



Grid MasterClass

Carl Kesselman

Information Sciences Institute
University of Southern California
Univa Corporation

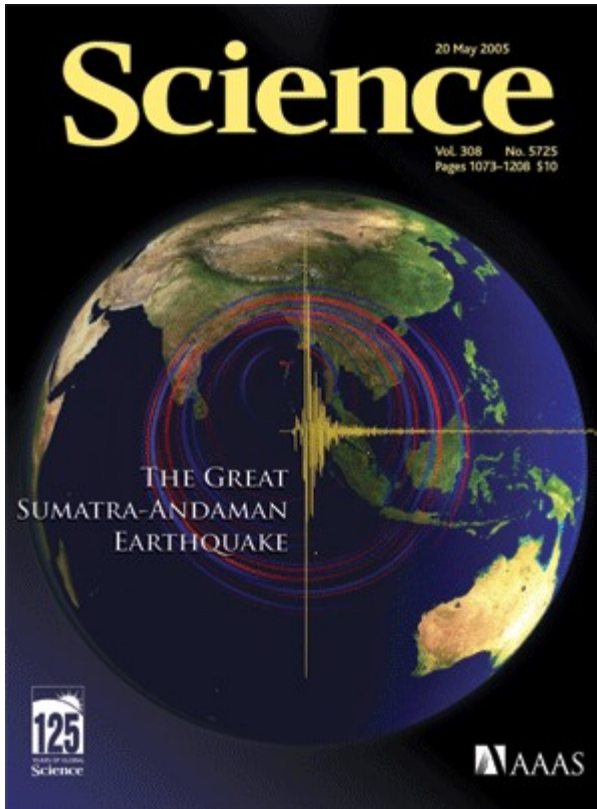
Credits

- Globus Toolkit v4 is the work of many talented Globus Alliance members, at
 - ◆ Argonne Natl. Lab & U.Chicago
 - ◆ USC Information Sciences Corporation
 - ◆ National Center for Supercomputing Applns
 - ◆ U. Edinburgh
 - ◆ Swedish PDC
 - ◆ Univa Corporation
 - ◆ Other contributors at other institutions
- Supported by DOE, NSF, UK EPSRC, and other sources

