Grid Data Management Systems & Services

Data Grid Management Systems – Part I

Arun Jagatheesan, Reagan Moore

Grid Services for Structured data –Part II

Paul Watson, Norman Paton

VLDB Tutorial
Berlin, 2003

Very
Large
Data
Bases

Part I:
Data Grid Management Systems

Arun Jagatheesan

Reagan Moore

{arun, moore}@sdsc.edu

San Diego Supercomputer Center

University of California, San Diego

http://www.npaci.edu/DICE/SRB/

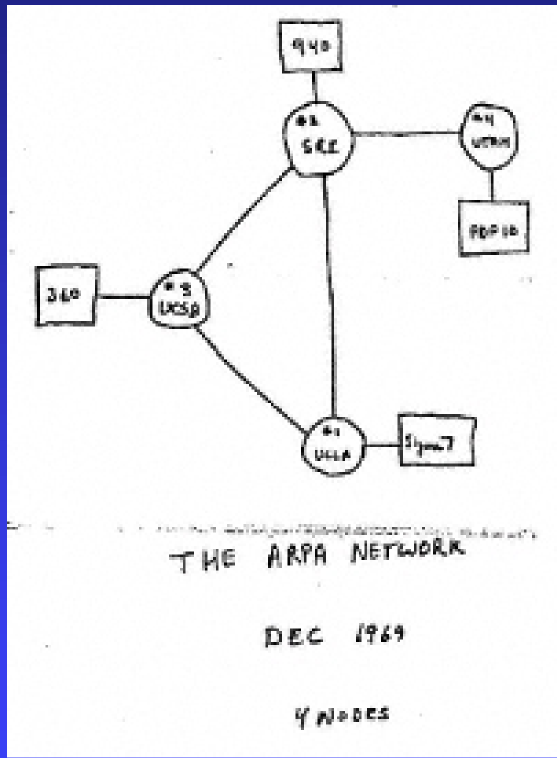
VLDB Tutorial
Berlin, 2003

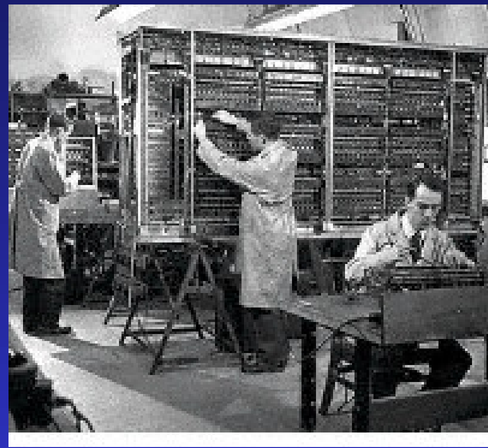
Very
Large
Data
Bases

Tutorial Part I Outline

- **Concepts**
  - Introduction to Grid Computing
  - Proliferation of Data Grids
  - Data Grid Concepts
- **Practice**
  - Real life use cases SDSC Storage Resource Broker (SRB)
  - Hands on Session
- **Research**
  - Active Datagrid Collections
  - Data Grid Management Systems (DGMS)
  - Open Research Issues

Distributed Computing





Distributed Data Management

- **Data collecting**
  - **Sensor systems**, object ring buffers and portals
- **Data organization**
  - **Collections**, manage data context
- **Data sharing**
  - **Data grids**, manage heterogeneity
- **Data publication**
  - **Digital libraries**, support discovery
- **Data preservation**
  - **Persistent archives**, manage technology evolution
- **Data analysis**
  - **Processing pipelines**, manage knowledge extraction

What is a Grid?

“Coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations”

Ian Foster, ANL

What is Middleware?

Software that manages distributed state information for results of remote services

Reagan Moore, SDSC

Data Grids

- ***A datagrid provides the coordinated management mechanisms for data distributed across remote resources.***
- **Data Grid**
  - Coordinated sharing of information storage
  - Logical name space for location independent identifiers
  - Abstractions for storage repositories, information repositories, and access APIs
- **Computing grid and the datagrid part of the Grid.**
  - Data generation versus data management

Tutorial Part I Outline

- **Concepts**
  - Introduction to Grid Computing
  - Proliferation of Data Grids
  - Data Grid Concepts

Are data grids in production use? How are they applied?

- **Practice**
  - Real life use cases SDSC Storage Resource Broker (SRB)
  - Hands on Session
- **Research**
  - Active Datagrid Collections
  - Data Grid Management Systems (DGMS)
  - Open Research Issues

Storage Resource Broker at SDSC

Storage Resource Broker (SRB)

*Data brokered by SDSC instances of SRB***

	As of 1/9/2002		As of 5/17/2002		As of 9/10/2002		As of 5/02/2003		As of 7/24/2003				
Project Instance	*Data_size (in GB)*	*Count (files)*	*Data_size (in GB)*	*Count (files)*	*Data_size (in GB)*	*Count (files)*	*Data_size (in GB)*	*Count (files)*	*Data_size (in GB)*	*Count (files)*	*Users*	*Comments*	*Funding Agency*
NPACI	2,828.18	807,737	1,972.00	1,083,230	2,214.00	1,131,017	4,480.00	1,818,530	6,050.00	2,317,368	367	NPACI Users	NSF/PACI
Digsky	10,565.00	5,079,883	17,800.00	5,139,249	29,006.00	5,225,347	33,930.00	5,292,161	46,100.00	5,719,025	68	2Mass,DPOSS,NVO	NSF/ITR
DigEmbryo	227.77	16,629	433.00	31,629	604.00	43,071	658.00	43,326	720.00	45,365	23	Visible Embryo	NLM
HyperLter	147.50	1,694	158.00	3,596	158.50	3,602	207.00	4,473	215.00	5,097	27	HyperSpectral Images	NSF/NPACI (ESS)
Hayden	3,917.80	18,112	6,800.00	41,391	6,827.00	42,227	7,078.00	59,399	7,078.00	59,399	142	FlyThrough for Planetarium	AMNH/Hayden
Portal	7.40	443	33.00	5,485	53.83	10,278	880.00	24,521	968.00	27,250	316	Grid Portal	NSF/NPACI
SLAC	434.80	9,905	514.00	77,168	605.60	83,839	1,663.00	236,688	1,790.00	254,974	43	Protein Crystallography	NSF/NPACI (Alpha)
NARA/Collection	0.02	301	7.00	2,455	7.80	30,890	47.00	34,077	52.80	79,195	51	Archival Documents	NARA
NSDL/SIO Exp			19.20	383	28.32	1,031	65.12	7,614	232.00	15,809	23	SIO Explorer Documents	NSF/NSDL
TRA			5.80	92	58.50	2,298	91.07	2,371	90.60	2,385	25	Classroom Videos	NSF/NPACI (EOT)
LDAS			239.00	1,766	424.41	3,157	477.86	9,368	498.00	9,858	60	LDAS	
BIRN					66.40	6,600	87.42	177,612	121.00	237,283	138	Biomedical Informatics	NIH (NCRR)
AfCS			27.00	4,007	49.36	6,985	65.80	11,654	95.30	18,762	20	Cell Signaling Images/Docs	NIH
UCSDLib							1,084.00	138,413	1,084.00	138,415	29	Archival Image Files	UCSD
NSDL/CI							177.20	775,959	278.00	993,886	113	K-12 Curriculum Web-sites	NSF/NSDL
SCEC									12.60	18,660	38	South Cal. Earthquake Ctr.	NSF/ITR
TeraGrid									623.00	36,508	1,978	TeraGrid	NSF
TOTAL	18,128.47	5,934,704	28,008.00	6,390,451	40,103.72	6,590,342	50,991.47	8,636,166	66,008.30	9,979,239	3,461		
	18 TB	6 million	28 TB	6.4 million	40.1 TB	6.59 million	51 TB	8.64 million	66 TB	9.97 million	3 thousand		

** Does not cover data brokered by SRB spaces administered outside SDSC.
Does not cover databases; covers only files stored in file systems and archival storage systems

More features, 60 Terabytes and counting

Very Large Data Bases

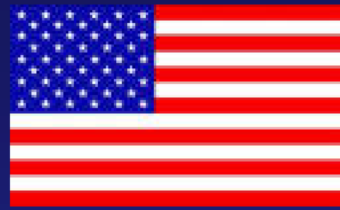
NSF Infrastructure Programs

- **Partnership for Advanced Computational Infrastructure - PACI**
  - Data grid - Storage Resource Broker
- **Distributed Terascale Facility - DTF/ETF**
  - Compute, storage, network resources
- **Digital Library Initiative, Phase II - DLI2**
  - Publication, discovery, access
- **Information Technology Research projects - ITR**
  - SCEC Southern California Earthquake Center
  - GEON GeoSciences Network
  - SEEK Science Environment for Ecological Knowledge
  - GriPhyN Grid Physics Network
  - NVO National Virtual Observatory
- **National Middleware Initiative - NMI**
  - Hardening of grid technology (security, job execution, grid services)
- **National Science Digital Library - NSDL**
  - Support for education curricula modules

Federal Infrastructure Programs

- **NASA**
  - Information Power Grid - IPG
  - Advanced Data Grid - ADG
  - Data Management System - Data Assimilation Office
    - Integration of DODS with Storage Resource Broker
  - Earth Observing Satellite EOS data pools
  - Consortium of Earth Observing Satellites CEOS data grid
- **Library of Congress**
  - National Digital Information Infrastructure and Preservation Program - NDIIPP
- **National Archives and Records Administration (NARA) and National Historical Public Records Commission**
  - Prototype persistent archives
- **NIH**
  - Biomedical Informatics Research Network data grid
- **DOE**
  - Particle Physics Data Grid

NSF GriPhyN/iVDGL



- **Petabyte scale Virtual Data Grids**
- **GriPhyN, iVDGL, PPDG – Trillium**
  - Grid Physics Network
  - International Virtual Data Grid Laboratory
  - Particle Physics Data Grid
- **Distributed worldwide**
  - Harness Petascale processing, data resources
- **DataTAG – Transatlantic with European Side**

- **Launched in August 2001**
  - SDSC, NCSA, ANL, CACR, PSC
- **20 Tera flops of computing power**
- **One peta byte of storage**
- **40 Gb/sec (FASTEST network on planet)**
- ***“Building the Computational Infrastructure for Tomorrow's Scientific Discovery”***

European Datagrid

- **European Union**
- **Different Communities**
  - High Energy Physics
  - Biology
  - Earth Science
- **Collaborate and complement other European and US projects**

NIH BIRN

- **Biomedical Informatics Research Network**
  - Access and analyze biomedical image data
  - Data resources distributed throughout the country
  - Medical schools and research centers across the nation
- **Stable high performance grid based environment**
  - Coordinate sharing of virtual data collections and data mining
  - Growing fast!

Commonality in all these projects

- **Distributed data management**
  - *Authenticity*
  - *Access controls*
  - *Curation*
- **Data sharing across administrative domains**
  - Common name space for all registered digital entities
- **Data publication**
  - Browsing and discovery of data in collections
- **Data Preservation**
  - Management of technology evolution

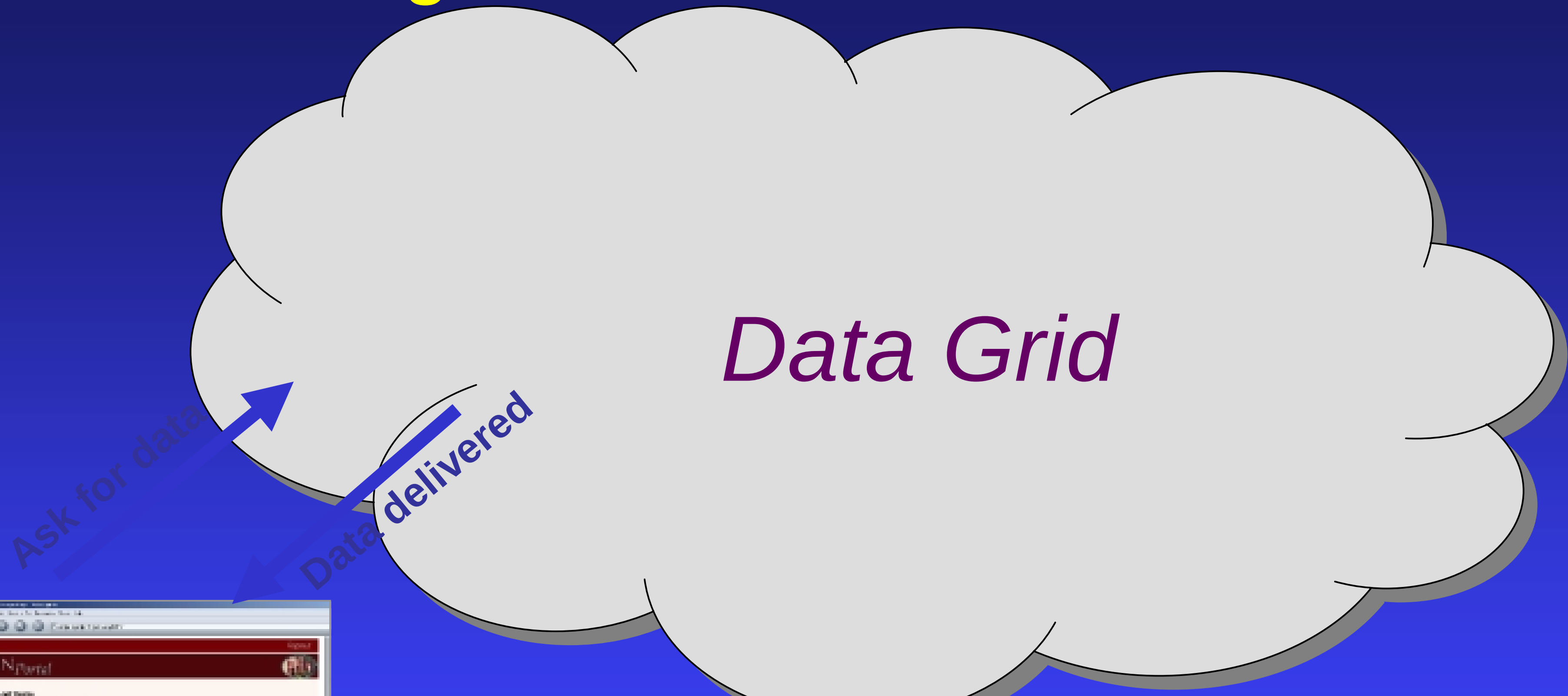
Data and Requirements

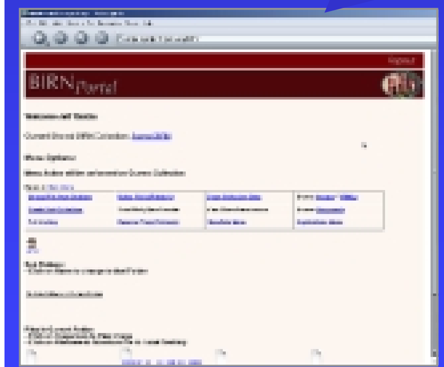
- **Mostly Unstructured, heterogeneous**
  - Images, Files, Semi-structured, databases, streams, …
  - File systems, SAN, FTP sites, Web servers, Archives
- **Community-Based**
  - Shared amongst one or more communities
- **Meta-data**
  - Different Meta-data schemas for the same data
  - Different Notations, ontologies
- **Sensitive to Sharing**
  - Nobel Prizes, Federal Agreements, project data

Tutorial Part I Outline

- **Concepts**
  - Introduction to Grid Computing
  - Proliferation of Data Grids
  - Data Grid Concepts
- **Practice**
  - Real life use cases SDSC Storage Resource Broker (SRB)
  - Hands on Session
- **Research**
  - Active Datagrid Collections
  - Data Grid Management Systems (DGMS)
  - Open Research Issues

Using a Data Grid – *in Abstract*





- User asks for data from the data grid
- The data is found and returned
- Where & how details are managed by data grid
- But access controls are specified by owner

Data Grid Transparencies

- **Find data without knowing the identifier**
  - Descriptive attributes
- **Access data without knowing the location**
  - Logical name space
- **Access data without knowing the type of storage**
  - Storage repository abstraction
- **Retrieve data using your preferred API**
  - Access abstraction
- **Provide transformations for any data collection**
  - Data behavior abstraction

Logical Layers (bits,data,information,..)

Semantic data Organization (with behavior)

Inter-organizational Information Storage Management

myActiveNeuroCollection   patientRecordsCollection

Virtual Data Transparency

image.cgi   image.wsdl   image.sql

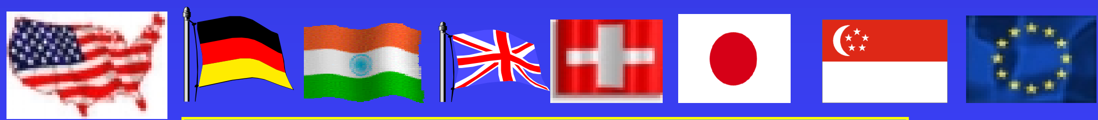
Data Replica Transparency

image_0.jpg…image_100.jpg

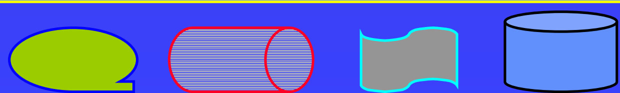
Data Identifier Transparency

E:\srbVault\image.jpg /users/srbVault/image.jpg Select … from srb.mdas.td where...

Storage Location Transparency



Storage Resource Transparency



Storage Resource Transparency (1)

- **Storage repository abstraction**
  - Archival systems, file systems, databases, FTP sites, …
- **Logical resources**
  - Combine physical resources into a logical set of resources
  - Hide the type and protocol of physical storage system
  - Load balancing – based on access patterns
  - Unlike DBMS, user is aware of logical resources
  - Flexibility to changes in mass storage technology

Storage Resource Transparency (2)

- **Standard operations at storage repositories**
  - POSIX like operations on all resources
- **Storage specific operations**
  - Databases - bulk metadata access
  - Object ring buffers - object based access
  - Hierarchical resource managers - status and staging requests

Storage Location Transparency

- **No fixed home or location for distributed data**
  - Transparent of physical location and physical resource
- **Virtualization of distributed data resources**
  - Data placement managed by data grid
- **Redundancy for preservation**
  - Resource redundancy – “m of n” resources in list
  - Location redundancy – replicate at multiple locations

Data Identifier Transparency

- **Four Types of Data Identifiers:**
- **Unique name**
  - OID or handle
- **Descriptive name**
  - Descriptive attributes – meta data
  - Semantic access to data
- **Collective name**
  - Logical name space of a collection of data sets
  - Location independent
- **Physical name**
  - Physical location of resource and physical path of data

Data Replica Transparency

- **Replication**
  - Improve access time
  - Improve reliability
  - Provide disaster backup and preservation
  - Physically or Semantically equivalent replicas
- **Replica consistency**
  - Synchronization across replicas on writes
  - Updates might use “m of n” or any other policy
  - Distributed locking across multiple sites
- **Versions of files**
  - Time-annotated snapshots of data

Virtual Data Abstraction

- **Virtual Data or *"On Demand Data"***
  - Created on demand is not already available
  - *Recipe* to create derived data
  - Grid based computation to create derived data product
- **Object based storage (extended data operations)**
  - Data subsetting at the remote storage repository
  - Data formatting at the remote storage repository
  - Metadata extraction at the remote storage repository
  - Bulk data manipulation at the remote storage repository

Data Organization

- **Physical Organization of the data**
  - Distributed Data
  - Heterogeneous resources
  - Multiple formats (structured and unstructured)
- **Logical Organization**
  - Impose logical structure for data sets
  - ***Collections*** of semantically related data sets
  - Users create their own views (collections) of the data grid
- **Digital Ontology**
  - Characterization of structures in data sets and collections
  - Mapping of semantic labels to the structures

Data Behavior Abstraction

- **Loose coupling between data and behavior**
  - Collection provides a structure to related data sets
  - Related data sets operated using a collective behavior
  - Any behavior (set of operations) is associated with a collection
- **Data Grid Collections impose behavior**
  - Describe a generic standard behavior using WSDL
  - Each collection gets its specific behavior by extending the generic behavior
  - Generic WSDL is extended using portType (or interface) inheritance

Tutorial Part I Outline

- **Concepts**
  - Introduction to Grid Computing
  - Proliferation of Data Grids
  - Data Grid Concepts
- **Practice**
  - Real life use cases SDSC Storage Resource Broker (SRB)
  - Hands on Session
- **Research**
  - Active Datagrid Collections
  - Data Grid Management Systems (DGMS)
  - Open Research Issues

SDSC SRB – The History

- **Started in 1995 funded by DARPA**
  - Massive Data Analysis System **(MDAS)**
  - PI: Reagan Moore
  - “Support data-intensive applications that manipulate very large data sets by building upon object-relational database technology and archival storage technology”
- **Multiple projects for many federal agencies**
  - DoD, NSF, NARA, NIH, DoE, NLM, Library of Congress, NASA
  - In production or evaluation at multiple academic and research institutions round the world

SDSC SRB Team - Data “R” Us :-)

Camera-shy

- Wayne Schroeder
- Vicky Rowley (BIRN)
- Lucas Gilbert
- Marcio Faerman (SCEC)
- Antoine De Torcy (IN2P3)
- **Students & *emeritus***
  - *Erik Vandekieft*
  - Reena Mathew
  - Xi (Cynthia) Sheng
  - Allen Ding
  - Grace Lin
  - Qiao Xin
  - Daniel Moore
  - Ethan Chen

World’s first ‘datagrid engineer’?

