRB pool'. Performance is improved by reducing latency during the initial connect to SRB.

### Scalability

SRB's architecture – comprised of Clients, Agents, and the MCAT database – allows for nearly infinite scalability by adding more Agents as more capacity is needed and scaling the MCAT database for higher transaction rates using the database vendor's mechanisms (like Oracle Real Application Clusters).

### Containers

Three major problems typically occur with distributed archival storage systems. The first is a high latency rate when retrieving data from tape. The second is file storage limitations. Archives are designed to store small numbers of very large files and data Collections are more likely to contain large numbers of small files. Finally, if the data is distributed over a Wide Area Network (WAN), serious latencies can result when transferring a large number of files. Through its patented Container technology, SRB can overcome all of these obstacles.

SRB manages Containers so that archives can be transparently integrated with all other enterprise-wide data. SRB incorporates Containers so it can manage distributed Data Collections across multiple storage systems.

Figure 5: Latency Minimization

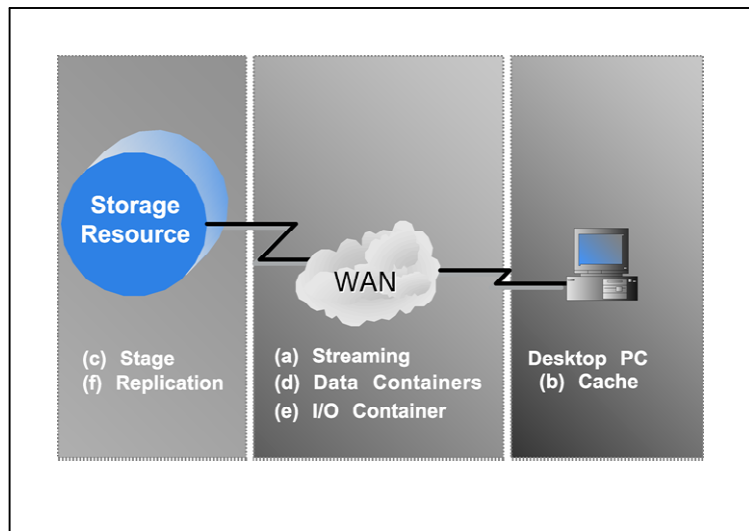
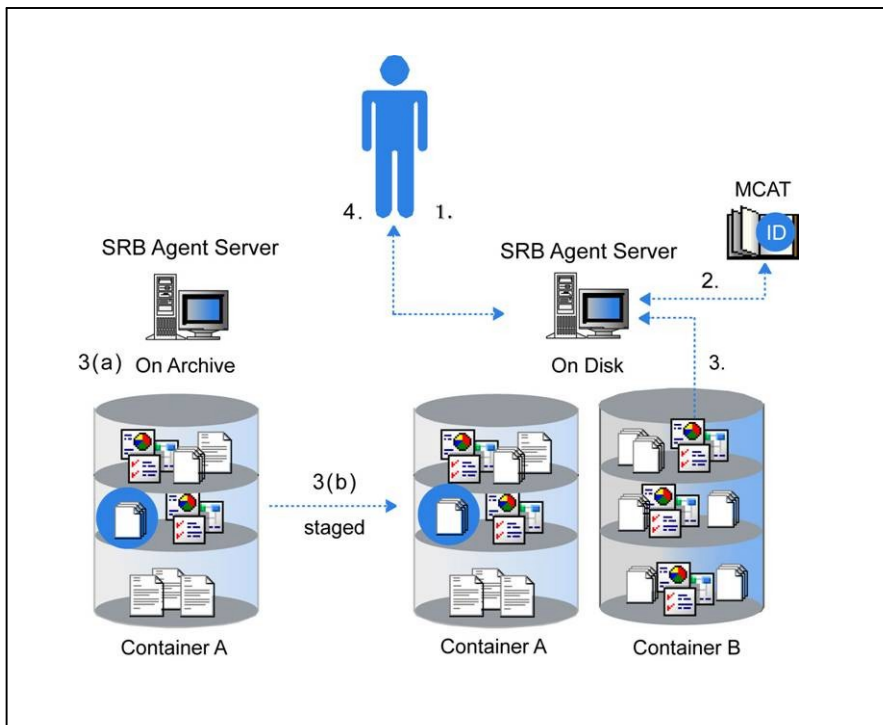


Figure 6: SRB Container Design



Additionally, SRB manages Containers transparent to end-users or applications by presenting in-container objects just like any other SRB Data Object in the Global Namespace.

Data Objects are physically packed into a Container before being stored into an archive. This assures that associated Data Objects will reside on the same tape while decreasing the number of Data Objects managed by the archive. When retrieving these Data Objects at a later time, the correct Container is automatically staged to disk and all the read access happens against the staged copy. Since related Data Objects are retrieved together, the access time can improve dramatically.

### Automation

SRB has the ability to automatically launch helper processes or daemons upon start-up. Two of those daemons are particularly useful when it comes to easing the data transfer into the SRB Federation or automating data management.

#### Directory Synchronization Daemon

As files are added to or deleted from local directories, the *Directory Synchronization Daemon* (DS) running in the background detects these activities. When changes are noted to the local directory structure, the new information is registered and updated in the MCAT.