Acknowledgements

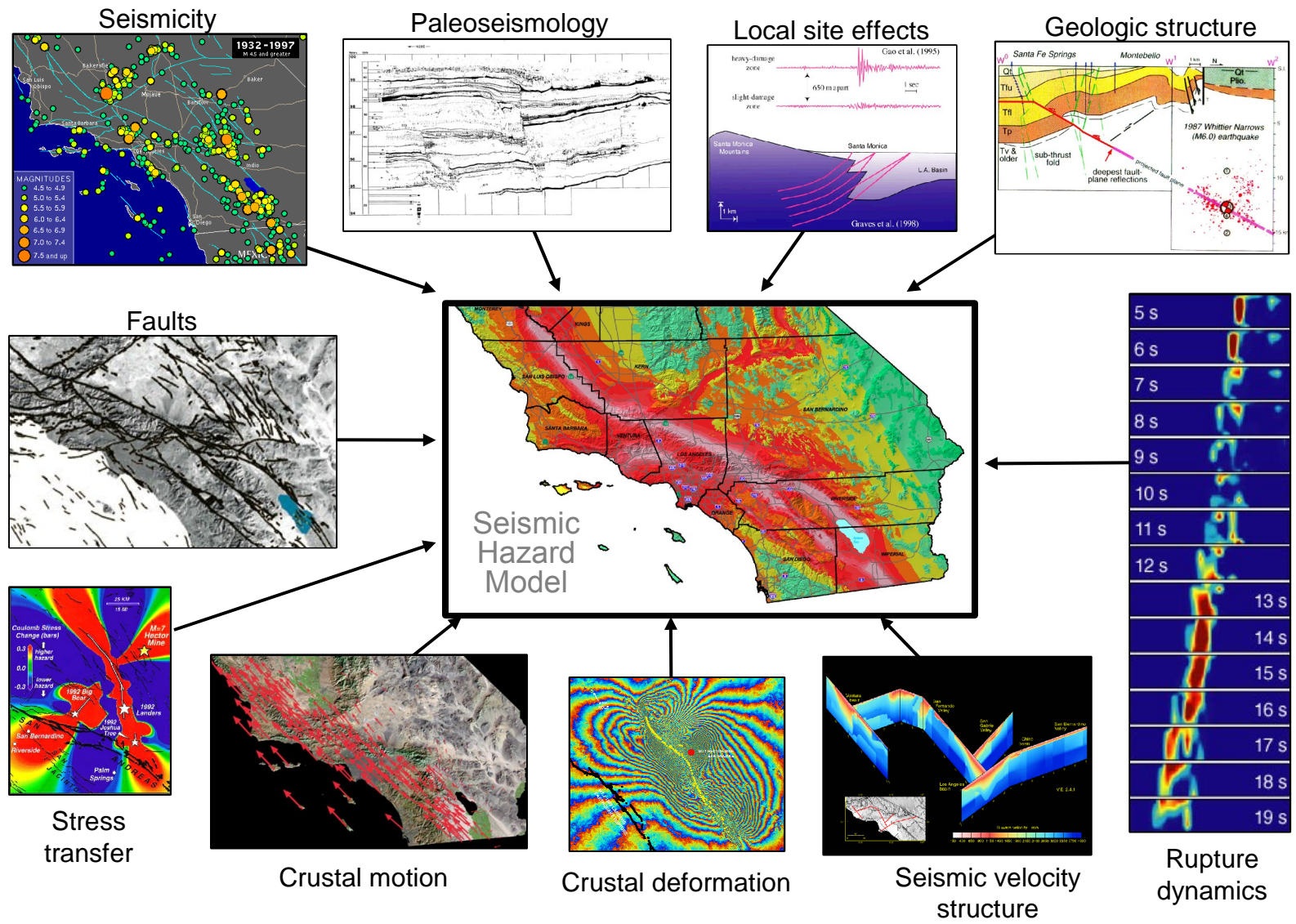
- Ian Foster with whom I developed many of these slides
- Bill Allcock, Lisa Childers, Kate Keahey, Jennifer Schopf, Frank Siebenlist, Mike Wilde @ ANL/UC
- Ann Chervenak, Ewa Deelman, Laura Pearlman, Mike D'Arcy, Rob Schuler @ USC/ISI
- Karl Czajkowski, Steve Tuecke @ Univa
- Numerous other fine colleagues
- NSF, DOE, IBM for research support

Context: System-Level Science



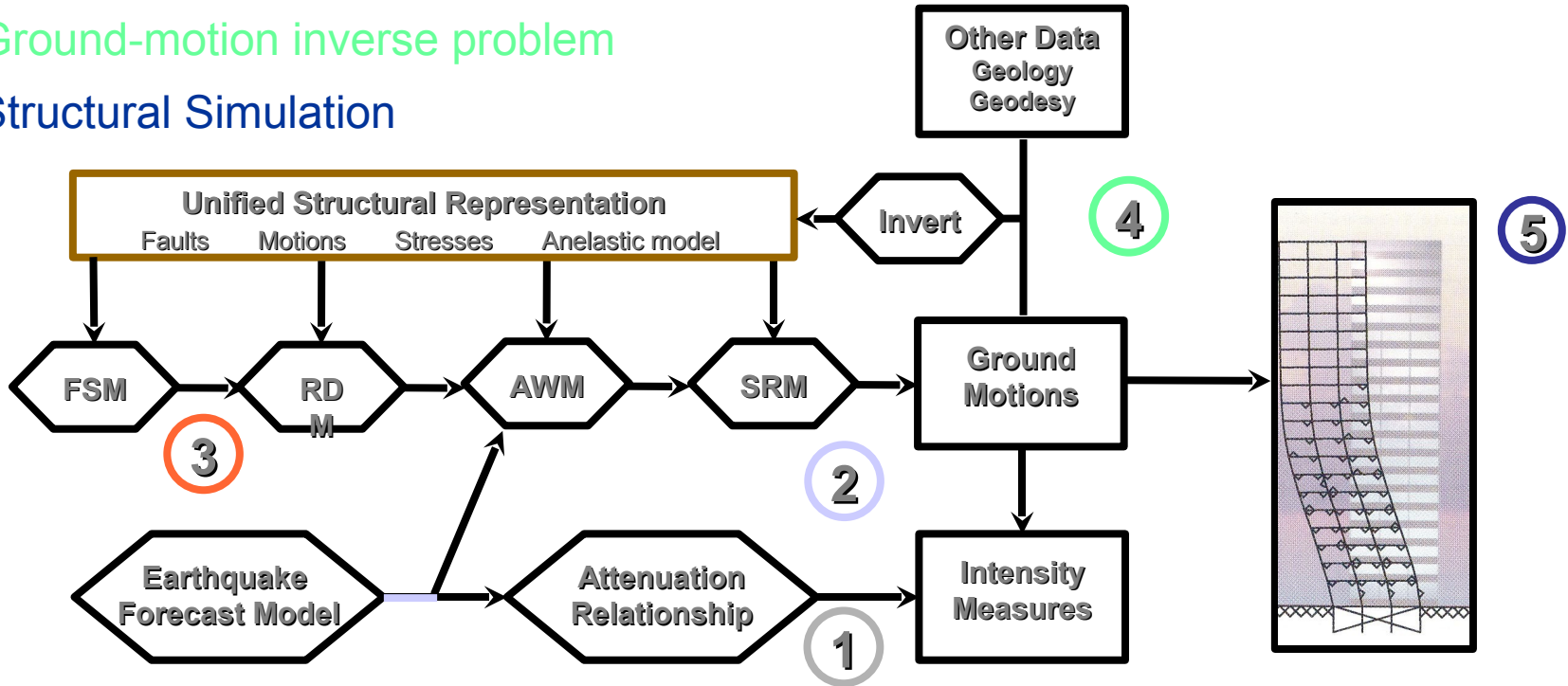
Problems too large &/or complex to tackle alone ...