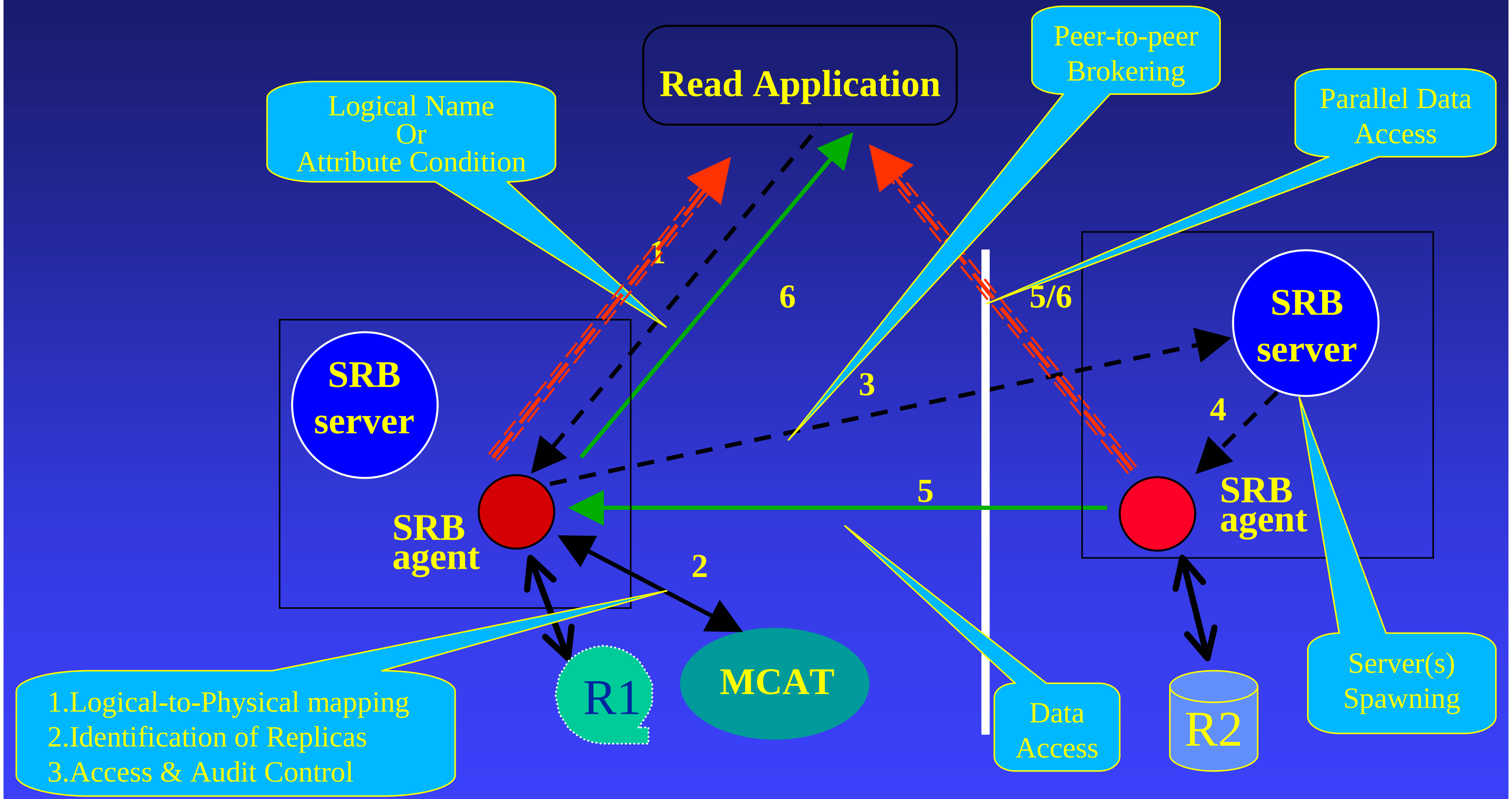
SDSC Collaborations

- Hayden Planetarium Simulation & Visualization
- NVO -Digital Sky Project (NSF)
- ASCI - Data Visualization Corridor (DOE)
- **Particle Physics Data Grid (DOE) {GrPhyN (NSF)}**
- **Information Power Grid (NASA)**
- **Biomedical Informatics Research Network (NIH)**
- **Knowledge Network for BioComplexity (NSF)**
- **Mol Science – JCSG, AfCS**
- **Visual Embryo Project (NLM)**
- RoadNet (NSF)
- **Earth System Sciences – CEED, Bionome, SIO Explorer**
- Advanced Data Grid (NASA)
- **Hyper LTER**
- **Grid Portal (NPACI)**
- **Tera Scale Computing (NSF)**
- **Long Term Archiving Project (NARA)**
- **Education – Transana (NPACI)**
- **NSDL – National Science Digital Library (NSF)**
- **Digital Libraries – ADL, Stanford, UMichigan, UBerkeley, CDL**
- **… 31 additional collaborations**

Production Data Grid

- **SDSC Storage Resource Broker**
  - Federated client-server system, managing
    - Over 50 TBs of data at SDSC
    - Over 8.64 million files
  - Manages data collections stored in
    - Archives (HPSS, UniTree, ADSM, DMF)
    - Hierarchical Resource Managers
    - Tapes, tape robots
    - File systems (Unix, Linux, Mac OS X, Windows)
    - FTP sites
    - Databases (Oracle, DB2, Postgres, SQLserver, Sybase, Informix)
    - Virtual Object Ring Buffers

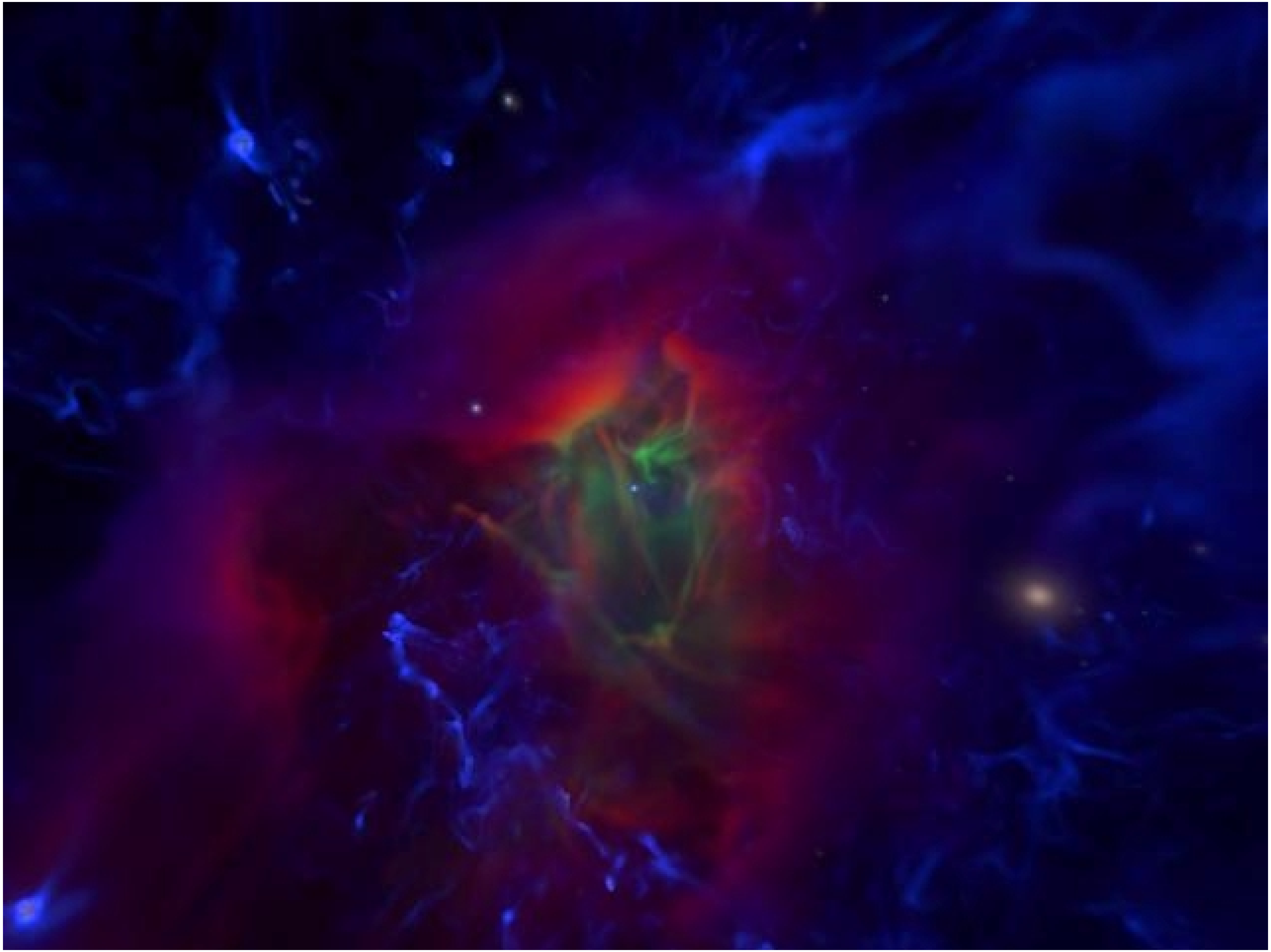
Federated SRB server model



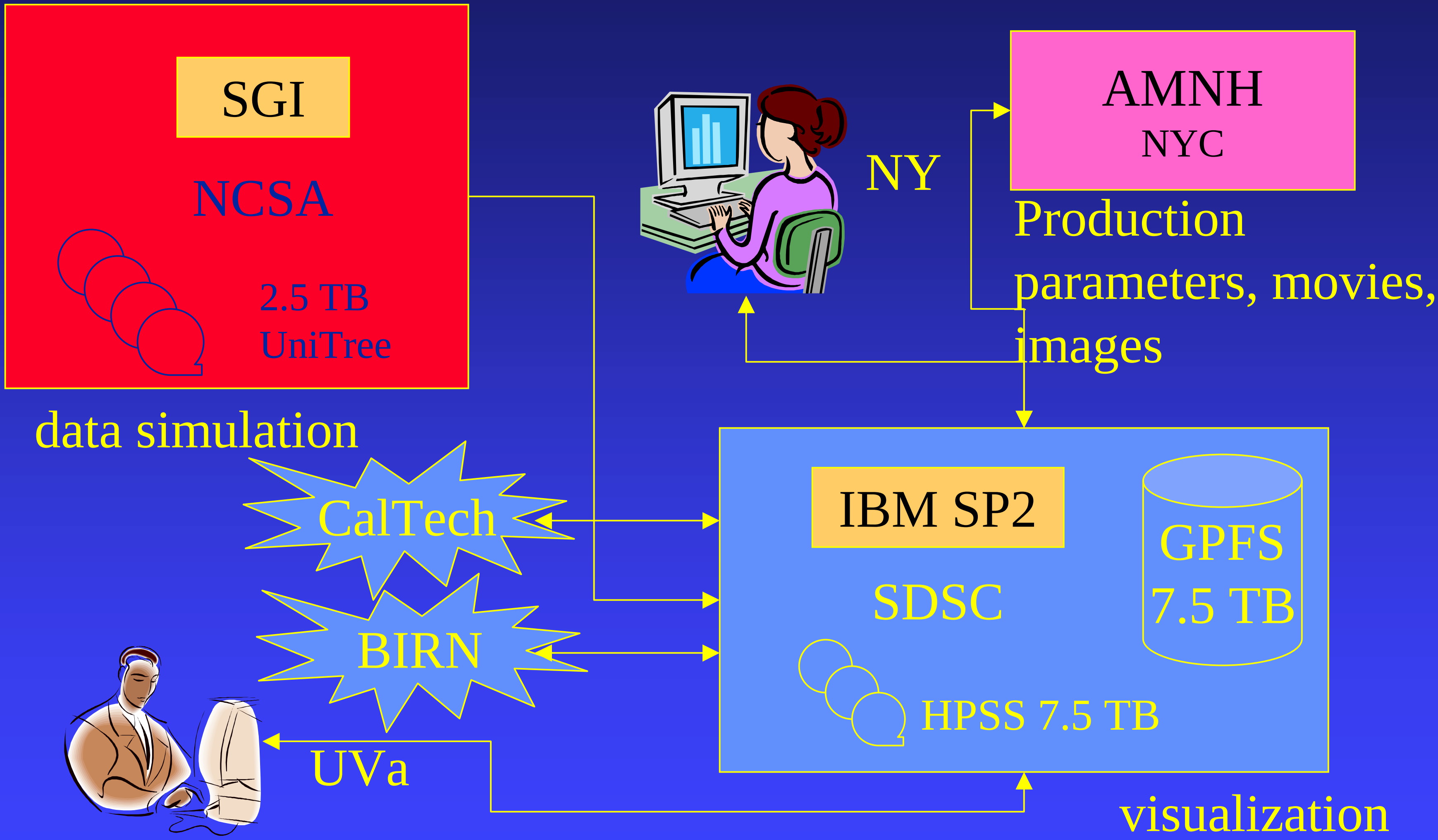
Very
Large
Data
Bases

Logical Name Space
Example - Hayden Planetarium

- Generate “fly-through” of the evolution of the solar system
- Access data distributed across multiple administration domains
- Gigabyte files, total data size was 7 TBytes
- Very tight production schedule - 3 months



Hayden Data Flow



Mappings on Name Space

- **Define logical resource name**
  - List of physical resources
- **Replication**
  - Write to logical resource completes when all physical resources have a copy
- **Load balancing**
  - Write to a logical resource completes when copy exist on next physical resource in the list
- **Fault tolerance**
  - Write to a logical resource completes when copies exist on “k” of “n” physical resources

Hayden Conclusions

- The SRB was used as a logical central repository for all original, processed or rendered data.
- Location transparency crucial for data storage, data sharing and easy collaborations.
- SRB successfully used for a commercial project in “impossible” production deadline situation dictated by marketing department.
- Collaboration across sites made feasible with SRB

Latency Management Example - ASCI - DOE

- **Demonstrate the ability to load collections at terascale rates**
  - Large number of digital entities
  - Terabyte sized data
- **Optimize interactions with the HPSS High Performance Storage System**
  - Server-initiated I/O
  - Parallel I/O

SRB Latency Management

Remote Proxies, Staging

Data Aggregation Containers

Prefetch

Replication
Server-initiated I/O

Streaming
Parallel I/O

Caching
Client-initiated I/O

ASCI Small Files

- **Ingest a very large number of small files into SRB**
  - time consuming if the files are ingested one at a time
- **Bulk ingestion to improve performance**
  - Ingestion broken down into two parts
    - the registration of files with MCAT
    - the I/O operations (file I/O and network data transfer)
  - Multi-threading was used for both the registration and I/O operations.
- **Sbload was created for this purpose.**
  - reduced the ASCI benchmark time of ingesting ~2,100 files from ~2.5 hours to ~7 seconds.

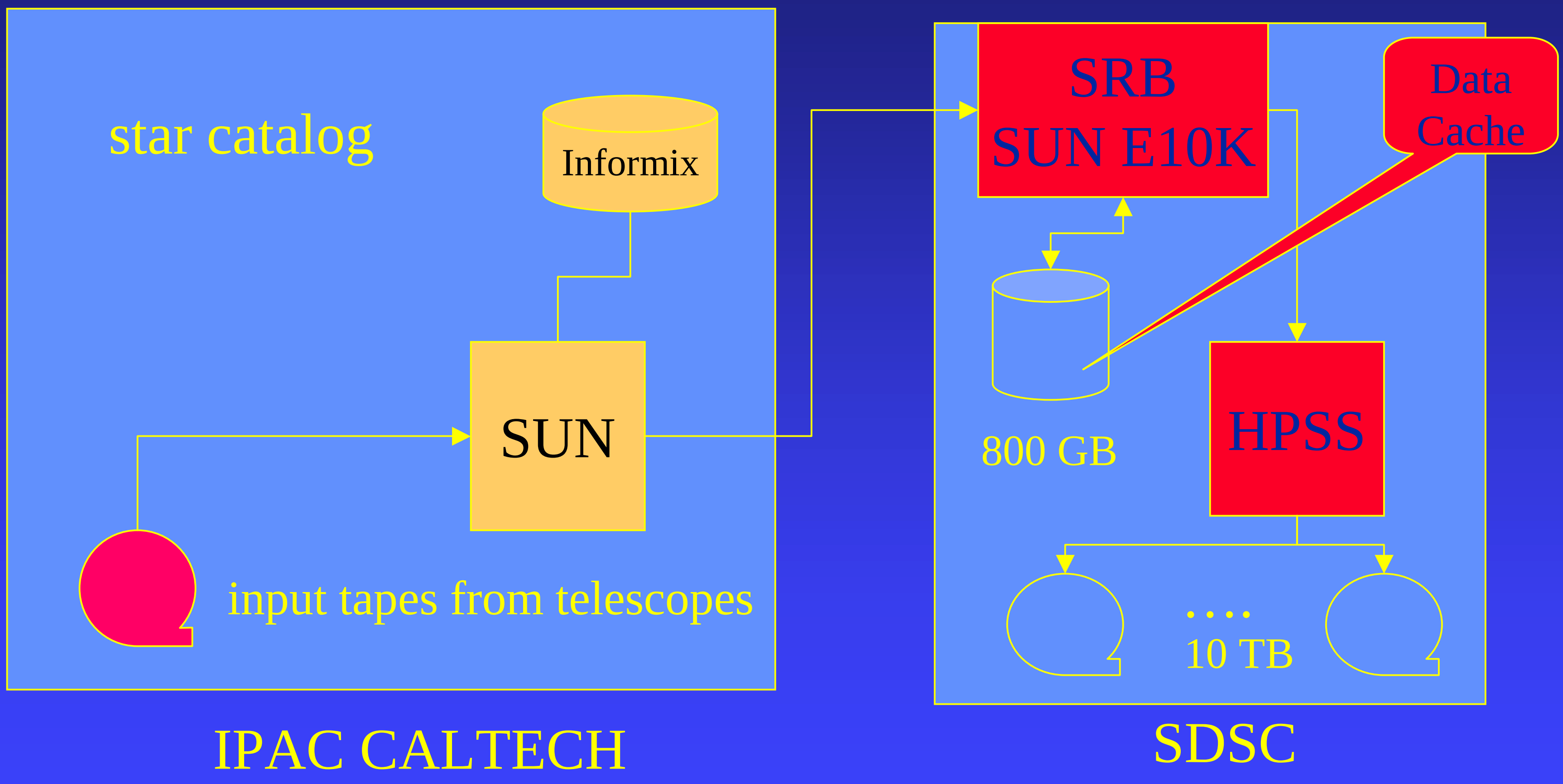
Latency Management
Example - Digital Sky Project

- **2MASS (2 Micron All Sky Survey):**
  - Bruce Berriman, IPAC, Caltech; John Good, IPAC, Caltech, Wen-Piao Lee, IPAC, Caltech
- **NVO (National Virtual Observatory):**
  - Tom Prince, Caltech, Roy Williams CACR, Caltech, John Good, IPAC, Caltech
- **SDSC – SRB :**
  - Arcot Rajasekar, Mike Wan, George Kremenek, Reagan Moore

Digital Sky

Very
Large
Data
Bases

Digital Sky Data Ingestion



Digital Sky - 2MASS

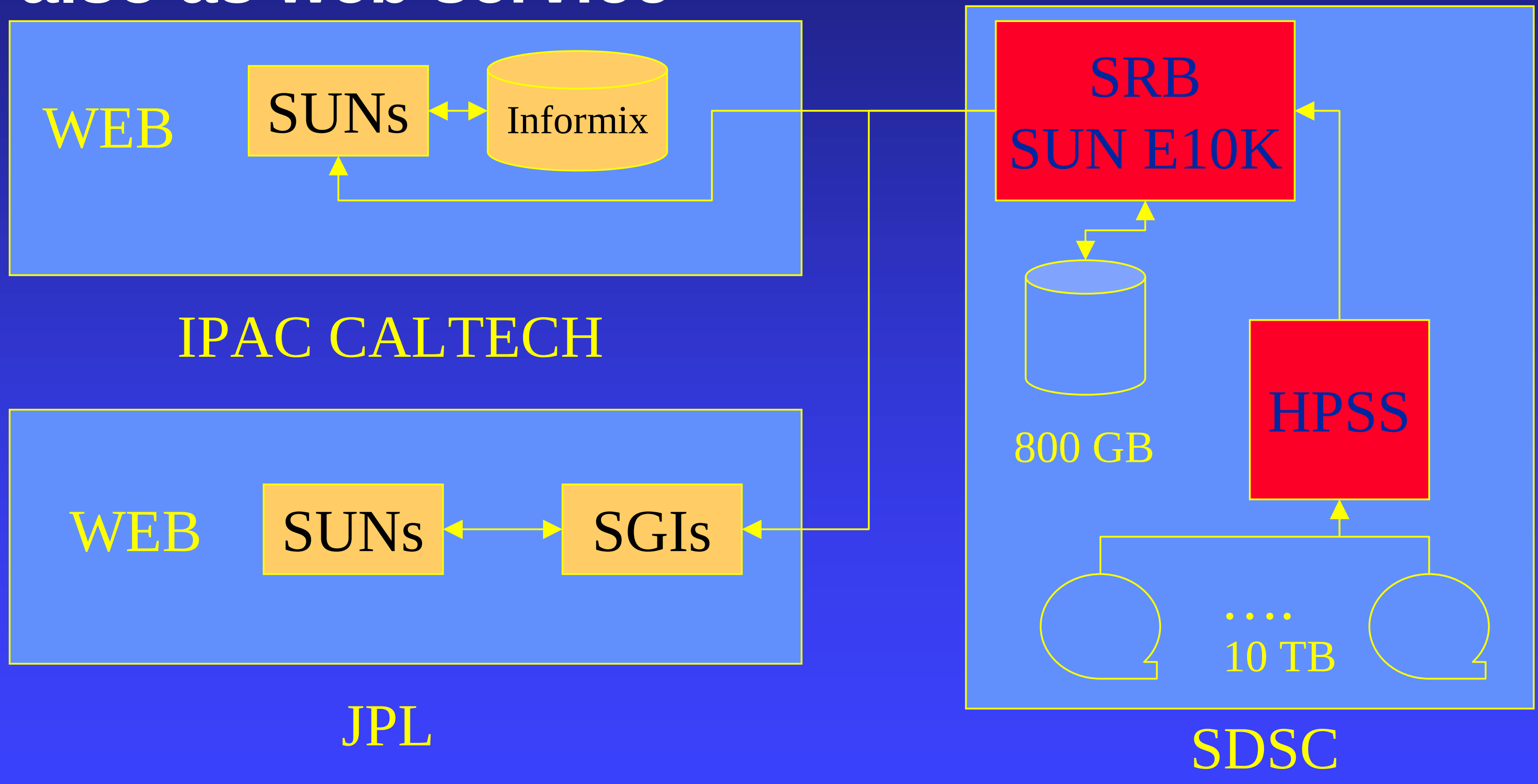
- http://www.ipac.caltech.edu/2mass
- The input data was originally written to DLT tapes in the order seen by the telescope
  - 10 TBytes of data, 5 million files
- Ingestion took nearly 1.5 years - almost continuous reading of tapes retrieved from a closet, one at a time
- Images aggregated into 147,000 containers by SRB

Containers

- **Images sorted by spatial location**
  - Retrieving one container accesses related images
- **Minimizes impact on archive name space**
  - HPSS stores 680 Tbytes in 17 million files
- **Minimizes distribution of images across tapes**
- **Bulk unload by transport of containers**

Digital Sky – “Stars at finger tips”