Figure 7: DS Daemons

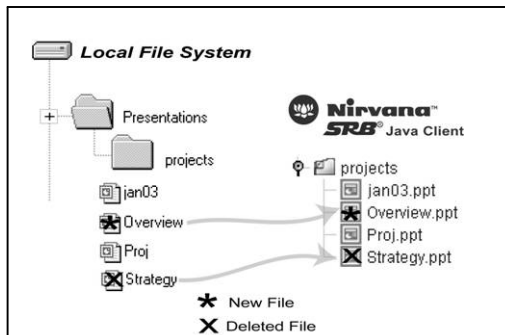
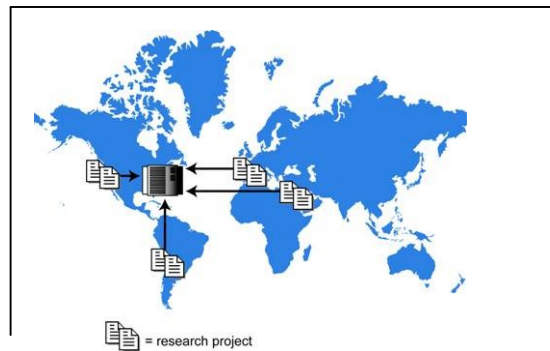


Figure 8: HSM Daemons



### Hierarchical Storage Management Daemon

A *Hierarchical Storage Management (HSM)* Daemon is a policy engine that routinely queries the MCAT. For example, an administrator sets a policy to migrate data from distributed Locations to one particular archival storage system with the criterion that all data over 10MB be included. In another example, data associated with a specific project and that has been accessed fewer than five times on average per day would be migrated. Besides migration, other actions can be performed on the data including replication, deletion, backup, or simply reporting. The policies are extremely flexible and can contain standard metadata, customized metadata or expressions.

All this happens behind the scenes, transparent to the end-user or applications, so that the Global Namespace does not change.

### Additional Drivers

A parallel disk driver is available that extends the standard SRB disk driver with extra performance using parallel threads and various caching mechanisms optimized for both random and sequential data access.

SRB includes relational database resources to permit DBMS-style row-based access to data stored in databases. SRB includes an HTML, XML or customized template interface to databases so that queries can be returned in any format and records within one format can easily be transferred into another format.

Most relational databases can also be used to store binary large objects (LOBs). SRB supports drivers for such large object databases in order to store Data Objects directly into database tables.

A driver to SCSI tape drives and tape libraries is also natively supported by SRB. Since tape storage is still relatively more cost effective compared to disk, SRB makes tape management also very straightforward and therefore reduces Total Cost of Ownership (TCO) for tape archives significantly.

Internet data sources can be seamlessly integrated with drivers for FTP and HTTP servers.

### Tickets

SRB employs an additional authentication mechanism whereby data-sharing Tickets can be sent out to internal or external SRB Users. The ticket then grants controlled access to Data Objects or entire Collections. Additional restrictions such as time limits and limits on the number of accesses can be built into every Ticket.

### Reports

In combination with relational database drivers, the MCAT can be used to report on various SRB Objects. Examples are:

- Resource Usage (how are resources utilized over time)
- Data Object Access Pattern
- Data Ownership report (which users uses how much storage)
- Valid Tickets
- Audit Trails (for certain Data Objects or entire Collections)

## Conclusion

SRB is a mature and stable product, under continuous development since 1995 and has been implemented at the core of some of the world's most advanced image repositories. SRB's simple and intuitive interface permits the easy management of different systems throughout a department, organization, partnership, or consortium, reducing complexity to simplicity in both ease-of-use as well as administration.

While satisfying the needs of data security and regulatory compliance, SRB serves as the fundamental framework of a data storage grid, whose time has arrived amid the exponential growth of data and data uses. The SRB is ideally suited to complex computing environments where knowledge workers would benefit from simplified, easy access to shared, high-value data in heterogeneous working environments. The advantages in data availability, resource utilization, record preservation and storage abstraction readily translate into significant productivity gain and cost benefits. More importantly, SRB enables research and development which are otherwise impossible.

As the most widely adopted data federation software of its kind, SRB is a proven solution to the rising needs in biomedical research, pharmaceutical development and clinical practice.

## About Us

*Nirvana Storage, a division of General Atomics, specializes in advanced information management systems for complex storage environments.*

*General Atomics ([www.ga.com](http://www.ga.com)), established in San Diego, CA since 1955, is a recognized leader in high-tech system development from nuclear fission and fusion to military surveillance airplane, the Predator UAV. General Atomics and its affiliated companies employ approximately 3,000 people in 12 locations worldwide. General Atomics founded the San Diego Supercomputer Center in 1985 and continues to pioneer in IT development such as the UWB (Ultra WideBand) technology.*

**Nirvana Storage, General Atomics**  
**3550 General Atomics Court**  
**San Diego, CA 92121-1122**  
**1, (858) 455-2500 Voice**  
**1, (858) 455-2529 Fax**  
**[srb@nirvanastorage.com](mailto:srb@nirvanastorage.com)**  
**[www.nirvanastorage.com](http://www.nirvanastorage.com)**