Seismic Hazard Analysis (T. Jordan & SCEC)



SCEC Community Model

- ① Standardized Seismic Hazard Analysis
- ② Ground motion simulation
- ③ Physics-based earthquake forecasting
- ④ Ground-motion inverse problem
- ⑤ Structural Simulation



FSM = Fault System Model
RDM = Rupture Dynamics Model

AWP = Anelastic Wave Propagation
SRM = Site Response Model

Science Takes a Village ...

- Teams organized around common goals
 - ◆ People, resource, software, data, instruments...
- With diverse membership & capabilities
 - ◆ Expertise in multiple areas required
- And geographic and political distribution
 - ◆ No location/organization possesses all required skills and resources
- Must adapt as a function of the situation
 - ◆ Adjust membership, reallocate responsibilities, renegotiate resources

Virtual Organizations

- From organizational behavior/management:
 - ◆ "a group of people who interact through interdependent tasks guided by common purpose [that] works across space, time, and organizational boundaries with links strengthened by webs of communication technologies" (Lipnack & Stamps, 1997)
- **The impact of cyberinfrastructure**
 - ◆ **People → computational agents & services**
 - ◆ **Communication technologies → IT infrastructure, i.e. Grid**

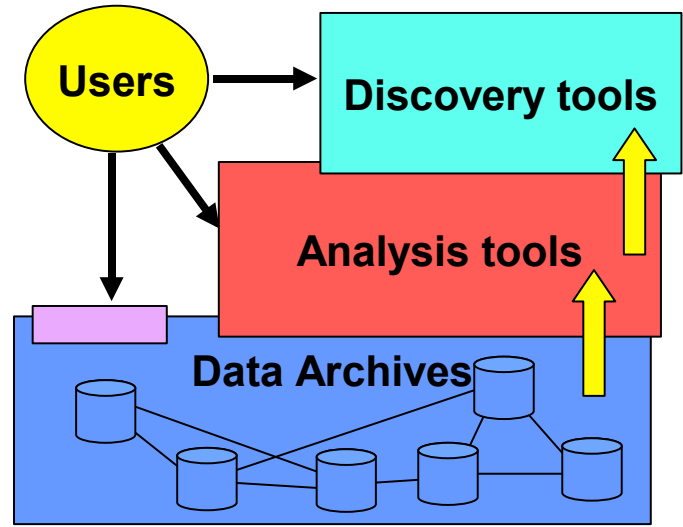
"The Anatomy of the Grid", Foster, Kesselman, Tuecke, 2001