- Average 3000 images a day – web clients and also as web service



Very Large Data Bases

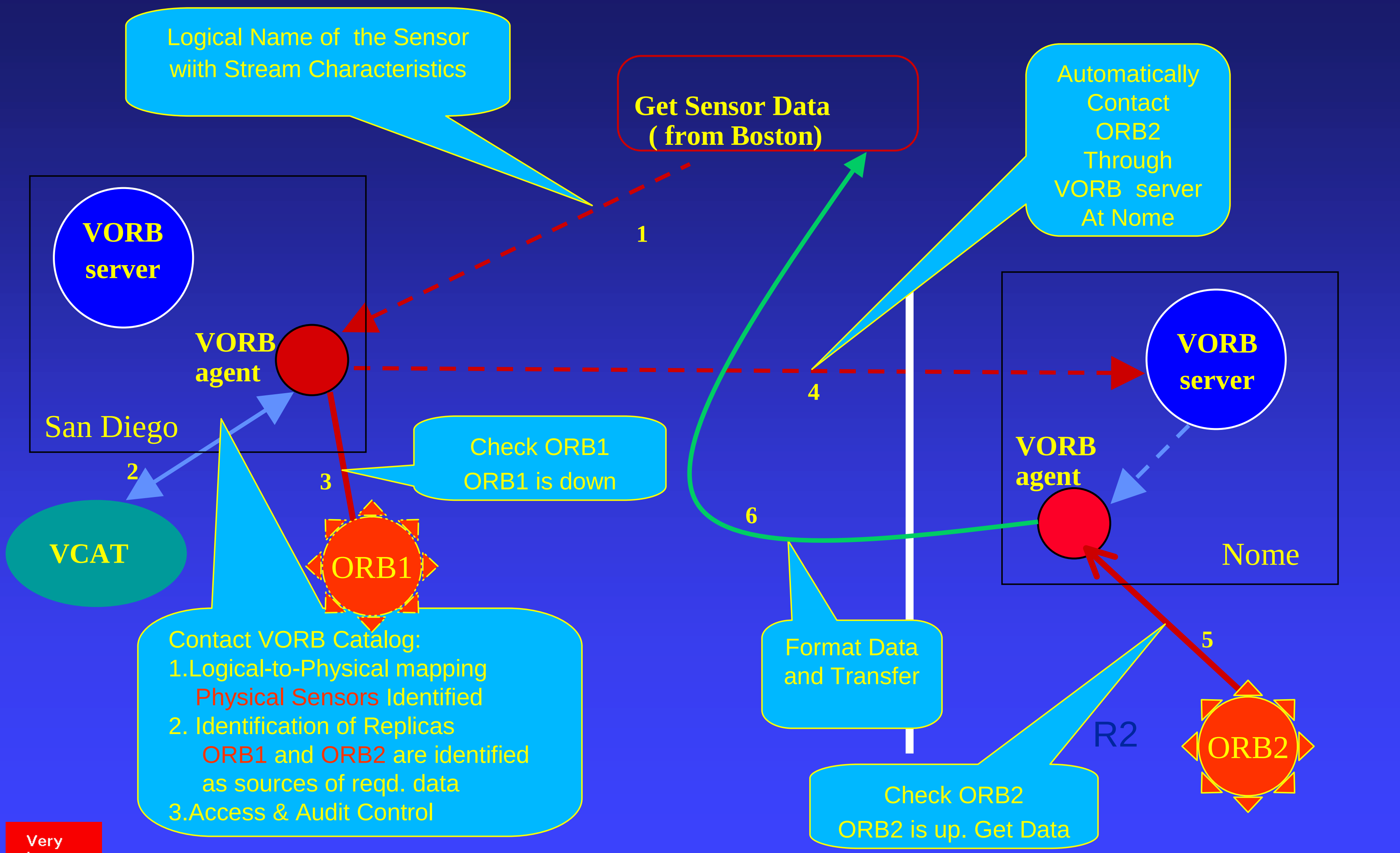
Remote Proxies

- **Extract image cutout from Digital Palomar Sky Survey**
  - Image size 1 Gbyte
  - Shipped image to server for extracting cutout took 2-4 minutes (5-10 Mbytes/sec)
- **Remote proxy performed cutout directly on storage repository**
  - Extracted cutout by partial file reads
  - Image cutouts returned in 1-2 seconds
- **Remote proxies are a mechanism to aggregate I/O commands**

Real-Time Data
Example - RoadNet Project

- Manage interactions with a virtual object ring buffer
- Demonstrate federation of ORBs
- Demonstrate integration of archives, VORBs and file systems
- Support queries on objects in VORBs

Federated VORB Operation



Information Abstraction
Example - Data Assimilation Office

HSI has implemented metadata schema in SRB / MCAT

Origin: host, path, owner, uid, gid, perm_mask, [times]

Ingestion: date, user, user_email, comment

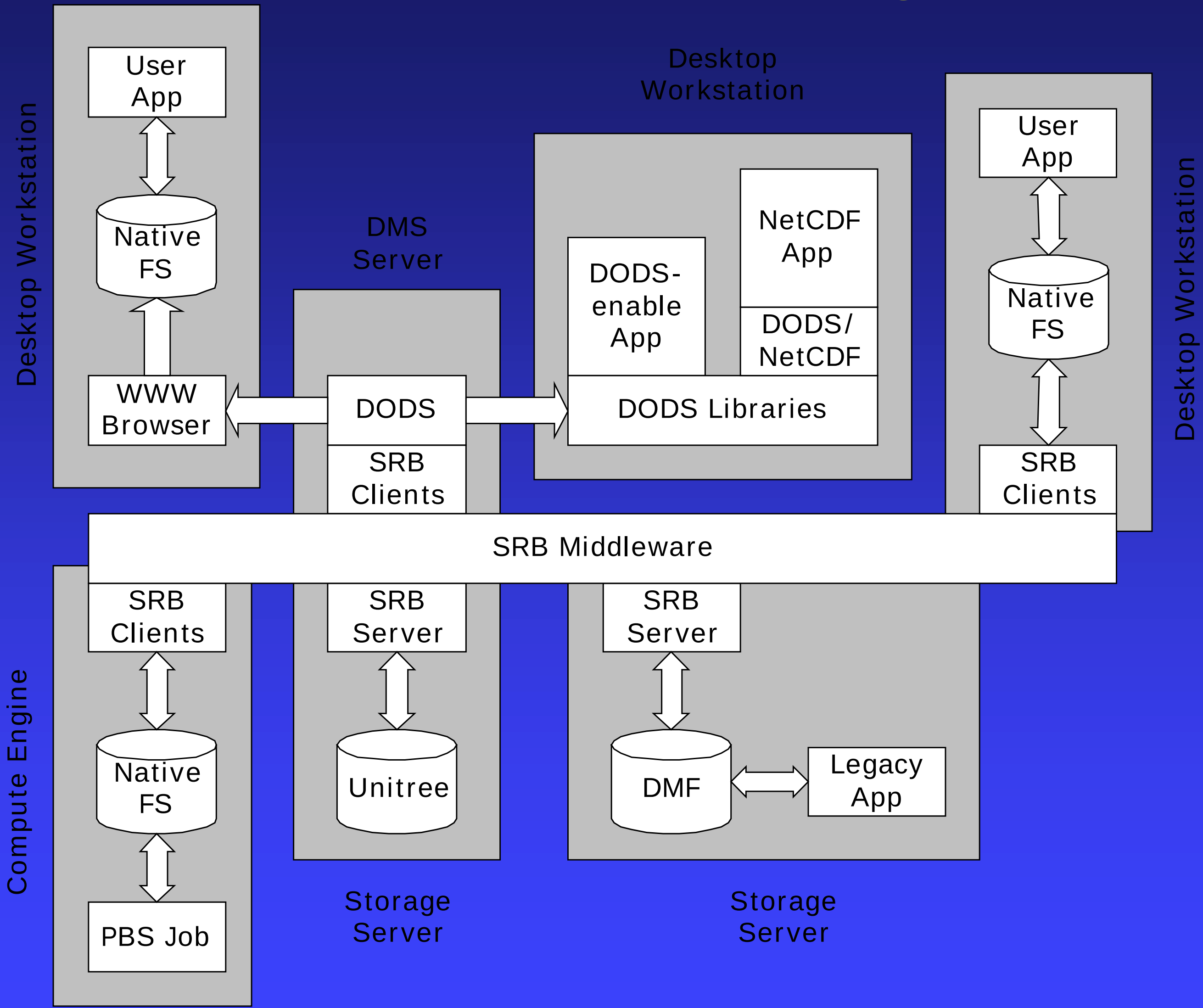
Generation: creator (name, uid, user, gid), host (name, arch, OS name & flags), compiler (name, version, flags), library, code (name, version), accounting data

Data description: title, version, discipline, project, language, measurements, keywords, sensor, source, prod. status, temporal/spatial coverage, location, resolution, quality

Fully compatible with GCMD

HALCYON SYSTEMS, INC.

DODS Access Environment Integration

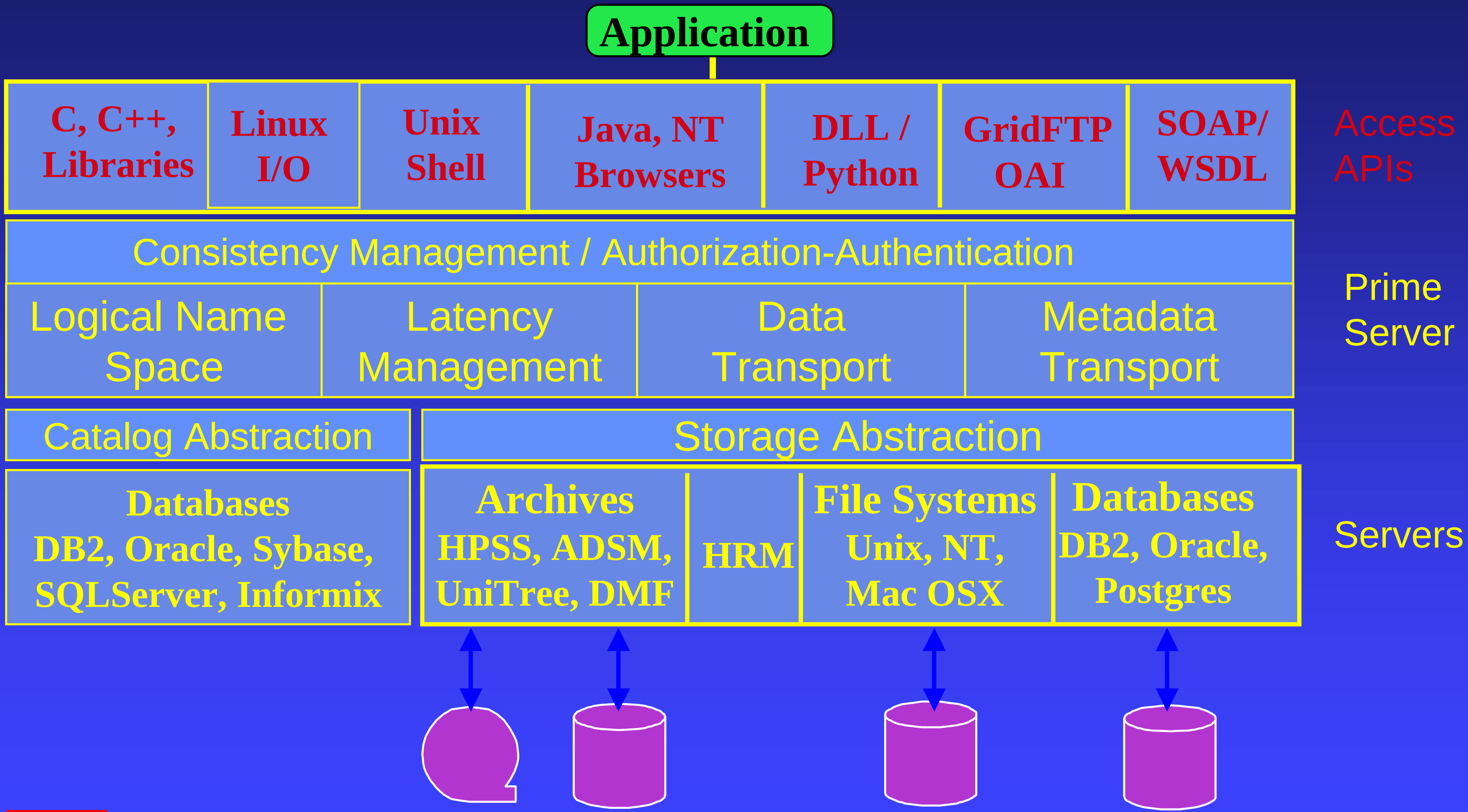


Very Large Data Bases

Data Grid Brick

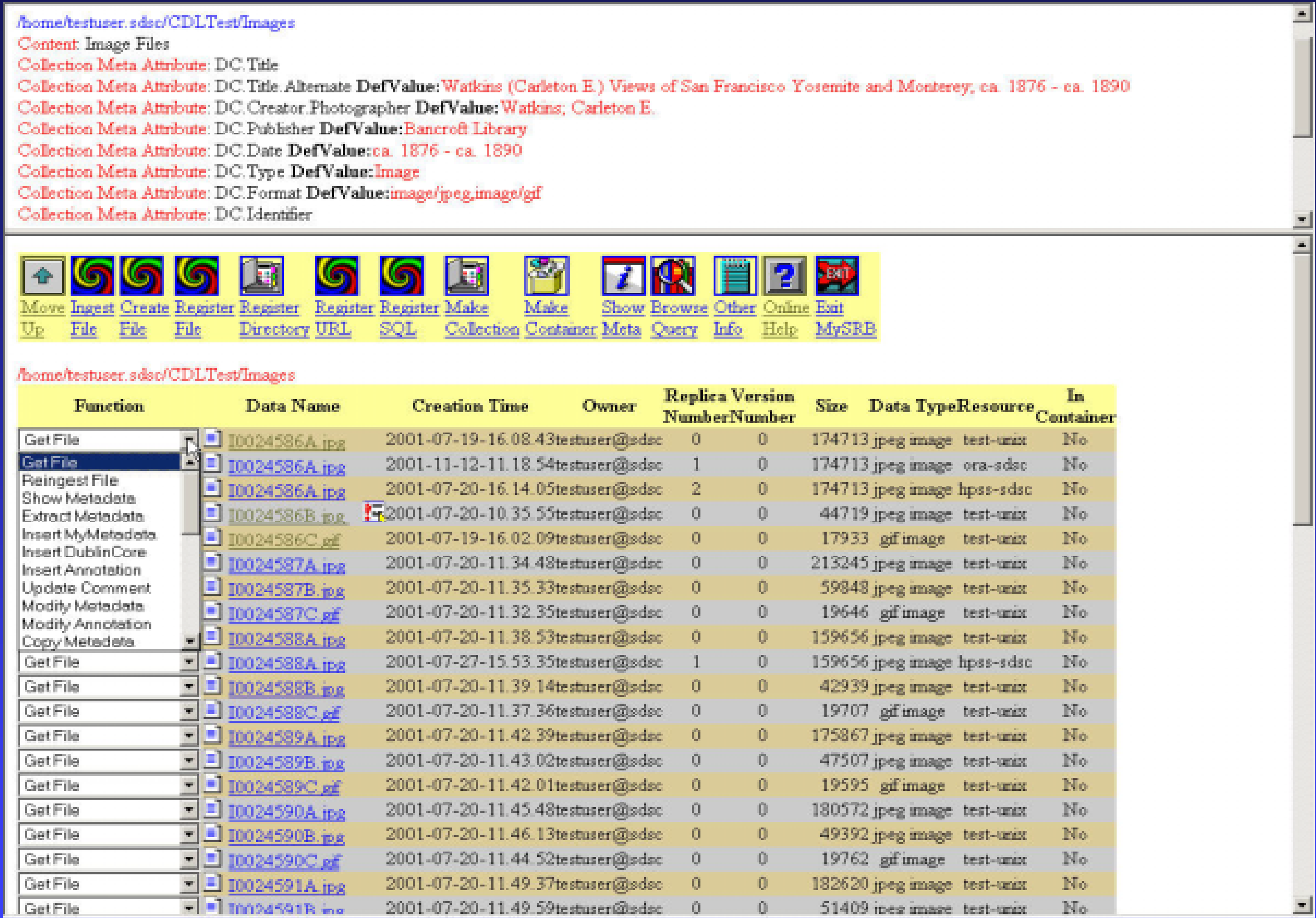
- **Data grid to authenticate users, manage file names, manage latency, federate systems**
  - Intel Celeron 1.7 GHz CPU
  - SuperMicro P4SGA PCI Local bus ATX mainboard
  - 1 GB memory (266 MHz DDR DRAM)
  - 3Ware Escalade 7500-12 port PCI bus IDE RAID
  - 10 Western Digital Caviar 200-GB IDE disk drives
  - 3Com Etherlink 3C996B-T PCI bus 1000Base-T
  - Redstone RMC-4F2-7 4U ten bay ATX chassis
  - Linux operating system
- **Cost is $2,200 per Tbyte plus tax**
  - Gig-E network switch costs $500 per brick
  - Effective cost is about $2,700 per TByte

SDSC Storage Resource Broker & Meta-data Catalog - Access Abstraction



Tutorial Part I Outline

- **Concepts**
  - Introduction to Grid Computing
  - Proliferation of Data Grids
  - Data Grid Concepts
- **Practice**
  - Real life use cases SDSC Storage Resource Broker (SRB)
  - Hands on Session
- **Research**
  - Active Datagrid Collections
  - Data Grid Management Systems (DGMS)
  - Open Research Issues



/home/testuser.sdsc/CDLTest/Images
Content: Image Files
Collection Meta Attribute: DC.Title
Collection Meta Attribute: DC.Title.Alternate **DefValue:** Watkins (Carleton E.) Views of San Francisco Yosemite and Monterey, ca. 1876 - ca. 1890
Collection Meta Attribute: DC.Creator.Photographer **DefValue:** Watkins; Carleton E.
Collection Meta Attribute: DC.Publisher **DefValue:** Bancroft Library
Collection Meta Attribute: DC.Date **DefValue:** ca. 1876 - ca. 1890
Collection Meta Attribute: DC.Type **DefValue:** Image
Collection Meta Attribute: DC.Format **DefValue:** image/jpeg,image/gif
Collection Meta Attribute: DC.Identifier

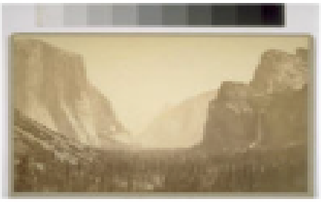
Move Up | Ingest File | Create File | Register File | Register Directory | Register URL | Register SQL | Make Collection | Make Container | Show Meta | Browse Query | Other Info | Online Help | Exit MySRB

/home/testuser.sdsc/CDLTest/Images

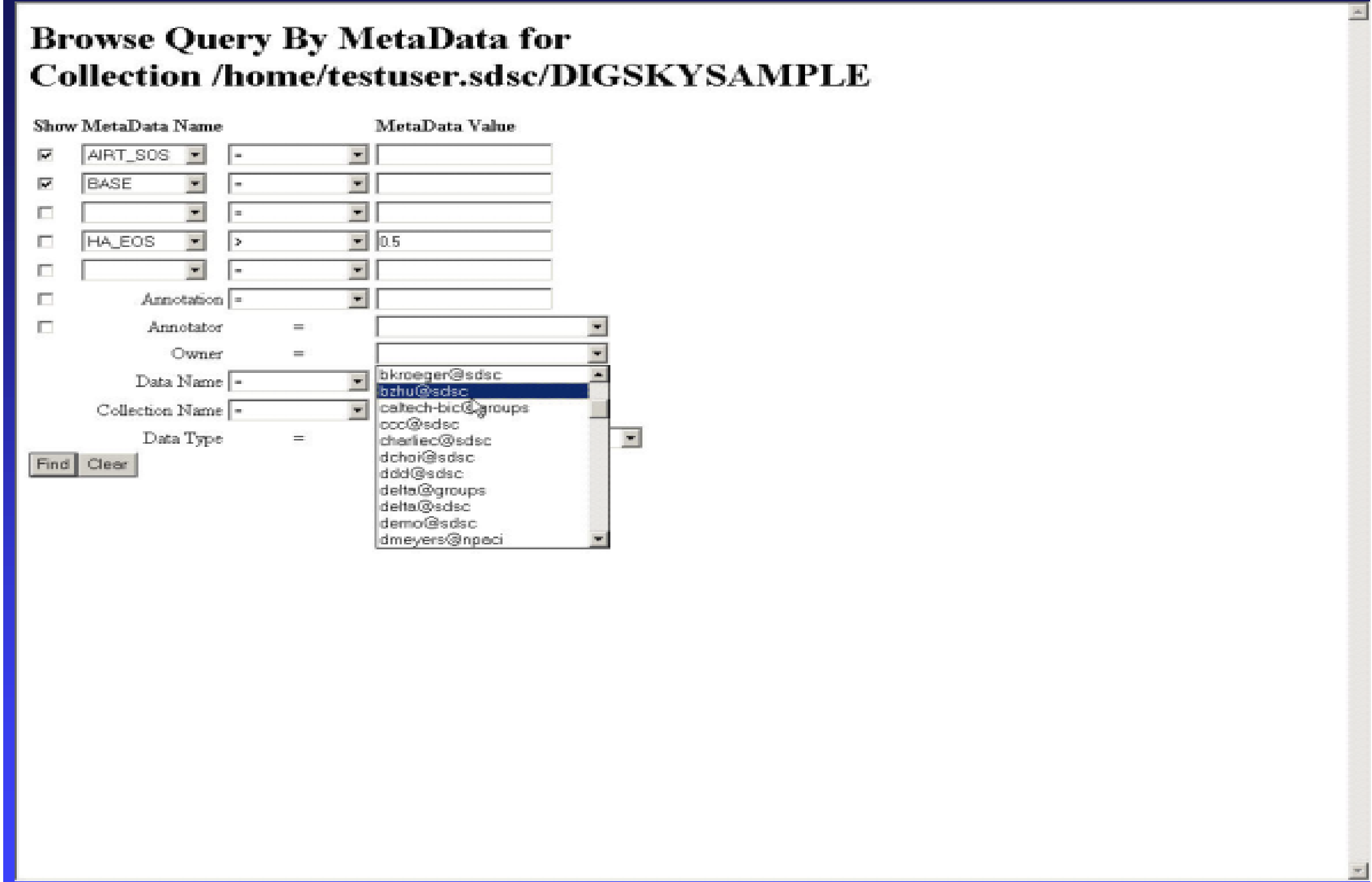
Get File
Reingest File
Show Metadata
Extract Metadata
Insert MyMetadata
Insert DublinCore
Insert Annotation
Update Comment
Modify Metadata
Modify Annotation
Copy Metadata

Function	Data Name	Creation Time	Owner	Replica Number	Version Number	Size	Data Type	Resource	In Container
Get File	I0024586A.jpg	2001-07-19-16.08.43	testuser@sdsc	0	0	174713	jpeg image	test-unix	No
	I0024586A.jpg	2001-11-12-11.18.54	testuser@sdsc	1	0	174713	jpeg image	ora-sdsc	No
	I0024586A.jpg	2001-07-20-16.14.05	testuser@sdsc	2	0	174713	jpeg image	hpss-sdsc	No
	I0024586B.jpg	2001-07-20-10.35.55	testuser@sdsc	0	0	44719	jpeg image	test-unix	No
	I0024586C.gif	2001-07-19-16.02.09	testuser@sdsc	0	0	17933	gif image	test-unix	No
	I0024587A.jpg	2001-07-20-11.34.48	testuser@sdsc	0	0	213245	jpeg image	test-unix	No
	I0024587B.jpg	2001-07-20-11.35.33	testuser@sdsc	0	0	59848	jpeg image	test-unix	No
	I0024587C.gif	2001-07-20-11.32.35	testuser@sdsc	0	0	19646	gif image	test-unix	No
	I0024588A.jpg	2001-07-20-11.38.53	testuser@sdsc	0	0	159656	jpeg image	test-unix	No
Get File	I0024588A.jpg	2001-07-27-15.53.35	testuser@sdsc	1	0	159656	jpeg image	hpss-sdsc	No
Get File	I0024588B.jpg	2001-07-20-11.39.14	testuser@sdsc	0	0	42939	jpeg image	test-unix	No
Get File	I0024588C.gif	2001-07-20-11.37.36	testuser@sdsc	0	0	19707	gif image	test-unix	No
Get File	I0024589A.jpg	2001-07-20-11.42.39	testuser@sdsc	0	0	175867	jpeg image	test-unix	No
Get File	I0024589B.jpg	2001-07-20-11.43.02	testuser@sdsc	0	0	47507	jpeg image	test-unix	No
Get File	I0024589C.gif	2001-07-20-11.42.01	testuser@sdsc	0	0	19595	gif image	test-unix	No
Get File	I0024590A.jpg	2001-07-20-11.45.48	testuser@sdsc	0	0	180572	jpeg image	test-unix	No
Get File	I0024590B.jpg	2001-07-20-11.46.13	testuser@sdsc	0	0	49392	jpeg image	test-unix	No
Get File	I0024590C.gif	2001-07-20-11.44.52	testuser@sdsc	0	0	19762	gif image	test-unix	No
Get File	I0024591A.jpg	2001-07-20-11.49.37	testuser@sdsc	0	0	182620	jpeg image	test-unix	No
Get File	I0024591B.jpg	2001-07-20-11.49.59	testuser@sdsc	0	0	51409	jpeg image	test-unix	No