Beyond Science Silos: Service-Oriented Architecture



Function
Resource

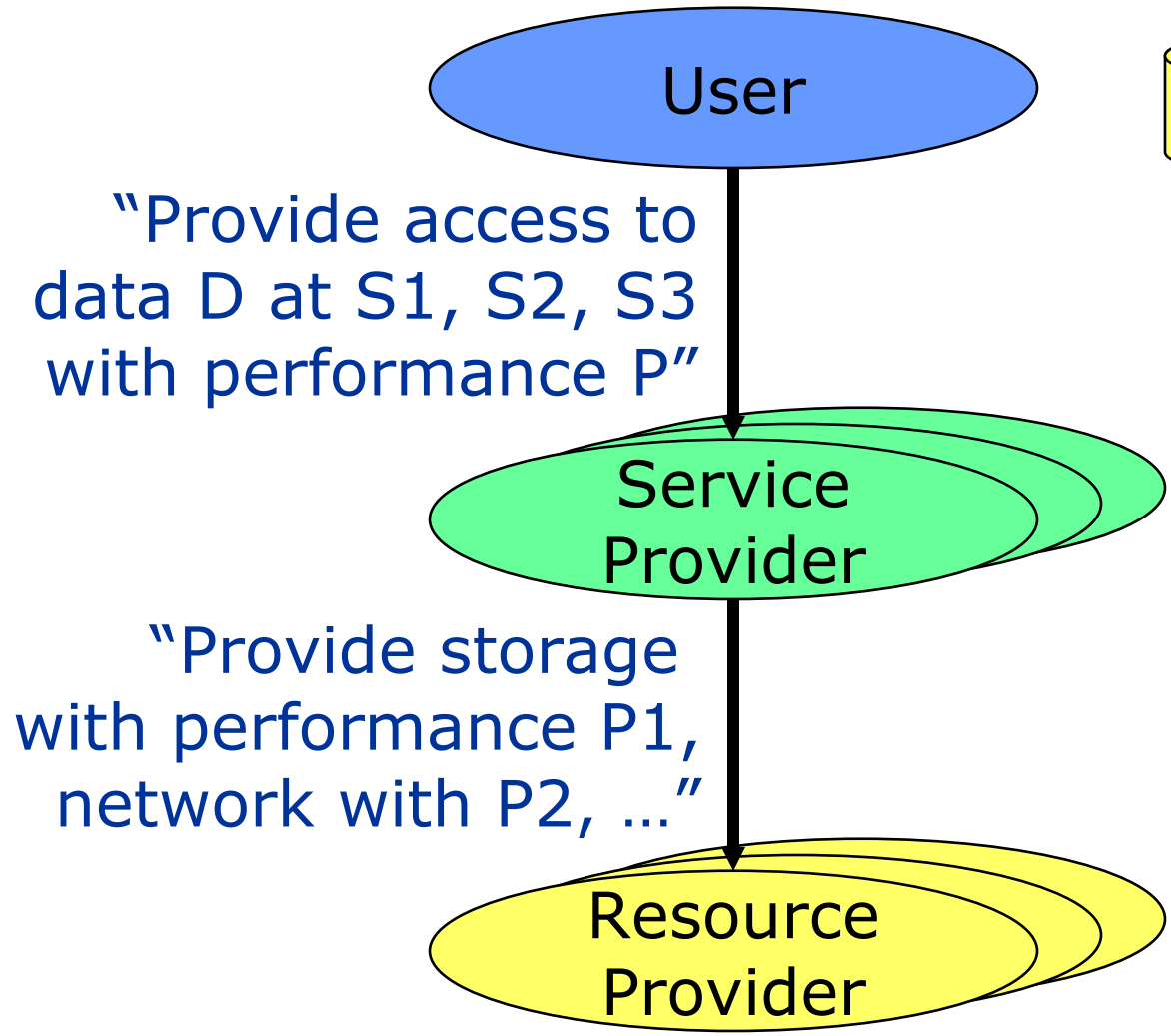
- **Decompose** across network
- Clients **integrate** dynamically
 - ◆ Select & compose services
 - ◆ Select "best of breed" providers
 - ◆ Publish result as a new service