DC.Coverage.PeriodName.: ca. 1876 - ca. 1890
DC.Coverage.PlaceName.: San Francisco, California
DC.Rights.URL: Rights Document
Resolution: hi-res
color: beige



ThumbNail Image: srb Image
MediumSize Image: /home/testuser.sdsc/CDLTest/Images/I0024586B.jpg srbObject
Image Size: Large


Browse Query By MetaData for Collection /home/testuser.sdsc/DIGSKYSAMPLE
Show MetaData Name
MetaData Value
AIRT_SOS
BASE
HA_EOS
0.5
Annotation
Annotator
Owner
Data Name
Collection Name
Data Type
Find
Clear
bkroeger@sdsc
bzhu@sdsc
caltech-bio@groups
ccc@sdsc
charliec@sdsc
dchoi@sdsc
ddd@sdsc
delta@groups
delta@sdsc
demo@sdsc
dmeyers@npaci

Tutorial Part I Outline

- **Concepts**
  - Introduction to Grid Computing
  - Proliferation of Data Grids
  - Data Grid Concepts
- **Practice**
  - Real life use cases SDSC Storage Resource Broker (SRB)
  - Hands on Session
- **Research**
  - Active Datagrid Collections
  - Data Grid Management Systems (DGMS)
  - Open Research Issues

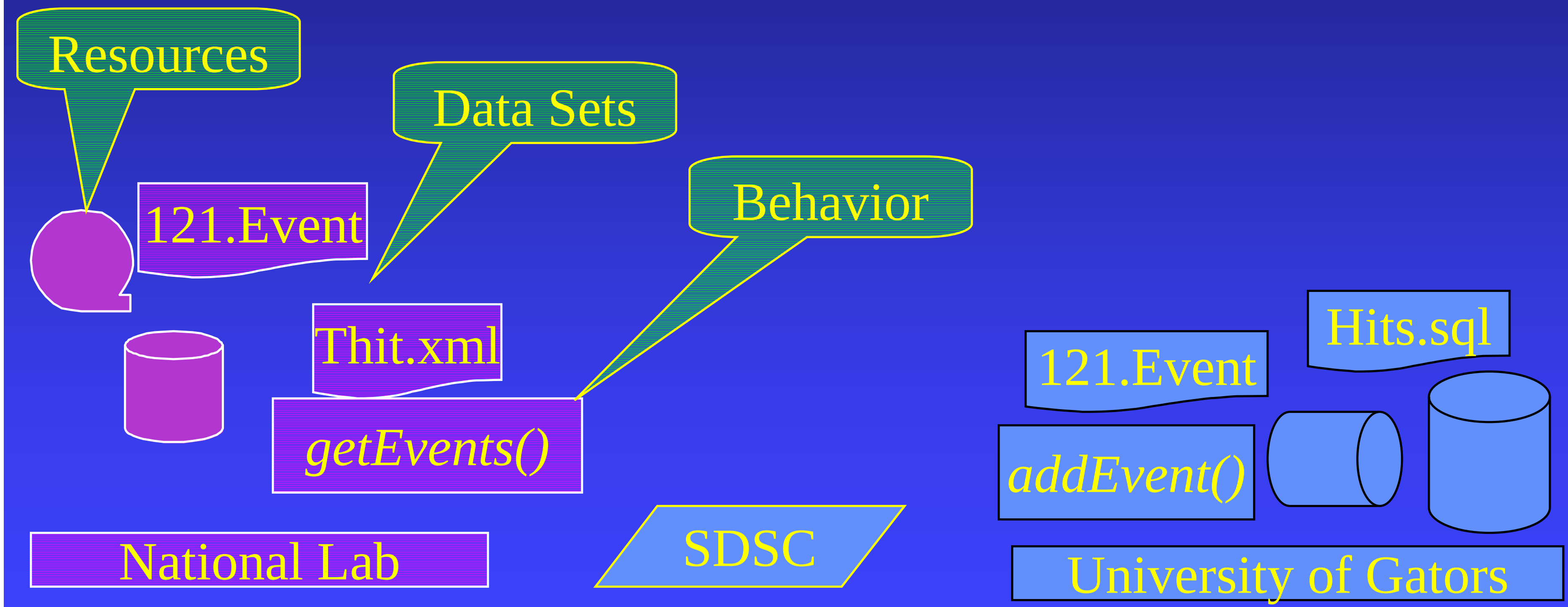
Datagrid Management System (DGMS)

- **DGMS manages**
  - State information of the datagrid collections (data)
  - Knowledge of events, rules and services (data behavior)
  - Collaborative communities (data users and resources)
- **Differences from DBMS**
  - Manages “community-owned” unstructured data along with its behavior and inter-organizational resources
  - Logical organization has the (logical) resources where the data be present (hidden in DBMS)
  - Basic unit = *Active Datagrid Collection*
  - Also uses concepts got from decades of DB Research

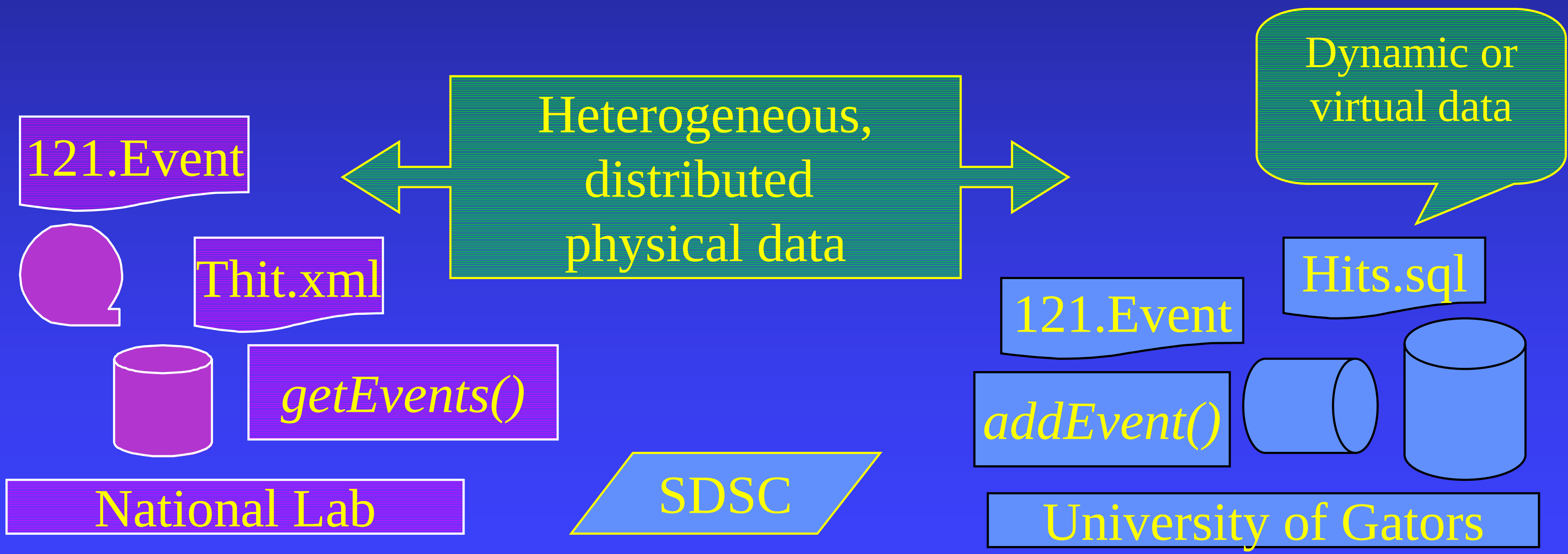
DGMS Philosophy

- **Collective view of**
  - Inter-organizational data
  - Operations on datagrid space
- **Local autonomy and global state consistency**
- **Collaborative datagrid communities**
  - Multiple administrative domains or “Grid Zones”
- **Self-describing and self-manipulating data**
  - Horizontal and vertical behavior
  - Loose coupling between data and behavior (dynamically)
  - Relationships between a digital entity and its *Physical locations*, *Logical names*, *Meta-data*, *Access control*, *Behavior, “Grid Zones”.*

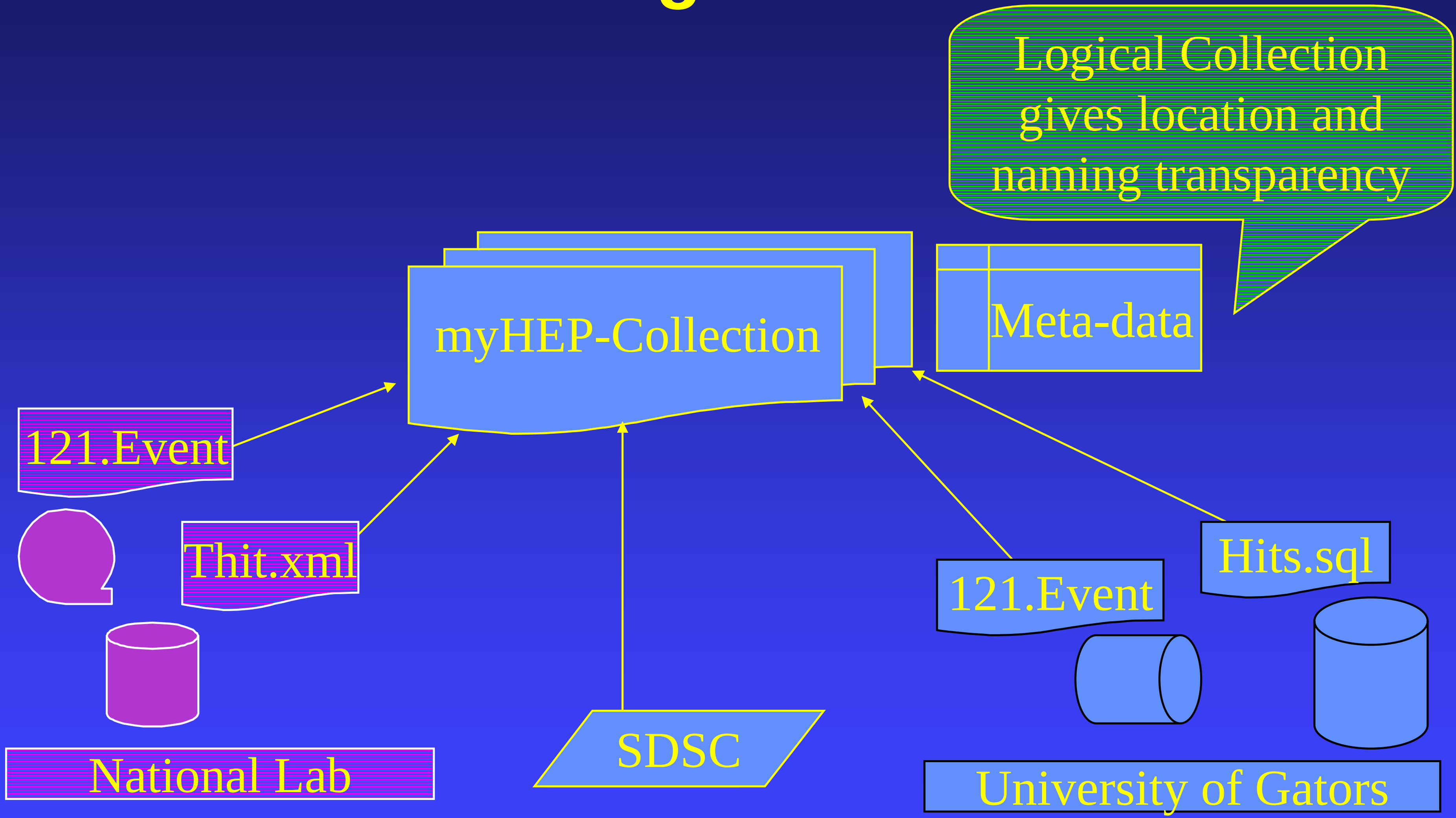
Active Datagrid Collections



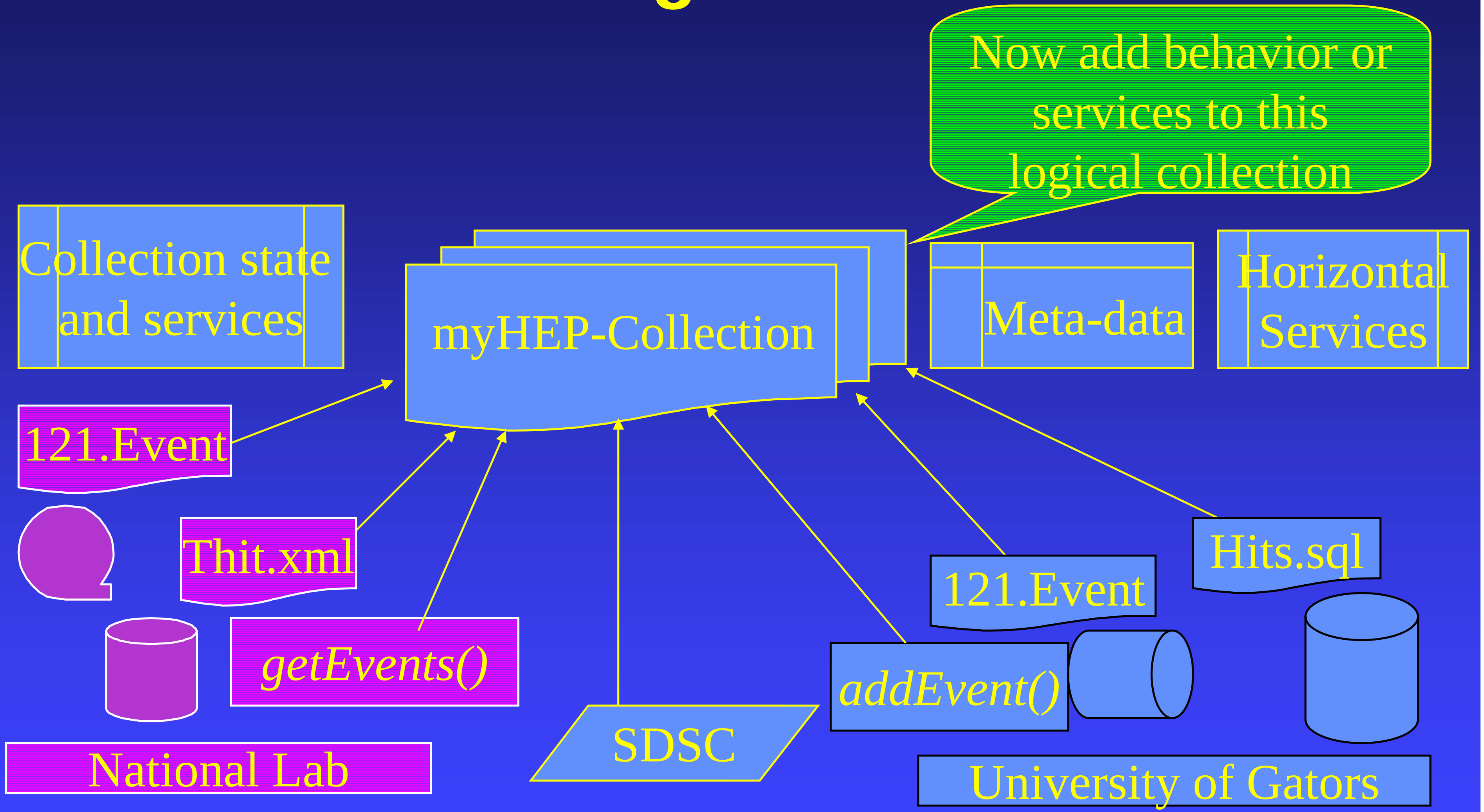
Active Datagrid Collections



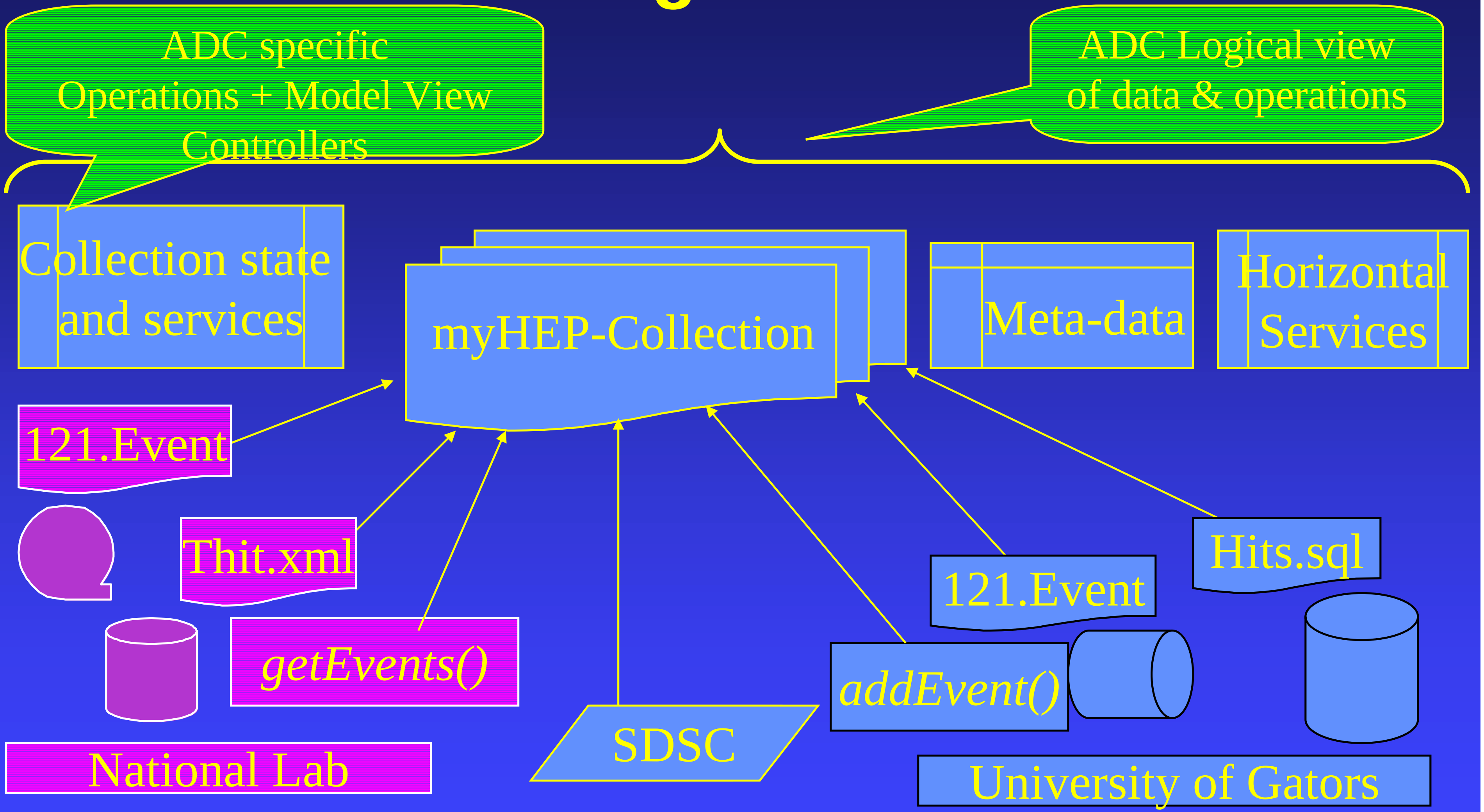
Active Datagrid Collections



Active Datagrid Collections

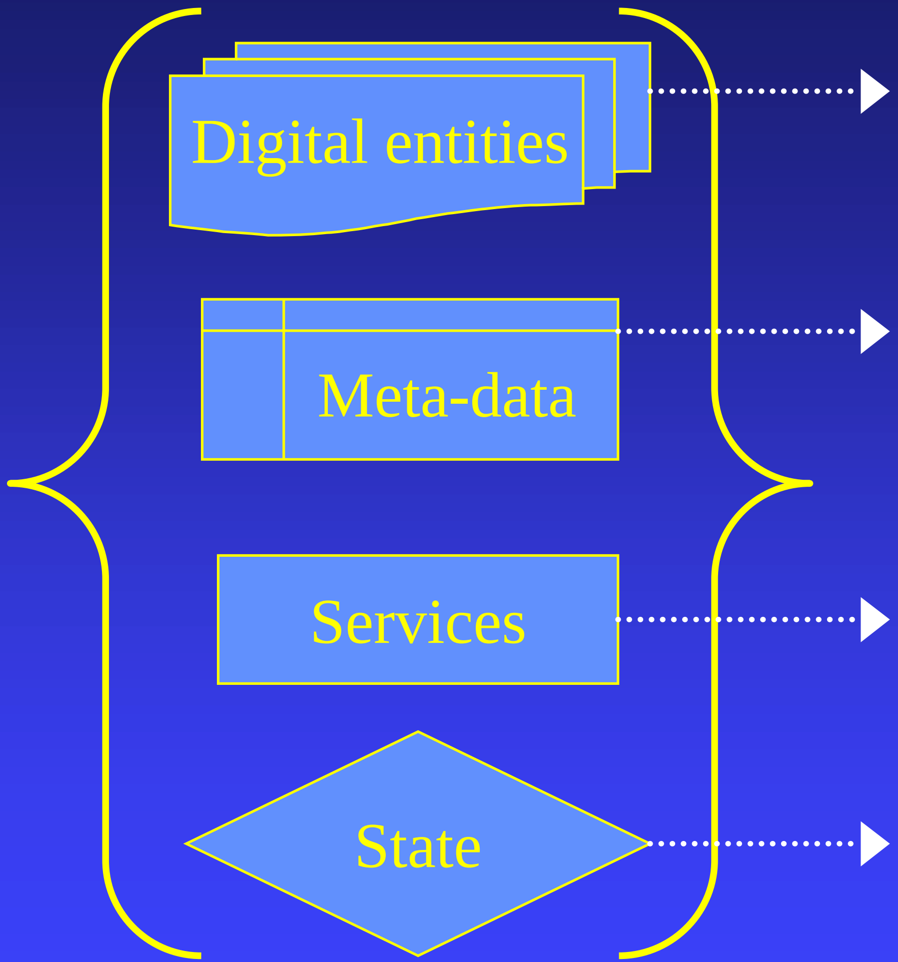


Active Datagrid Collections



Very
Large
Data
Bases

Active Datagrid Collections



Physical and virtual data present in the datagrid

Standardized schema with domain specific schema extensions

Horizontal datagrid services and vertical domain specific services

Events, collective state, mappings to domain services to be invoked

Active Datagrid Collections

- ***Logical set consisting of related digital entities and references to their collective behavior for self-organization and manipulation of the data.***
- **Basic unit or data model managed in DGMS**

Collections facilitate the transparencies and abstractions required to manage data in grids and inter-organizational enterprises

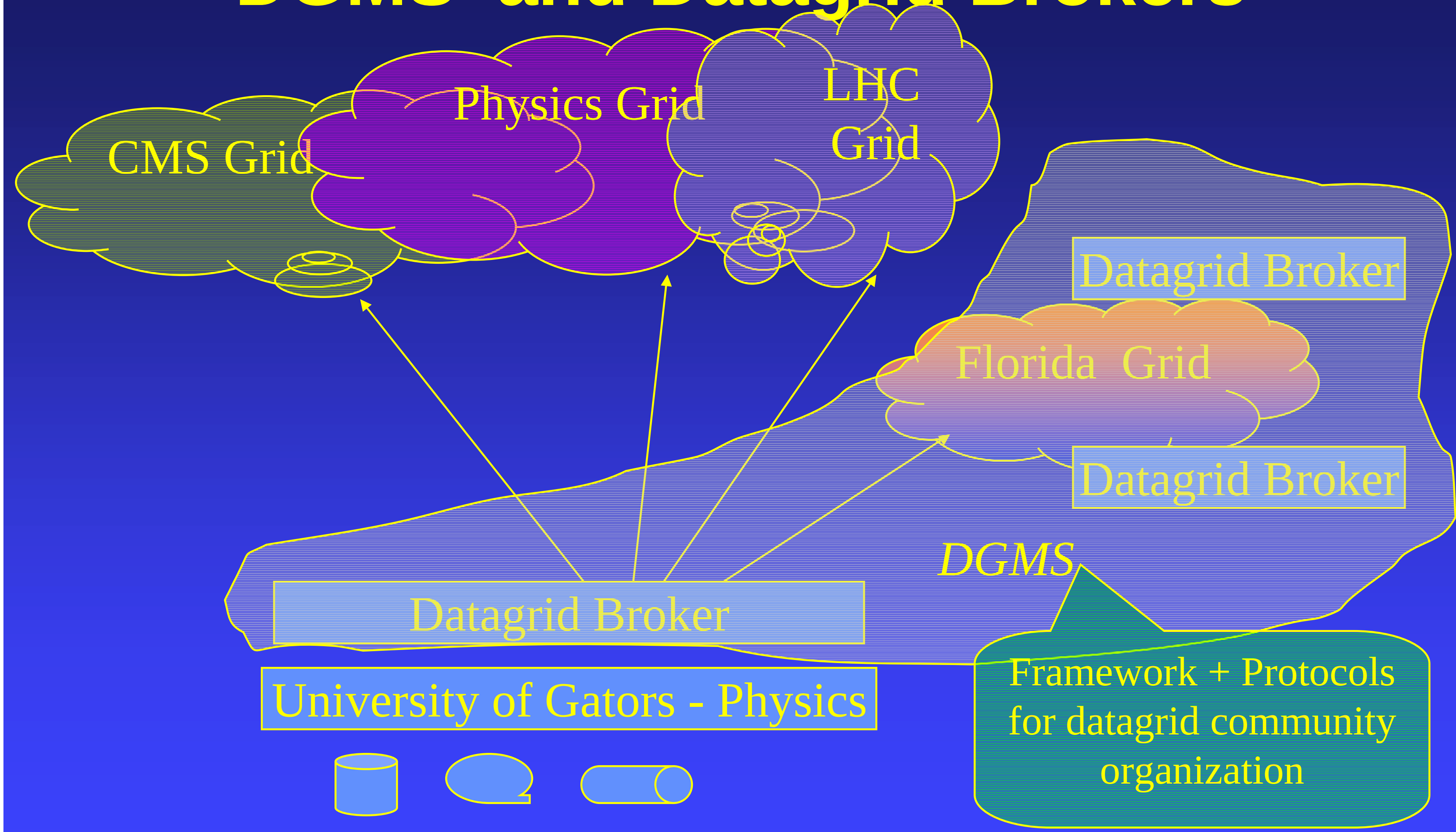
DGMS

- ***Datagrid Management System consists of a set of services (protocols) and a hierarchical framework for:***
  - *Confluence of datagrid communities*
  - *Coordinated sharing of inter-organizational information storage space and active datagrid collections*

Datagrid Broker

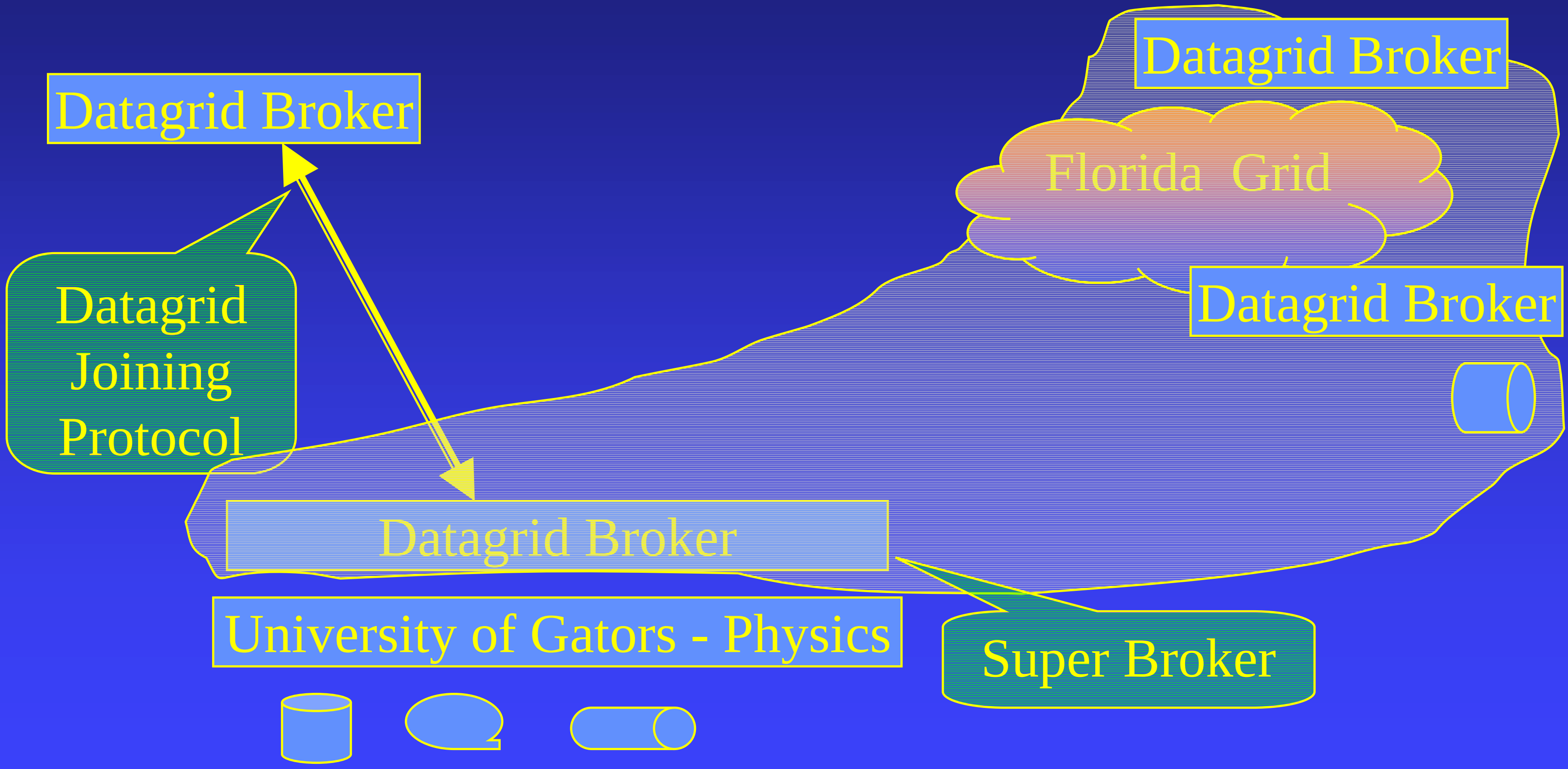
- ***A datagrid broker acts as an agent for an administrative domain in a DGMS framework.***
- **Datagrid communities**
  - formed by confluence of datagrid brokers
  - Peer2peer network of brokers resulting in DGMS
- **Datagrid brokers facilitate**
  - sharing of services and data as components of active datagrid collections in the datagrid.
  - Ensure the users in its domain are benefited by participating in datagrid communities.

DGMS and Datagrid Brokers



Very
Large
Data
Bases

Datagrid Brokerage Protocols

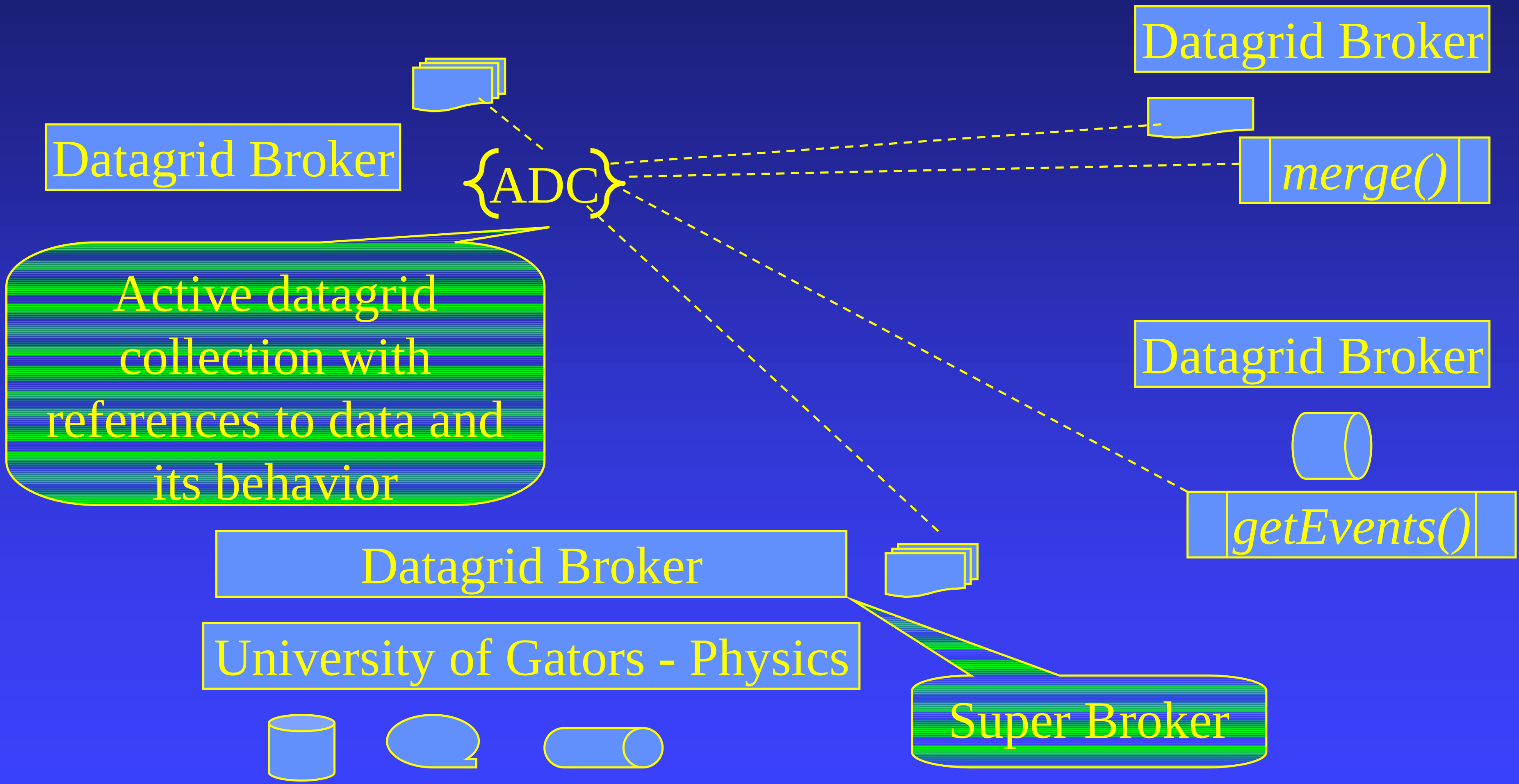


Datagrid Brokerage Protocols

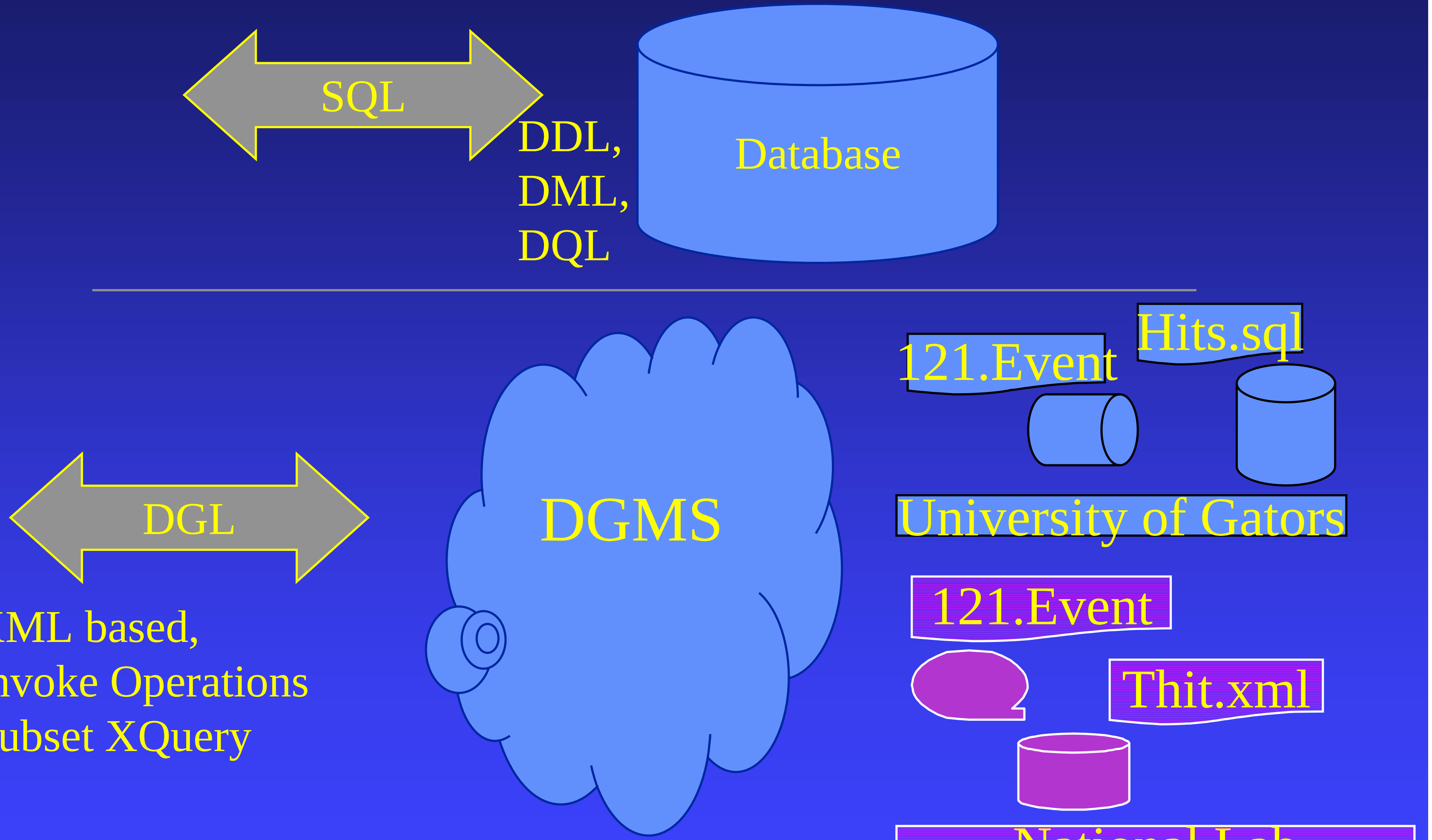
Datagrid Brokerage Protocols (org)

- **Organizing datagrid community**
- **Managing the inter-organizational data**
- **Datagrid Operations**
  - Converted into datagrid brokerage protocols
  - Protocols implemented as services by the datagrid brokers
- **Hence, DGMS is nothing but these datagrid brokers which form these communities and the protocols (services) which operate on the collections**

Datagrid Brokerage Protocols (data)



Need for Standard DGL



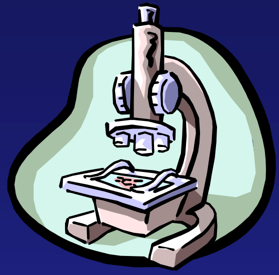
Data Grid Language

- **XML based asynchronous protocol**
  - Describe data sets, collections, datagrid operations, ...
  - Access and manage data grids, data flow pipelines
  - Query on data resource (based on W3C XQuery)
- **Facilitates Grid Workflow**
  - Sharing of granular state information about execution of each datagrid operation amongst different processes or services
- **Implementation Status**
  - Reference Implementation by SDSC Matrix Project
  - On top of SRB protocol stack as W3C SOAP Web Service

Data Grid Language

- **Datagrid Request**
  - Asynchronous requests for data/process-flow in datagrids
  - Requests are either a *Transaction* or a *Status Query*
  - Each Transaction consists of one or more *Flows*
  - Each Flow consists of one ore more datagrid operations
  - Datagrid operation = data transformation or data query
  - A flow can be executed sequential or parallel
- **Datagrid Response**
  - Either *Transaction Acknowledgement* or *Status Response*
  - Status Response contains the results of a Transaction

Data ⇨ Discovery



New data

Digital entities

Meta-data

Services

State

updates relationships among data in collections

Services invoked to analyze new relationships

DGMS applications get notified of state updates

Data ⇨ Discovery (Issues)

- **“More data; More discovery” Fermi National Lab**
- **DGMS applications to automate knowledge discovery**
  - Work flow Management Systems (WfMS) subscribe to updates in datagrid collections
  - Trigger like mechanisms on this large scale dynamic and distributed data is a needed
  - Dynamic rule description and execution based on events
  - Semantic Mediation of datagrid collections
  - [SDSC Grid Enabled Mediation (GeMS)]

Very Large Data Bases

DGMS Research Issues

- **Self-organization of datagrid communities**
  - Using knowledge relationships across the datagrids
  - Inter-datagrid operations based on semantics of data in the communities (different ontologies)
- **High speed data transfer**
  - Terabyte to transfer - TCP/IP not final answer
  - Protocols, routers needed
- **Latency Management**
  - Data source speed >> data sink speed
- **Datagrid Constraints**
- **Data placement and scheduling**
  - How many replicas, where to place them…

Other Grid Software

- **Legion (Avaki)**
  - Grid as a single virtual machine
- **Condor**
  - High Throughput Computing (HTC)
- **Globus**
  - Synonymous with Computational Grids
  - Data Handling Capabilities
  - GRAM (1000s), GridFTP (1000s), GSI, MDS, RLS, MCS
- **Entropia**
  - PC Grid, P2P
- **IBM**
- **SUN**
- **HP**

Part I Summary

- **Grids are evolving**
  - coming soon to a domain near you
- **DGMS**
  - Coordinate collaborative management of inter-organizational information storage using Active Datagrid Collections
- **Tools are available from research and academia.**
  - Industry getting involved.
  - SDSC SRB provides abstraction mechanisms required to implement data grids, digital libraries, persistent archives
- **Open Research issues for**
  - Distributed databases, Information management and Semantic web researchers

Part II: Grid Services for Structured Data

Paul Watson
University of Newcastle, UK
Paul.Watson@ncl.ac.uk
Norman Paton
University of Manchester, UK
norm@cs.man.ac.uk

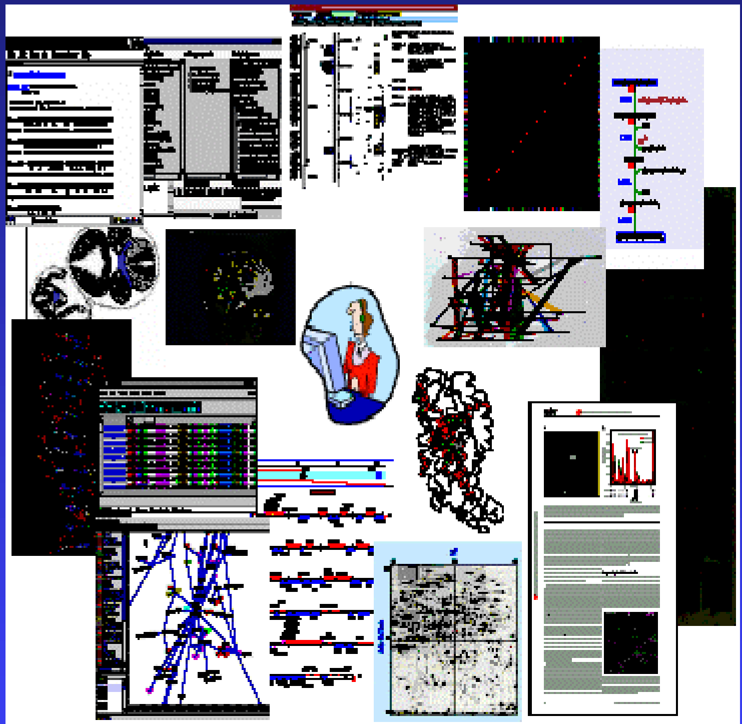
Part II a: Requirements & Grid Services

- **Motivation**
  - why is structured data important for Grid applications?
  - examples from Grid projects
- **Requirements for Grid Data Services**
  - Requirements generic across all services
  - Requirements specific to structured data services
- **Grid Services**
  - Brief history of Grid middleware
  - Open Grid Service Architecture
    - Web Services
    - Grid Services

Motivation

- **Structured data is important to Grid applications**
  - data
  - metadata
- **Level of structure varies**
  - Relational/Object DBs
  - XML
  - Structured files

Example: Bioinformatics



- Large quantities of biological data
- Different kinds of data
- Data sources are scattered and autonomous

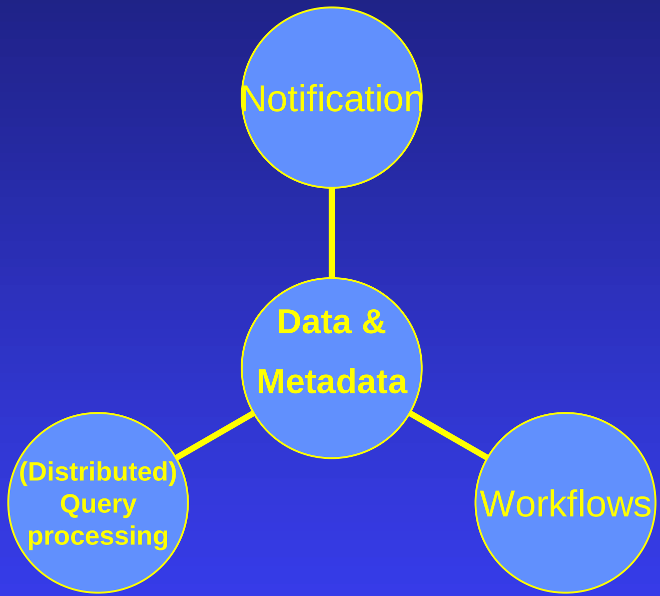
- We will take examples from one project that is aiming to build a Grid application to support bioinformaticians: myGrid (mygrid.man.ac.uk)

Very Large Data Bases

Bioinformatics: myGrid project data

- **Types of structured data**
  - Biological data
  - Provenance
    - Execution of biological workflows generates a provenance record
    - query to ask useful questions….
      - what experiments were run on this data?
      - how was this data derived?
      - what are the top 20 bio-services used by my organisation?
  - Annotation
    - users opinions on interpretation of data

What does myGrid do with the data?