- Decouple **resource** & **service** providers

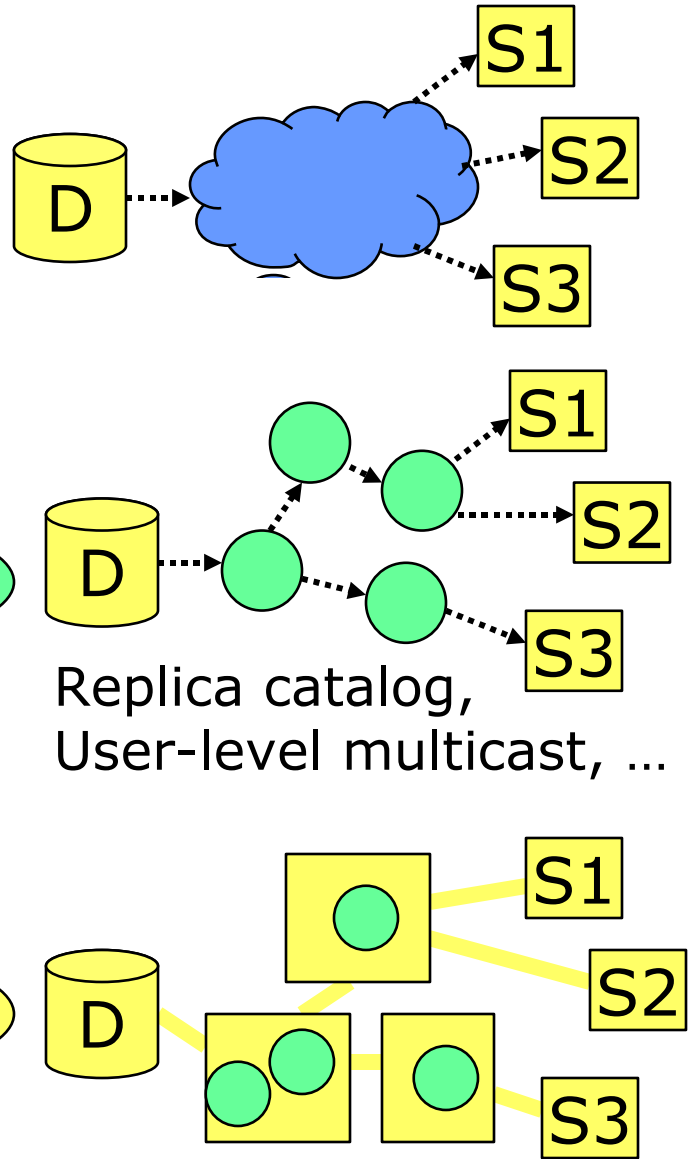
Fig: S. G. Djorgovski

Decomposition Enables Separation of Concerns & Roles



“Provide access to data D at S1, S2, S3 with performance P”

“Provide storage with performance P1, network with P2, ...”



Forming & Operating (Scientific) Communities

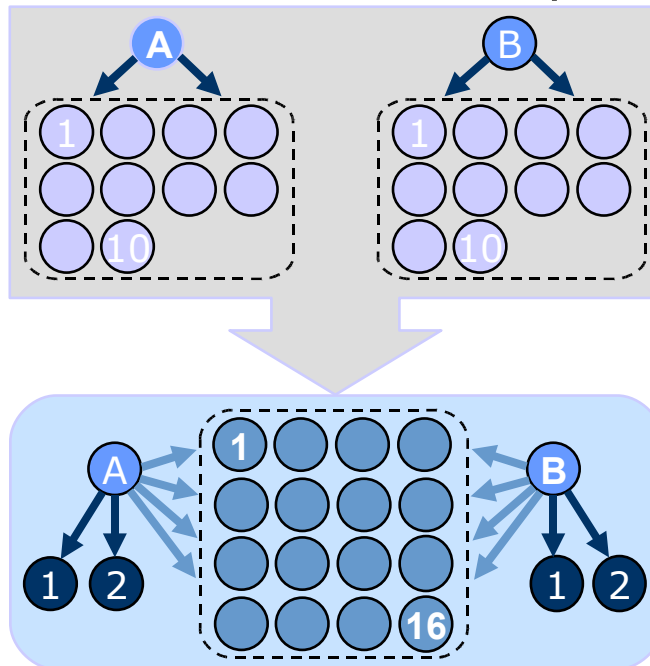
- Define VO membership and roles, & enforce laws and community standards
 - ◆ I.e., policy
- Build, buy, operate, & share community infrastructure
 - ◆ Data, programs, services, computing, storage, instruments
 - ◆ Service-oriented architecture
- Define and perform collaborative work
 - ◆ Use shared infrastructure, roles, & policy
 - ◆ Manage community workflow