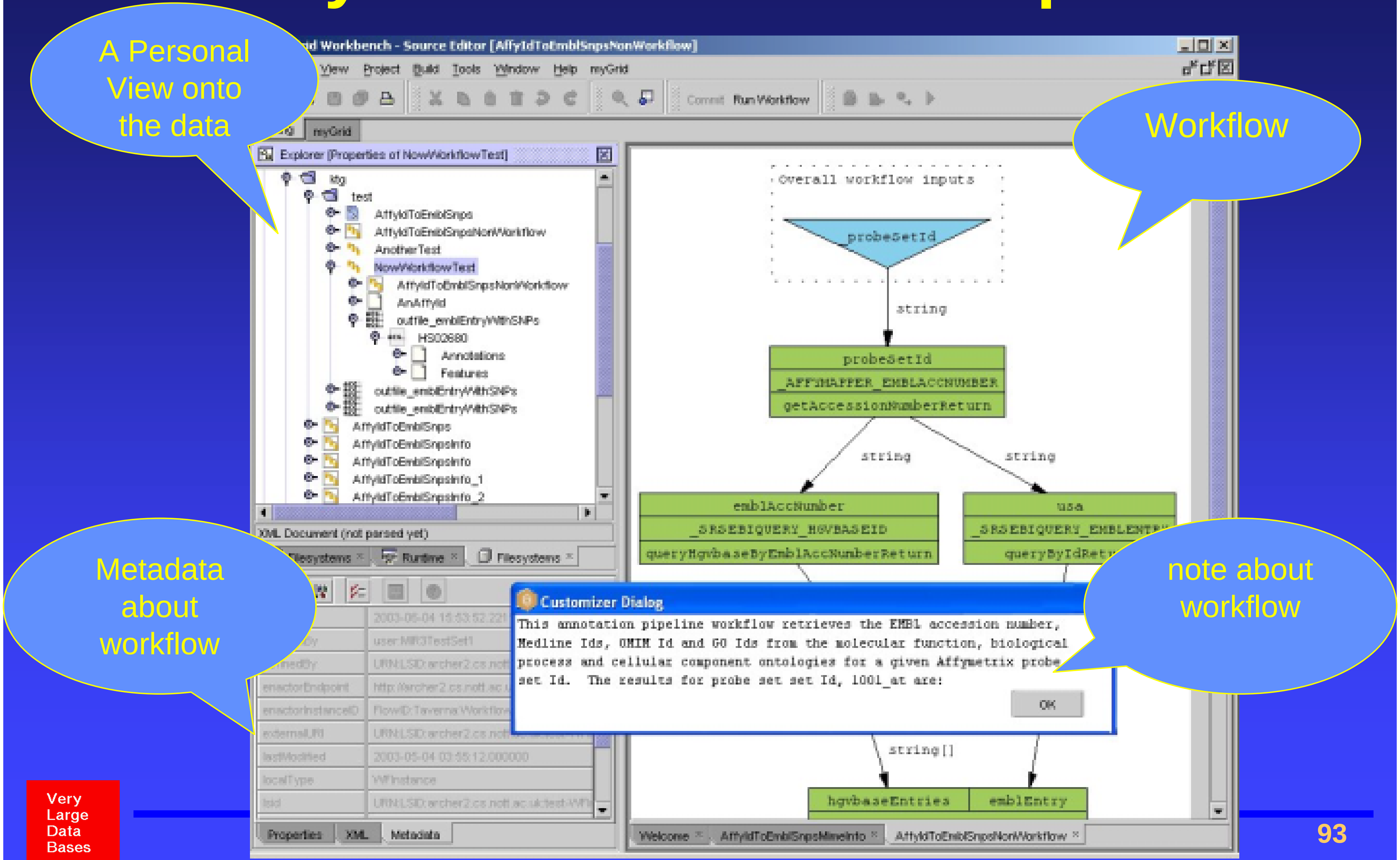
- **Biological databases and computations are represented as services**
- **(Distributed) Queries**
  - extract information
  - integration of data from distributed sources
- **Notification**
  - tell users when data/tools have changed
- **Workflows**
  - combine services

Very
Large
Data
Bases

Requirements

- **A key driver for requirements is that applications combine data access with computation**
  - Computation + Data

myGrid Workflow example



Categories of Requirements

- **The need to integrate computation and data access means data services cannot be designed in isolation**
- **Therefore there are two categories of requirements…**
  - those generic across all Grid services, allowing databases to be first-class components
  - those specific to Grid-enabled data services, allowing databases to be exploited in Grid applications
- **These are considered in turn…**

Generic Grid Requirements

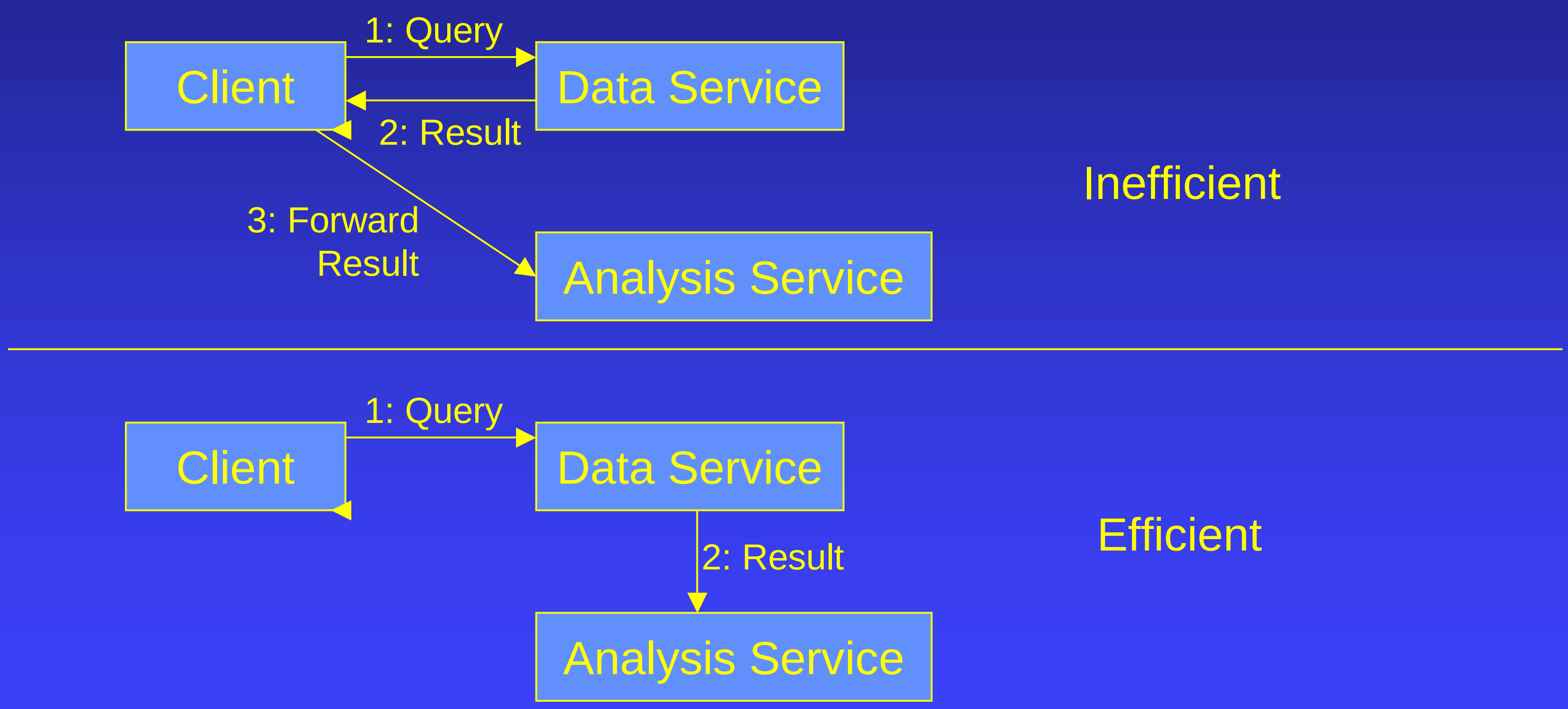
- **Allow compute and data components to be seamlessly combined, e.g.**
  - high performance data transfer (more later)
  - scheduling (more later)
  - security
  - accounting
  - management

Generic Grid requirements: High Performance Data Transfer

- **A database may deliver huge amounts of data to a remote computation for analysis**
- **Requirements:**
  - efficient communication of data
    - flow control
  - efficient encoding
  - direct routing of results….

Very
Large
Data
Bases

Direct Routing of Results



Generic Grid requirements: Scheduling

- **co-scheduling data service with other components of a distributed application**
- **it would be helpful if data service provided cost estimates to assist scheduling, e.g.**
  - for data-source selection (e.g. when replicas are available)
  - to estimate computational requirements (e.g. if we know a query will generate R rows then can schedule space and time on consumer)

Database Specific Requirements

- **Grid applications will require at least the same functionality, tools and properties as other database applications**
  - query, update, programming, indexing, integrity
  - availability, recovery, manageability, security
  - replication, versioning, evolution, archiving, change notification
  - concurrency control, transactions, bulk load
- **It has taken huge amounts of effort to make these available in today's database servers**
- **Conclusion: we must build on this effort by integrating existing database servers into the Grid**
  - starting from scratch isn't an option.

Grid-characteristic requirements

- **Are there any requirements of grid-enabled databases that may require special attention?**
  - performance
    - scalability
    - unpredictability
  - meta-data-driven access
  - federation

Performance: scalability

- **Data size**
  - examples of large dbs
- **Query execution time**
  - example of large queries
- **Concurrent Users**
  - example of large number of concurrent users

Performance: Unpredictability

- **Need to support curiosity-driven access**
  - c.f. pre-determined access in e-commerce
- **Must manage unpredictable usage**
  - avoid denial of service
  - share resources fairly
- **Need to share db resources in a controlled way**
  - cost estimation helpful
  - resource usage quotas
  - holistic approach required
    - CPU, Memory, Disk, Network, DB Cache, Locks

Metadata based access

- **Users & applications may potentially have access to large numbers of data repositories**
- **Repositories can be selected via registries**
- **Two stage access:**
  1. use metadata access to registries to find repository(s)
     - importance of metadata standards
  2. access (and federate data) from repository(s)
     - importance of repository access standards
     - importance of federation…

Federation

- **Workflows allow us to manually combine data and computational services**
- **But, they need to be manually constructed**
  - time consuming, error prone, inflexible
- **Ideally there would be tools to federate data across the Grid**
  - utilise dynamically acquired Grid resources for federation
  - distributed query processing

Grid Services

- **Grid applications are now being built within a service-based framework**
- **In this section we will consider**
  - Brief history of Grid middleware
  - Open Grid Service Architecture
    - Web Services
    - Grid Services

Brief history of Grid middleware

- **Until 2002, Grid middleware suffered from:**
  - lack of standards
    - Globus was a de facto standard rather than de jure standard
    - Also other approaches such as Unicore
  - monolithic, so difficult to build and integrate components
  - lack of synergy with commercial middleware standards and tools
- **Since 2001 there has been an attempt to address this:**
  - Open Grid Services Architecture (OGSA)
- **Key Text:**
  - “The Physiology of the Grid, An Open Services Architecture for Distributed Systems Integration”, I. Foster, C. Kesselman, J.M. Nick & S. Tuecke. June 2002 (available from www.globus.org)

Why Services?

- **Virtualisation of resources**
  - computers, repositories, networks, programs, databases, activities… can all be represented as services
- **Advantages**
  - standard interface definition
  - standard way to invoke service
  - local/remote invocation transparency
  - ability to hide diverse implementations & platforms behind the interface
  - composition

Web Services

- **Grid Services are based on Web Services (WS)**
- **WS allow the creation of components that offer a set of operations**
  - based on message exchange & XML
- **Key WS standards define:**
  - interfaces (WSDL)
  - message formats (SOAP)

Transient Service Instances

- **Web Services address the discovery and invocation of persistent services**
- **In Grids, many resources and activities will be created dynamically and may be short-lived, e.g.**
  - a job running on a computer
  - a database session
  - data produced as the result of a query
- **A key feature of OGSA is the introduction of transient service instances**
  - created dynamically to encapsulate a transient resource or activity
  - created by factories
  - “Grid Service Instances”

Stateful Consumer-Service Interactions

Stateful Consumer-Service interactions can be implemented with Grid Service Instances

1. Client asks a Factory to create a new Grid Service Instance

2. Factory creates Grid Service Instance and returns handle to client

3. Client can then call operations on the Grid Service Instance

Other Grid Service Instances

- **The last part of the tutorial will include examples of Grid Service Instances being created to represent**
  - database sessions
  - data
  - computation on data

Grid Service Instance Lifetimes

- **Lifetime management provided**
  - service, host and client (policy permitting) can kill GSI
- **What if the client is in charge of lifetime but it fails, or the network fails, and kill request is never received?**
- **Soft-state management also provided**
  - initial lifetime set when GSI created (can be at client request)
  - client can extend this by request
  - when lifetime reached, GSI or host can kill GSI

Accessing a Grid Service Instance

- **Each GSI is assigned a globally unique name**
  - the Grid Service Handle (GSH)
    - protocol and network address independent
- **To access a service, this must be mapped onto a Grid Service Reference (GSR)**
  - protocol & network specific
  - therefore a factory returns a GSH and a GSR
- **A GSR can have a finite lifetime or become invalid**
  - re-resolution provided by a Handle-to-Reference mapper
- **This 2-level naming opens the way for:**
  - service upgrading
  - failover
  - scalability

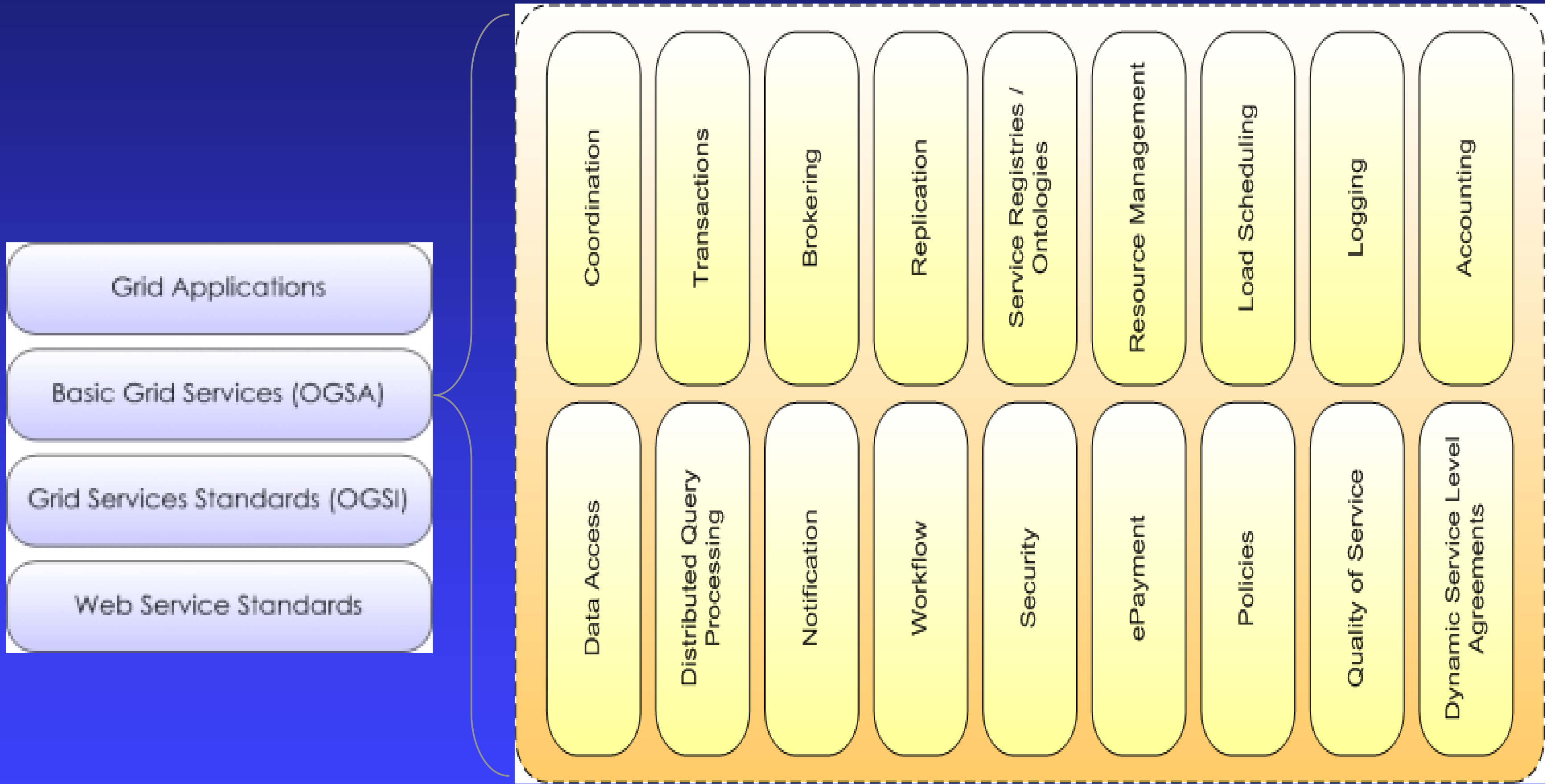
Exposing Service State

- **Service Data Elements (SDEs)**
- **Standard way for services to advertise and expose state**
- **Consumers can:**
  - find what state is exposed
  - query state
  - register to be notified of state changes

Open Grid Services Infrastructure (OGSI)

- **The basic interfaces and behaviours required by Grid Services are defined in the Open Grid Services Infrastructure (OGSI) specification**
  - v1 submitted to the Global Grid Forum as a recommendation track document (April 2003)
  - www.ggf.org/ogsi-wg
- **The first major implementation was made available in June 2003 as Globus Toolkit 3 (GT3)**
  - www.globus.org

The OGSA Architecture

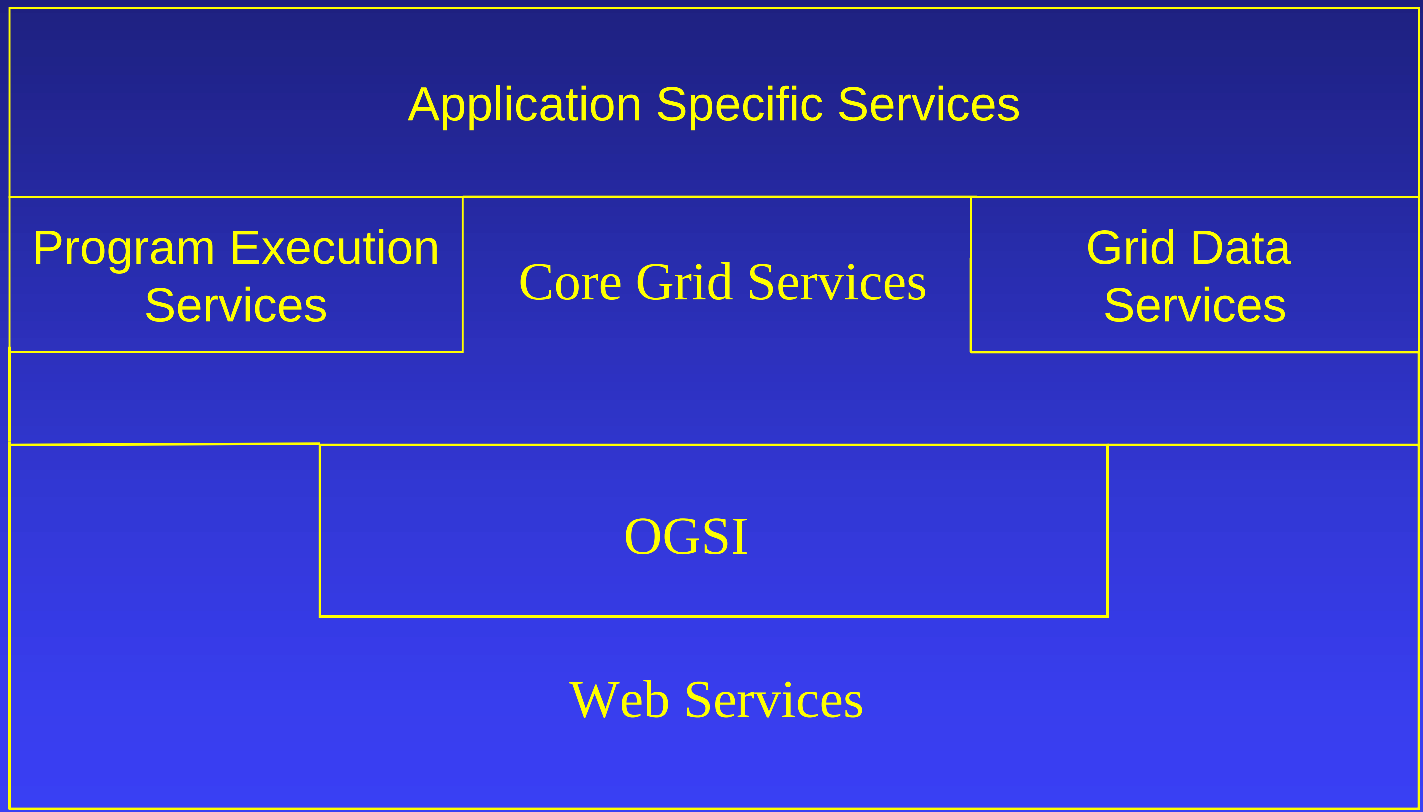


Part II b:
The Design of Grid Data Services

Grid Service Design/Development Status

- **Many projects have developed data grid components and infrastructures.**
- **Relatively few projects have yet deployed OGSI in anger.**
  - Individual projects are developing their own generic or application specific services.
  - Certain of these activities are feeding into standards within the Global Grid Forum (GGF).
  - The GGF OGSA Working Group and Data Area are seeking to coordinate activity.