Forming & Operating (Scientific) Communities

- **Define VO membership and roles, & enforce laws and community standards**
 - ◆ **I.e., policy**
- Build, buy, operate, & share community infrastructure
 - ◆ Data, programs, services, computing, storage, instruments
 - ◆ Service-oriented architecture
- Define and perform collaborative work
 - ◆ Use shared infrastructure, roles, & policy
 - ◆ Manage community workflow

Defining Community: Membership and Laws

- Identify VO participants and roles
 - ◆ For people and services
- Specify and control actions of members
 - ◆ Empower members → delegation
 - ◆ Enforce restrictions → federate policy

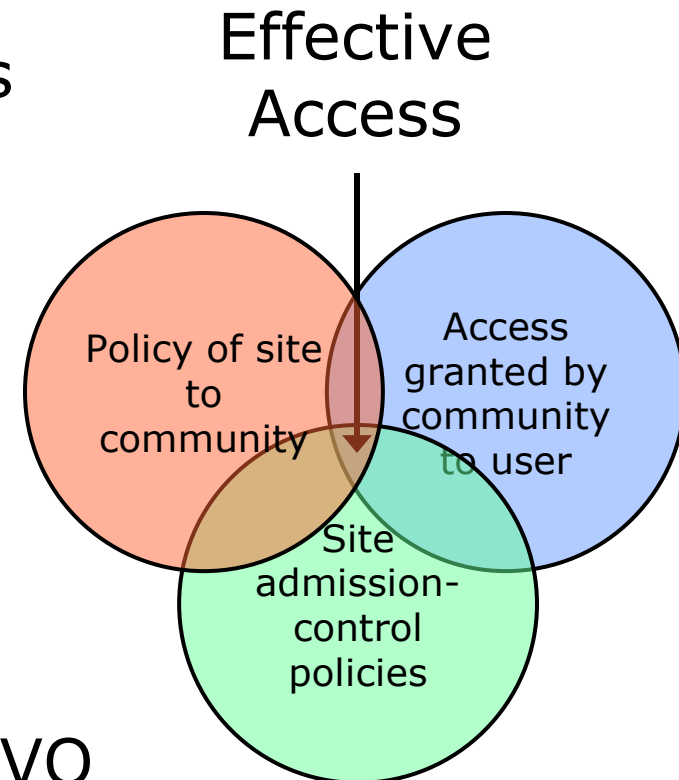


Security Services Objectives

- It's all about "policy"
 - ◆ Define a VO's operating rules
 - ◆ Security services facilitate the enforcement
- Policy facilitates "business objectives"
 - ◆ Related to goals/purpose of the VO
- Security policy often delicate balance
 - ◆ Legislation may mandate minimum security
 - ◆ More security ⇔ Higher costs
 - ◆ Less security ⇔ Higher exposure to loss
 - ◆ Risk versus Rewards

Policy Challenges in VOs

- Restrict VO operations based on characteristics of requestor
 - ◆ VO dynamics create challenges
- Intra-VO
 - ◆ VO specific roles
 - ◆ Mechanisms to specify/enforce policy at VO level
- Inter-VO
 - ◆ Entities/roles in one VO not necessarily defined in another VO



Core Security Mechanisms

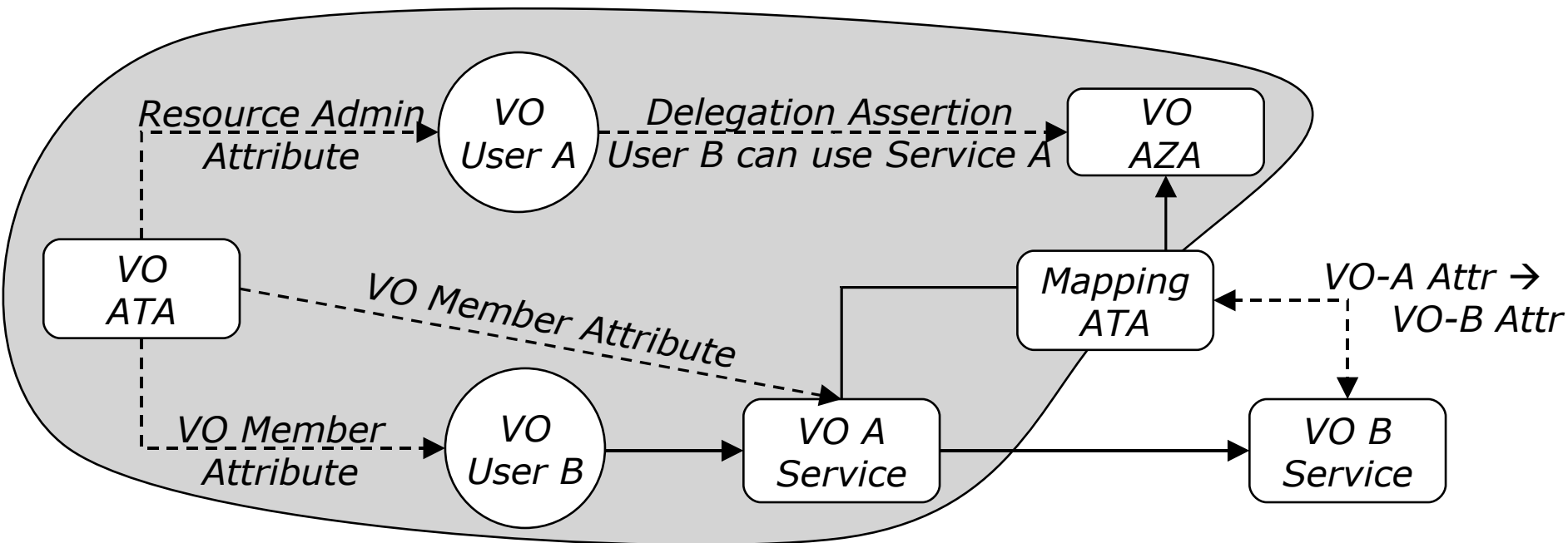
- Attribute Assertions
 - ◆ C asserts that S has attribute A with value V
- Authentication and digital signature
 - ◆ Allows signer to assert attributes
- Delegation
 - ◆ C asserts that S can perform O on behalf of C
- Attribute mapping
 - ◆ $\{A_1, A_2 \dots A_n\}_{vo1} \Rightarrow \{A'_1, A'_2 \dots A'_m\}_{vo2}$
- Policy
 - ◆ Entity with attributes A asserted by C may perform operation O on resource R

Trust in VOs

- Do I “believe” an attribute assertion?
 - ◆ Used to evaluate cost vs. benefit of performing an operation
 - E.g., perform untrusted operation with extra auditing
- Look at attributes of assertion signer
- Rooting trust
 - ◆ Externally recognized source, e.g., CA
 - ◆ Dynamically via VO structure → delegation
 - ◆ Dynamically via alternative sources, e.g., reputation

Security Services for VO Policy

- Attribute Authority (ATA)
 - ◆ Issue signed attribute assertions (incl. identity, delegation & mapping)
- Authorization Authority (AZA)
 - ◆ Decisions based on assertions & policy
- Use with message/transport level security