Abstract Service Classification



Global Grid Forum OGSA Working Group: www.gridforum.org

Designing Grid Services

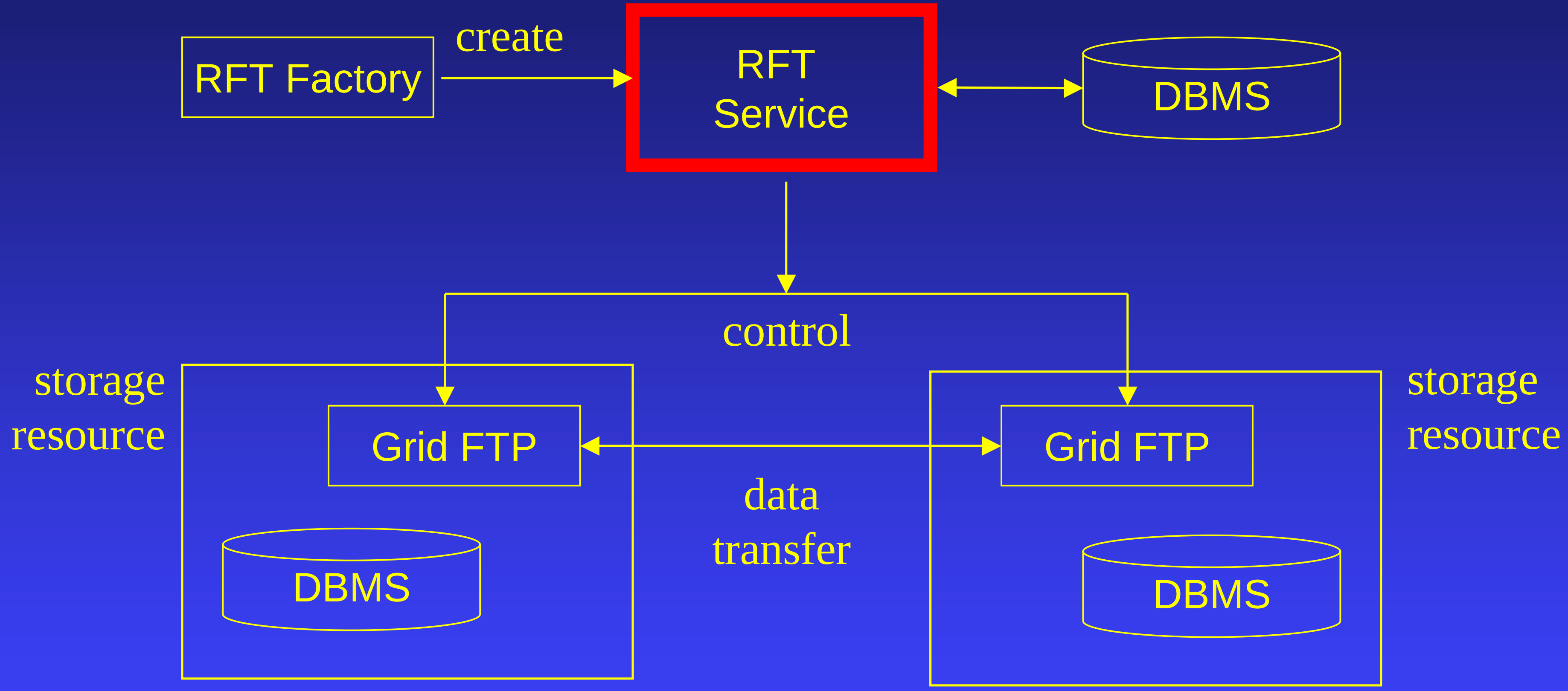
- **Service design:**
  - State – Service Data Elements (SDEs).
  - Operations – WSDL document.
- **Ancillary issues:**
  - Service lifetime.
  - Service granularity.
  - Operation granularity.
  - Extensibility.
- **Core services for data management under development:**
  - Data movement (RFT).
  - Data replication (GMR).
  - Database Access (OGSA-DAI).
  - Distributed Querying (OGSA-DQP).
- **There is no definitive "wedding cake" for Data Grid Services.**

Reliable File Transfer

- **RFT Features:**
  - Built over GridFTP.
  - Persistent transfer state.
  - Error recovery.
- **Service properties:**
  - Service lifetime bounds transfer duration.
  - Service handles a single job at a time.
  - A job may transfer many files.
- **Operations:**
  - start *JobDescription -> JobId.*
  - cancel *JobId*.
- **SDEs:**
  - FileTransfer Status.
  - FileTransferProgress.
  - FileTransferRestartMarker.
  - Version.

R.K. Madduri, W.E. Allcock, Reliable File Transfer in Grid Environments, Proc. 27th IEEE Conference on Local Computer Networks (LCN), 737-738, 2002.

Reliable File Transfer



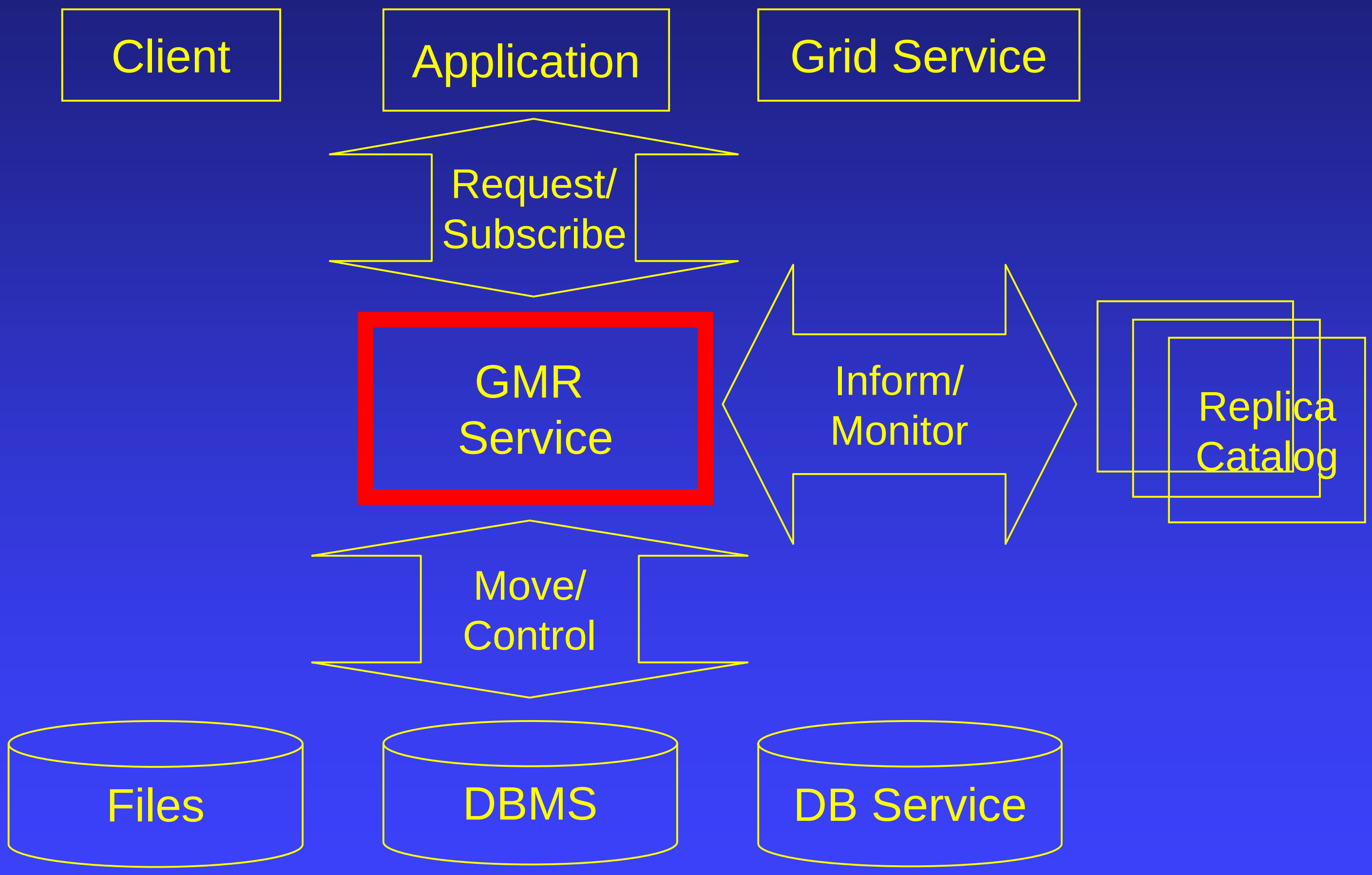
M. Atkinson, A.L. Chervenak, P. Kunszt, I. Narang, N.W. Paton, D. Pearson, A. Shoshani, P. Watson, Data Access, Integration and Management, The Grid (2nd Edition), I. Fister, C. Kesselman (eds), 2003.

Grid Movement and Replication

- **GMR Features:**
  - Pluggable architecture: scheduler, transport, adaptors.
- **Service properties:**
  - Service lifetime bounds replication refresh.
  - Service handles many jobs at a time.
  - A job may replicate many items.
- **Operations:**
  - createReplicationJob, updateReplicationJob, ...
- **SDEs:**
  - Job progress.
  - Job statistics.

A. Chokshi, S. Glanville, V Gogate, S. Jeffrey, C. Madsen, I. Narang, V. Raman, M Subramanian, OGSA Grid Data Movement and Replication Service, IBM Almaden, 2003.

GMR Architecture



Very
Large
Data
Bases

Example GMR Request

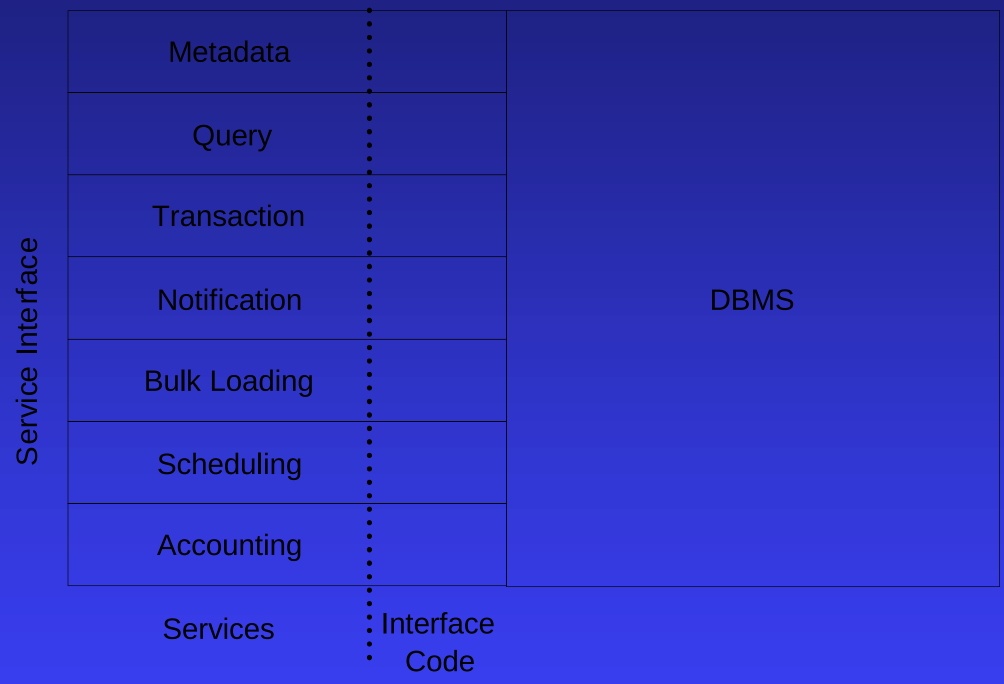
- **Request specifies:**
  - Source.
  - Target.
  - Requester.
  - Lifetime of replication.
  - Options on replication:
    - Replace.
    - Update.
    - NewCopy.
  - Scheduling information:
    - Scheduler name.
    - Arguments.

```
<GMRReplicationJobRequest>
  <Source>
    <DataGDR>
      <simple_path_GDR>
        <host>myHost1</host>
        <path>/foo/file1</path>
      </simple_path_GDR>
    </DataGDR>
  </Source>
  <Target>
    ...
  </Target>
  <RequestorInformation>
    <UserName>Bob</UserName>
  </RequestorInformation>
</GMRReplicationJobRequest>
```

Very Large Data Bases

Grid Services for Structured Data

- **General approach:**
  - wrap database as a service.
- **Aim to standardise where possible**
  - result format.
  - metadata:
    - operations supported.
    - driver employed.
    - schema.



P. Watson, Databases and the Grid, in Grid Computing: Making the Global Infrastructure a Reality, F. Berman, G. Fox, T. Hey (eds), Wiley, 2003.

Mapping Sessions to Services

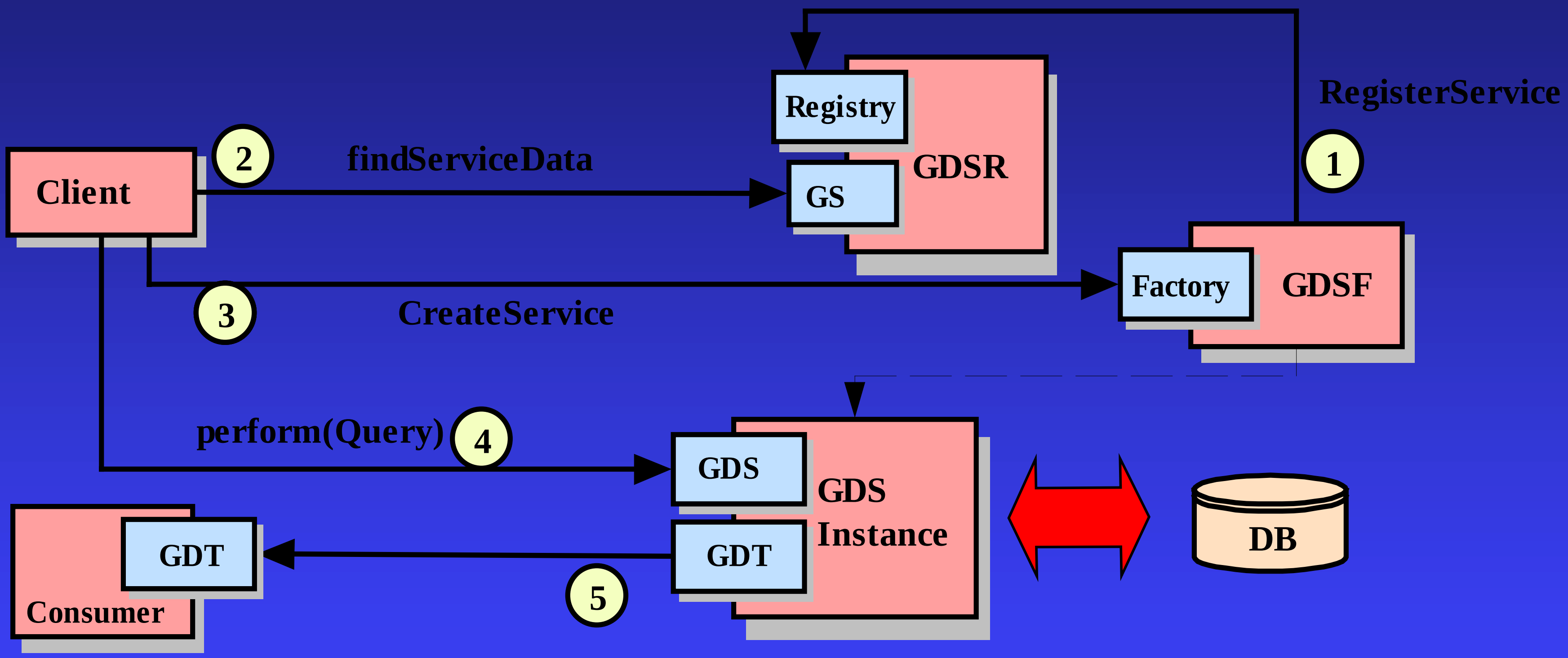
- **An important architectural issue is how to map client database sessions onto services.**
- **Options include:**
  - one service per database:
    - all clients connect to the same service.
    - natural for Web Services.
    - needs way to internally handle the state of multiple sessions.
  - one service instance per session:
    - a new service instance is created for each new session.
    - natural for Grid Services.

OGSA Data Access & Integration

- **OGSA-DAI Features:**
  - Wraps relational and XML databases.
  - Compound requests.
  - Flexible delivery.
- **Service properties:**
  - Represents a user session.
  - User can have many concurrent sessions.
  - Service actively processes one request at a time.
- **Operations:**
  - perform *Request -> Result.*
- **SDEs:**
  - Schema.
  - DBMS.
  - Driver.
  - Stored requests.
  - Request progress.
  - ...

OGSA-DAI Software: http://www.ogsa-dai.org.uk

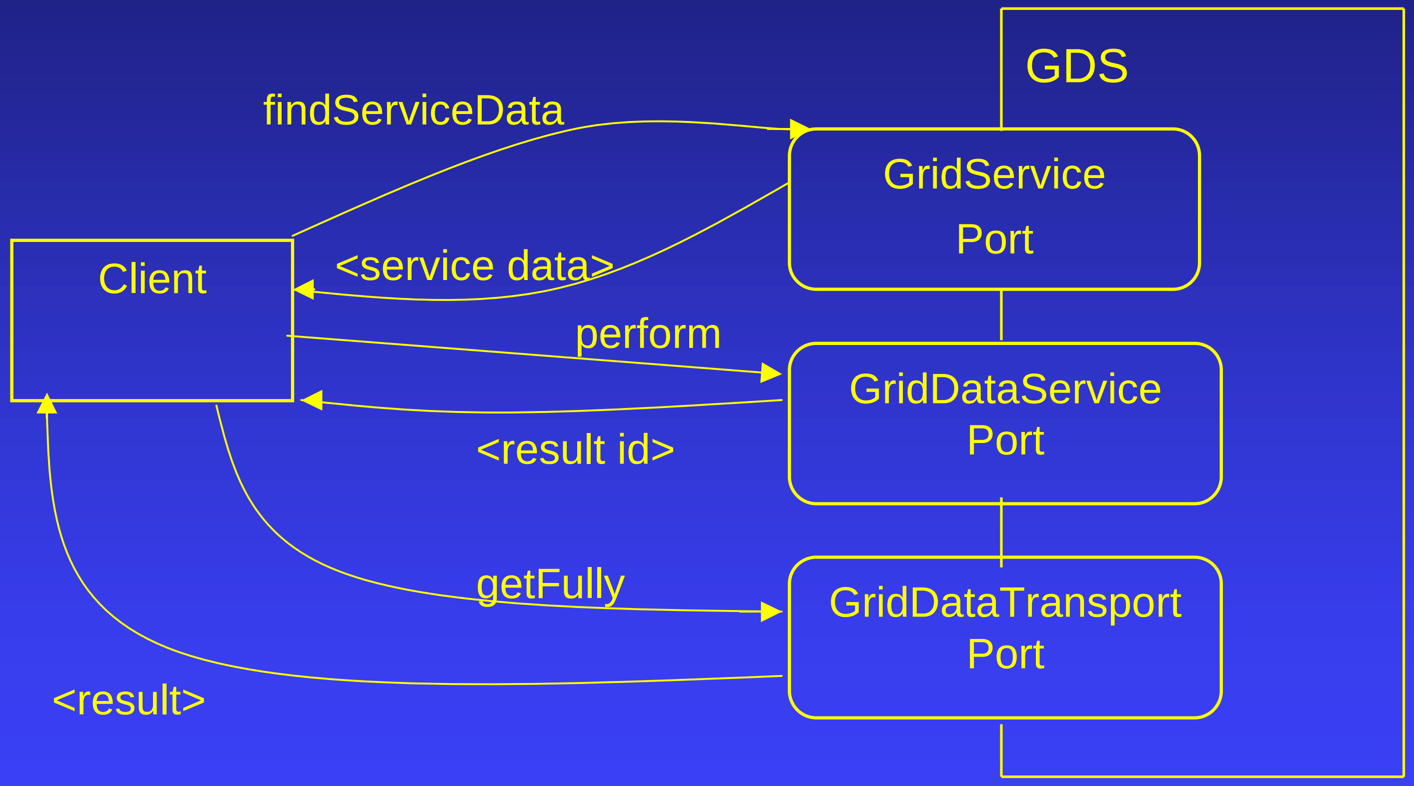
Service Interactions



Terminology

- **GDSR:**
  - Grid Database Service Registry.
  - Searchable registry of GDSFs.
- **GDSF:**
  - Grid Database Service Factory.
  - Supports the creation of GDS instances.
- **GS:**
  - GridService portType.
- **GDS:**
  - Grid Database Service.
  - Supports a client session with the database.
- **GDT:**
  - Grid Database Transport portType.
  - Endpoint for service-to-service data pull/push requests.

GDS PortTypes



Service Data Elements

- **Sample SDEs:**
  - dataResource.
  - driverType .
  - dataResourceManagementSystem.
  - database.
  - databaseSchema.
  - runningRequests.
  - definedRequests.
  - activitiesSupported.

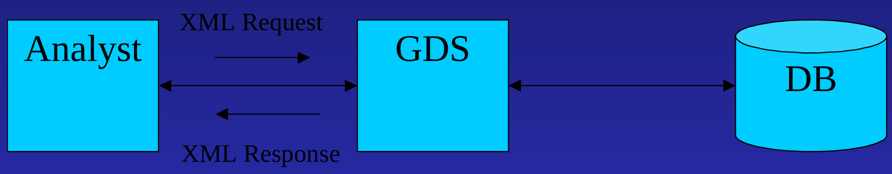
Example SDE

- **Suppose database has a table:**
  - Person (id int(11) primary key, …)
- **Selection of Service Data Element:**
  - <queryByServiceDataNames name="databaseSchema"/>
- **Sample of SDE document:**
  - <databaseSchema dataResource="mydatabase">
    <table name="Person">
    <column name="id" fullname="Person.id" length="11">
    <sqlType>INTEGER</sqlType>
    </column>
    …
    </table>
    </databaseSchema>

Perform Documents

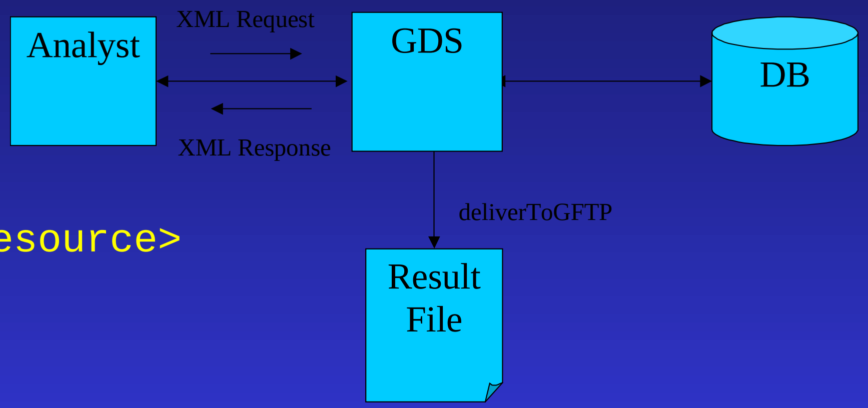
- **A *GDS::perform* operation takes a potentially complex document as input.**
- **The top-level elements indicate the action that should be taken:**
  - *Request* defines a request for storage by the GDS.
  - *Execute* runs a stored request.
  - *Terminate* stops a running request.
- **A *Request* element is a collection of linked activities, such as:**
  - *sqlQueryStatement* specifies an SQL select statement.
  - *sqlUpdateStatement* specifies an SQL DML statement.
  - Data delivery descriptions that specify movement of data between the service and some third party.

Query With Direct Response




```
<gridDataServicePerform … >
  <request name="myRequest">

    <sqlQueryStatement
        name="statement">
      <dataResource>frogGenome</dataResource>
      <expression>
        select * from chromFrag where len &lt 20000
      </expression>
      <webRowSetStream  name="statementresponse"/>
    </sqlQueryStatement>

    <deliverToResponse name="d1">
         <fromLocal from="statementresponse"/>
    </deliverToResponse>

  </request>
</gridDataServicePerform>
```

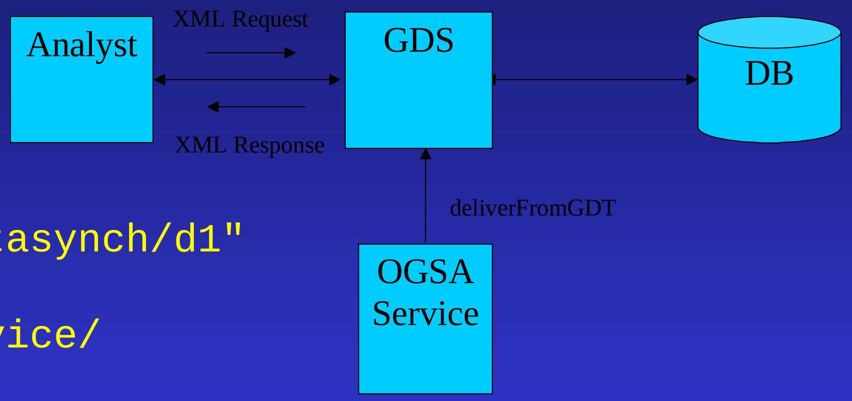
Query With Third Party Delivery




```
<gridDataServicePerform … >
  <request name=“myPushRequest">

    <sqlQueryStatement name=“stmt">
      <dataResource>frogGenome</dataResource>
      <expression>
        select * from chromFrag
        where len &lt 20000
      </expression>
      <webRowSetStream name="statementresponse"/>
    </sqlQueryStatement>

    <deliverToGFTP name="d1">
         <fromLocal from="statementresponse"/>
         <toGFTP host="ogsdai.org.uk"
                 port="8080" file="path/to/myfile.txt"/>
    </deliverToGFTP>
  </request>
</gridDataServicePerform>
```

Update Pulling From Third Party

```
<gridDataServicePerform>
  <request name="myPullRequest">

    <deliverFromGDT name="d1">
      <fromGDT streamId="otherrequestasynch/d1"
               mode="full">
        http://ogsadai.org.uk/GDTService/
            my/GDT/GSH
      </fromGDT>
      <toLocal name="datatoinsert"/>
    </deliverFromGDT>

    <sqlUpdateStatement name="statement">
      <sqlParameter position="1" from="datatoinsert"/>
      <expression>insert into chromFrag values ?</expression>
    </sqlUpdateStatement>
  </request>
</gridDataServicePerform>
```



Design Issues for GDS

- **Service granularity:**
  - Notions to capture:
    - Installed DBMS.
    - Database within DBMS.
    - Session over database.
    - Request.
    - Query result set.
  - Which should have:
    - PortType definitions.
    - Service Data Elements.
  - Which should be:
    - Service instances.
- **Operation granularity.**
  - Atomic operations:
    - Execute SQL Query.
    - Apply XSLT transformation.
    - Map directly to multi-operation PortType.
    - Good for tooling.
  - Compound operations:
    - Execute query, apply transformation, deliver result.
    - Need to define orchestration notation.
    - Fewer fine-grained calls.

Standardisation

- **Global Grid Forum:**
  - Standards body for Grid Computing.
  - Open meetings three times a year.
  - www.gridforum.org.
- **Database Access and Integration Services Working Group (DAIS-WG):**
  - Sessions at each GGF meeting.
  - Intermediate telcons and face-to-face sessions.
  - Developing standard proposal.
  - http://www.gridforum.org/6_DATA/dais.htm
- **OGSA-DAI 2.5 broadly implements the February 03 DAIS draft.**

Databases and the Grid

- **Deploying databases:**
  - Q: How are databases made available on the Grid?
  - A: Through integration with other Grid services, and provision of standard interfaces.
- **DAIS GGF Working Group.**

- **Deploying database technologies:**
  - Q: What database technologies might be broadly useful as Grid services?
  - A: transactions, materialised views, distributed query processing, …

Distributed Query Processing

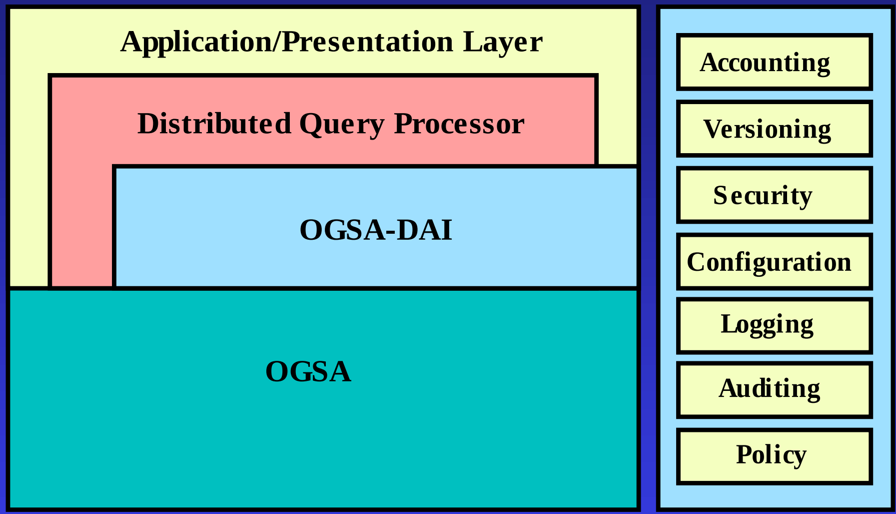
- DQP involves a single query referencing data stored at multiple sites.
- The locations of the data may be transparent to the author of the query.

```
select p.proteinId, Blast(p.sequence)
from   protein p, proteinTerm t
where  t.termId = 'GO:0005942' and
       p.proteinId = t.proteinId
```

J. Smith, A. Gounaris, P. Watson, N. Paton, A. Fernandes, R. Sakellariou, Distributed Query Processing on the Grid, 3rd Int. Workshop on Grid Computing, Springer-Verlag, 279-290, 2002.

Service-Based DQP

- **Service-based in two respects:**
  - Queries are expressed over services.
    - Grid database services.
    - Computational services.
  - The distributed query processor it itself implemented as a collection of services.



OGSA-DQP Software: http://www.ogsa-dai.org.uk

Service-based DQP

- **DQP often exploits source wrappers.**
- **In service-based DQP, the sources are Web or Grid Services.**
- **A query may refer to database (GDS) and computational services.**
- **Workflow languages are often used for service orchestration.**
  - Emerging standard: BPEL4WS.
  - Procedural request description.
  - Programmer controls order of evaluation.

Mutual Benefit

- **The Grid needs DQP:**
  - Declarative, high-level resource integration with implicit parallelism.
  - DQP-based solutions should in principle run raster than those manually coded.
- **DQP needs the Grid:**
  - Systematic access to remote data and computational resources.
  - Dynamic resource discovery and allocation.

Phases in Query Processing

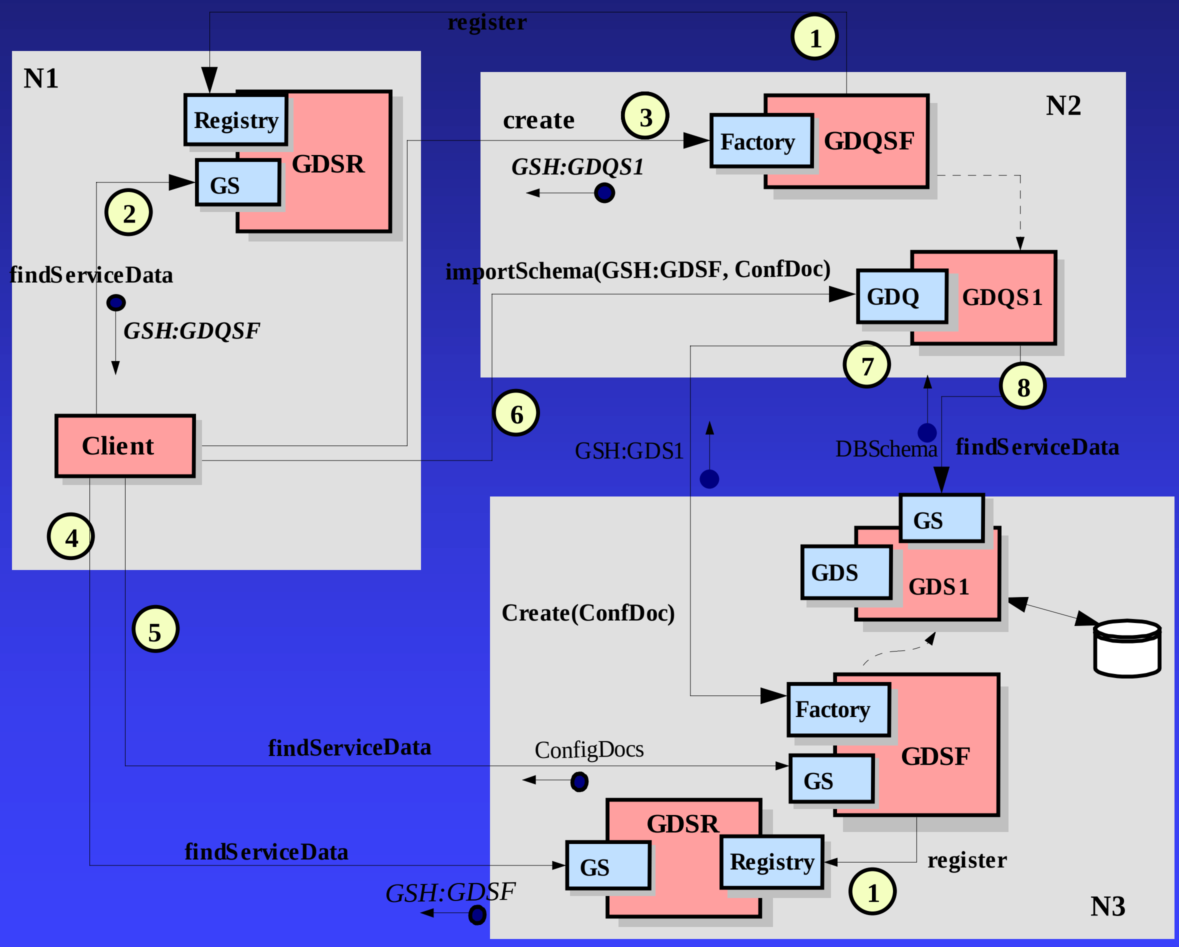
- **Initialisation:**
  - Grid Distributed Query Service is created.
  - Source descriptions imported.
- **Query compilation:**
  - Query optimisation and scheduling.
- **Query evaluation:**
  - Query evaluation services created as needed.
  - Query partitions allocated to evaluators for execution.

Distributed Query Services

- **Services:**
  - GridDistributedQueryService (GDQS).
  - GridQueryEvaluationService (GQES).
  - Associated factories.
- **Properties:**
  - Both *GDQS* and *GQES* services implement the portTypes of the Grid Database Service.
  - The *GDQS* is extended to support the importing of service descriptions.

Very Large Data Bases

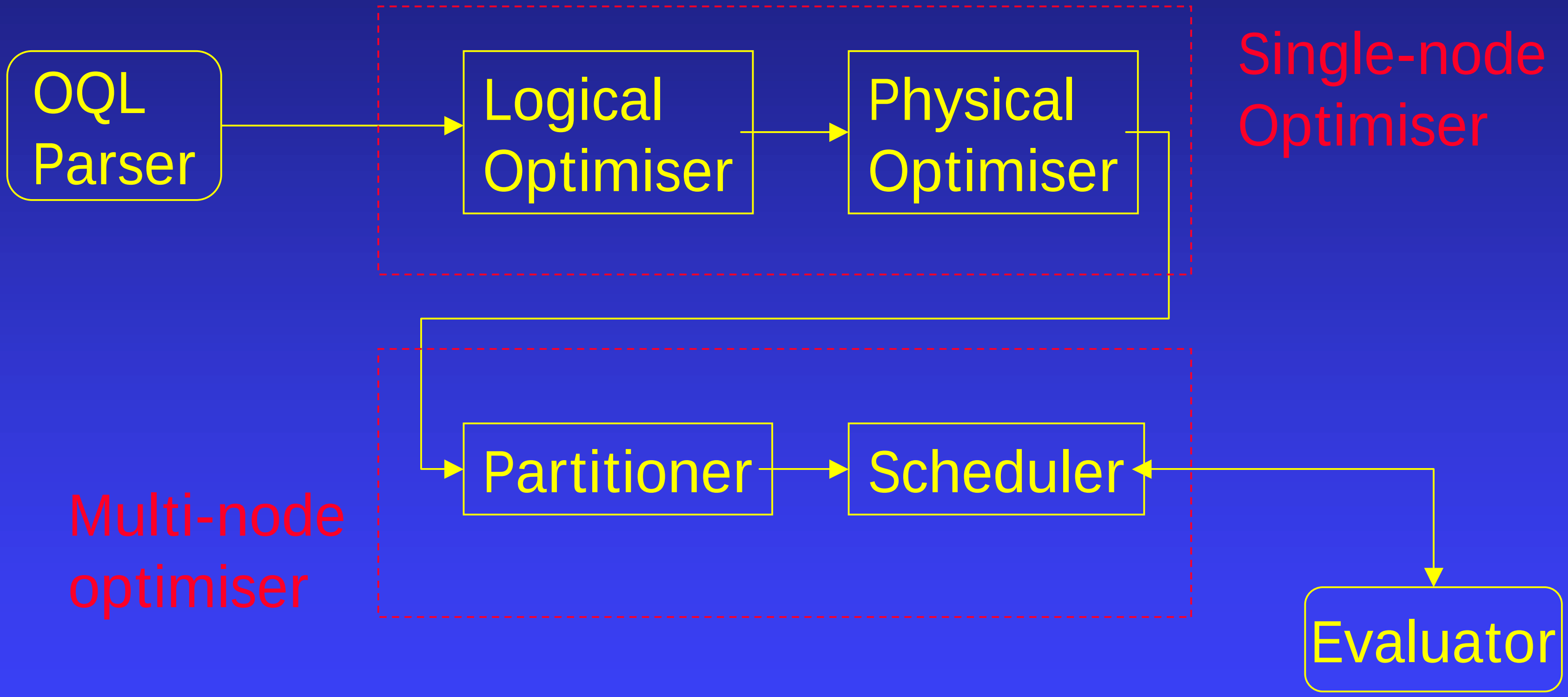
Initialising a GDQS



Issues in Initialisation

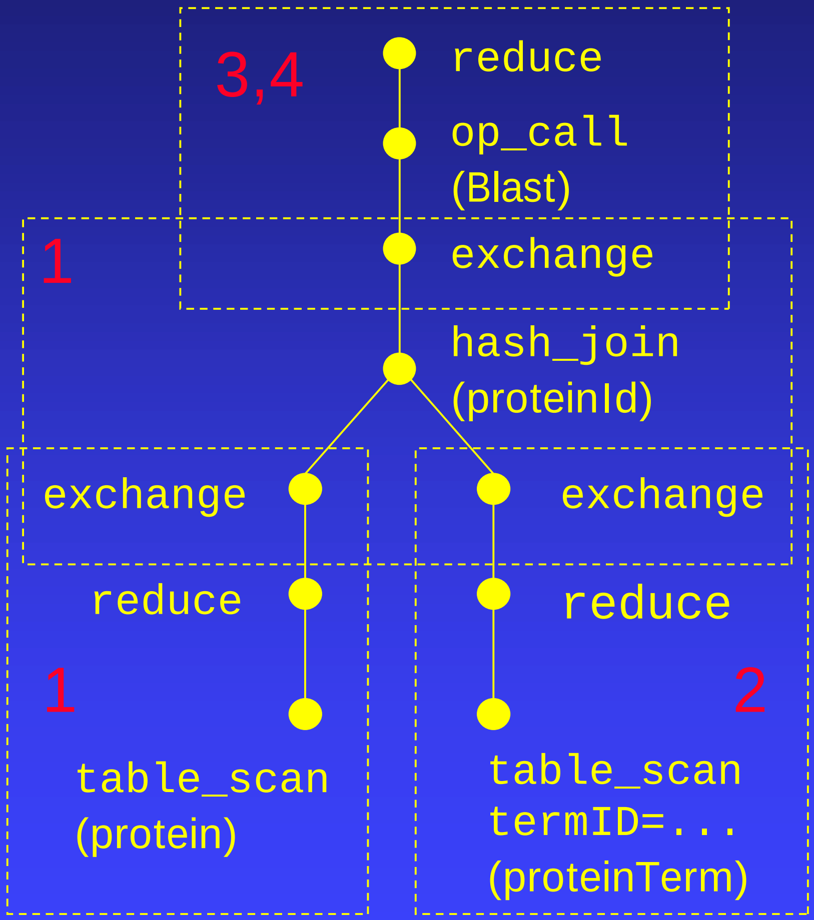
- **Q: When is a GDQS bound to a particular GDS?**
  - A: When the schema of the GDS is imported.
- **Q: What is the lifespan of a GDS used by a GDQS?**
  - A: The GDS is kept alive until the GDQS expires.
- **Q: Are GDSs shared by multiple GDQSs?**
  - A: No.
- **Q: When is a GQES created?**
  - A: When a query is about to be evaluated that needs it.
- **Q: What is the lifespan of a GQES?**
  - A: It lasts only as long as a single query.
- **Q: Is a GQES shared among several queries or GDQSs?**
  - A: No.

Query Compilation



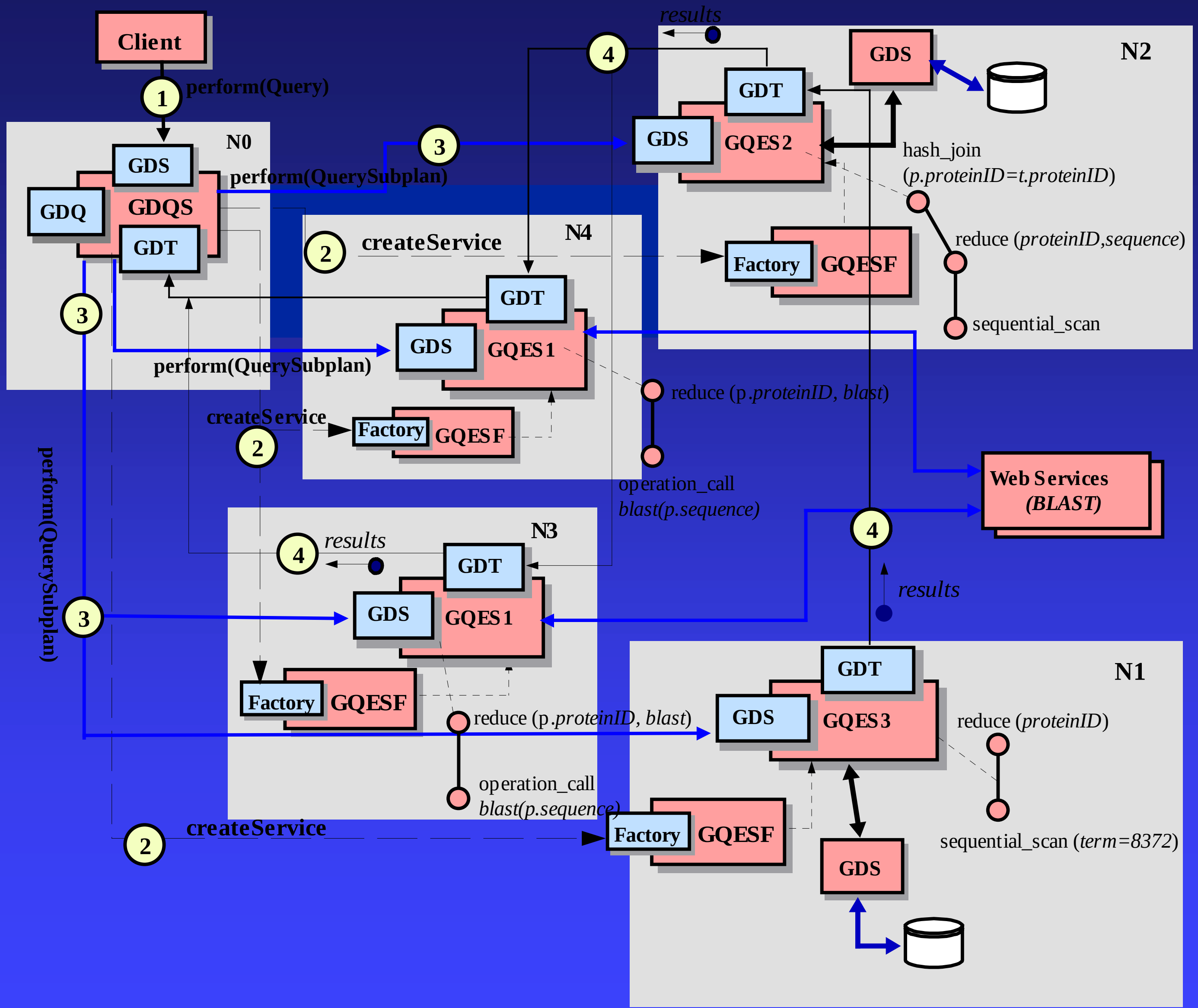
Scheduling

- Partitions are allocated to Grid nodes; partitions may be merged during scheduling.
- Expressed by parallel algebra expression.
- Heuristic algorithm considers memory use, network costs.



Query Evaluation

- **Query installation:**
  - GQESs created for partitions as required.
  - Partitions sent to GQESs.
- **Query evaluation:**
  - Partitions evaluated using iterator model.
  - Pipelined and partitioned parallelism.
  - Results conveyed to client.


Client
perform(Query)
1
N0
GDS
GDQ
GDQS
GDT
perform(QuerySubplan)
3
results
4
N2
GDS
GDT
GDS
GQES2
hash_join
(p.proteinID=t.proteinID)
reduce (proteinID,sequence)
Factory
GQESF
sequential_scan
2
createService
N4
GDT
GDS
GQES1
reduce (p.proteinID, blast)
perform(QuerySubplan)
createService
Factory
GQESF
operation_call
blast(p.sequence)
Web Services
(BLAST)
perform(QuerySubplan)
N3
results
GDT
GDS
GQES1
Factory
GQESF
reduce (p.proteinID, blast)
operation_call
blast(p.sequence)
results
N1
GDT
GDS
GQES3
reduce (proteinID)
sequential_scan (term=8372)
Factory
GQESF
GDS
createService

Features of OGSA-DQP

- **Low cost of entry:**
  - Imports source descriptions through GDSs.
  - Imports service descriptions as WSDL.
- **Throw-away GDQS:**
  - Import sources on a task-specific basis.
  - Discard GDQS when task completed.
- **Builds on parallel database technology:**
  - Implicit parallelism.
  - Pipelined + partitioned parallel evaluation.
- **Integrates data access with operation invocation.**

Related Work on Database Services

- **Web services interfaces to databases (e.g.):**
  - K. Mensah, Web Services Enable Your Database, Web Services Journal, Vol 3, No 4, 2003.
  - S. Malaika et al., DB2 and Web Services, IBM Systems Journal, Vol 41, No 4, 666-685, 2002.
- **Distributed querying using services:**
  - T. Malik and A. Szalay, SkyQuery: A Web Service Approach to Federate Databases, Proc. CIDR, http://www-db.cs.wisc.edu/cidr/program/p17.pdf, 2003.
  - R. Braumandl, et al., ObjectGlobe: Ubiquitous query processing on the Internet, VLDB J., Vol 10, 48-71, 2001.

Grid Data Management Services

- **Data Grid applications can benefit from many lower level services:**
  - Data movement.
  - Replication.
  - Database access and integration.
- **Work is underway on designing, developing and standardising many core Grid Data Management services.**
- **Designing services in a dynamic and heterogeneous environment is non-trivial, and there is much still to be done.**

Outstanding Research Issues

- Adaptability.
- Cost modelling.
- Data encoding.
- Data placement.
- Caching and replication.
- Glide-in databases.
- Management of Grid Resources.
- Orchestration.
- Quality of service.
- Scheduling.
- Security.
- Service description.
- Service frameworks.