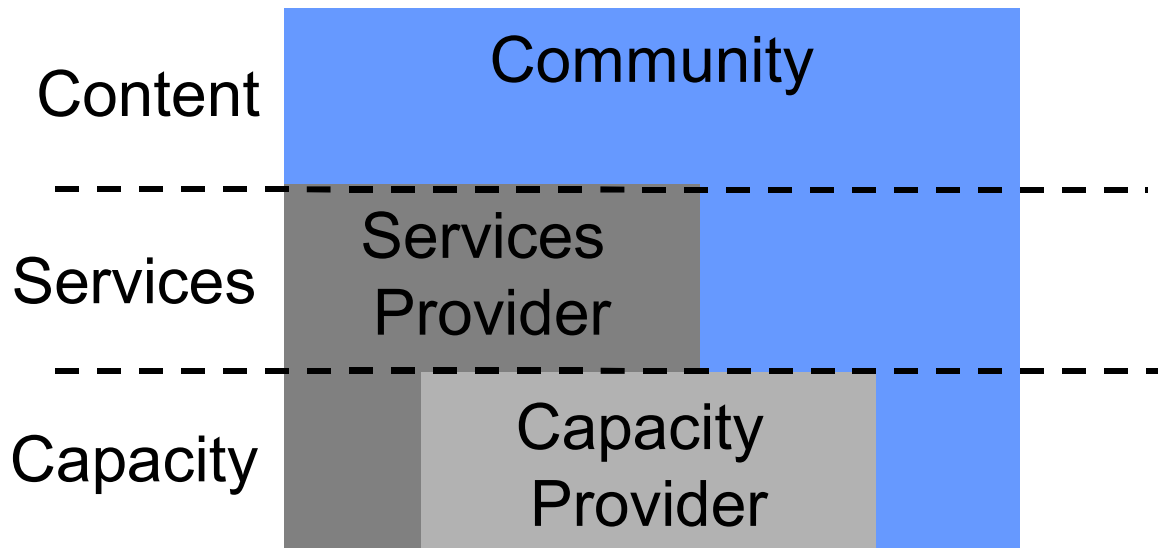


Forming & Operating Scientific Communities

- Define VO membership and roles, & enforce laws and community standards
 - ◆ I.e., policy
- **Build, buy, operate, & share community infrastructure**
 - ◆ **Data, programs, services, computing, storage, instruments**
 - ◆ **Service-oriented architecture**
- Define and perform collaborative work
 - ◆ Use shared infrastructure, roles, & policy
 - ◆ Manage community workflow

Bootstrapping a VO by Assembling Services

- 1) Integrate services from other sources
 - ◆ Virtualize external services as VO services



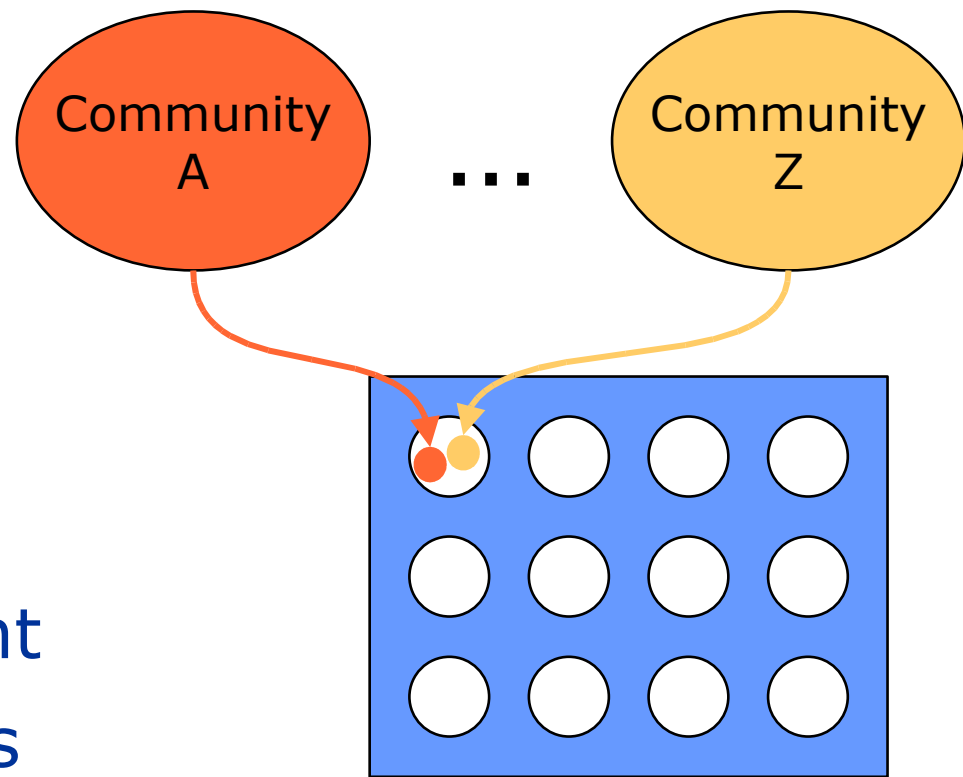
- 2) Coordinate & compose

- ◆ Create new services from existing ones

"Service-Oriented Science", Foster, 2005

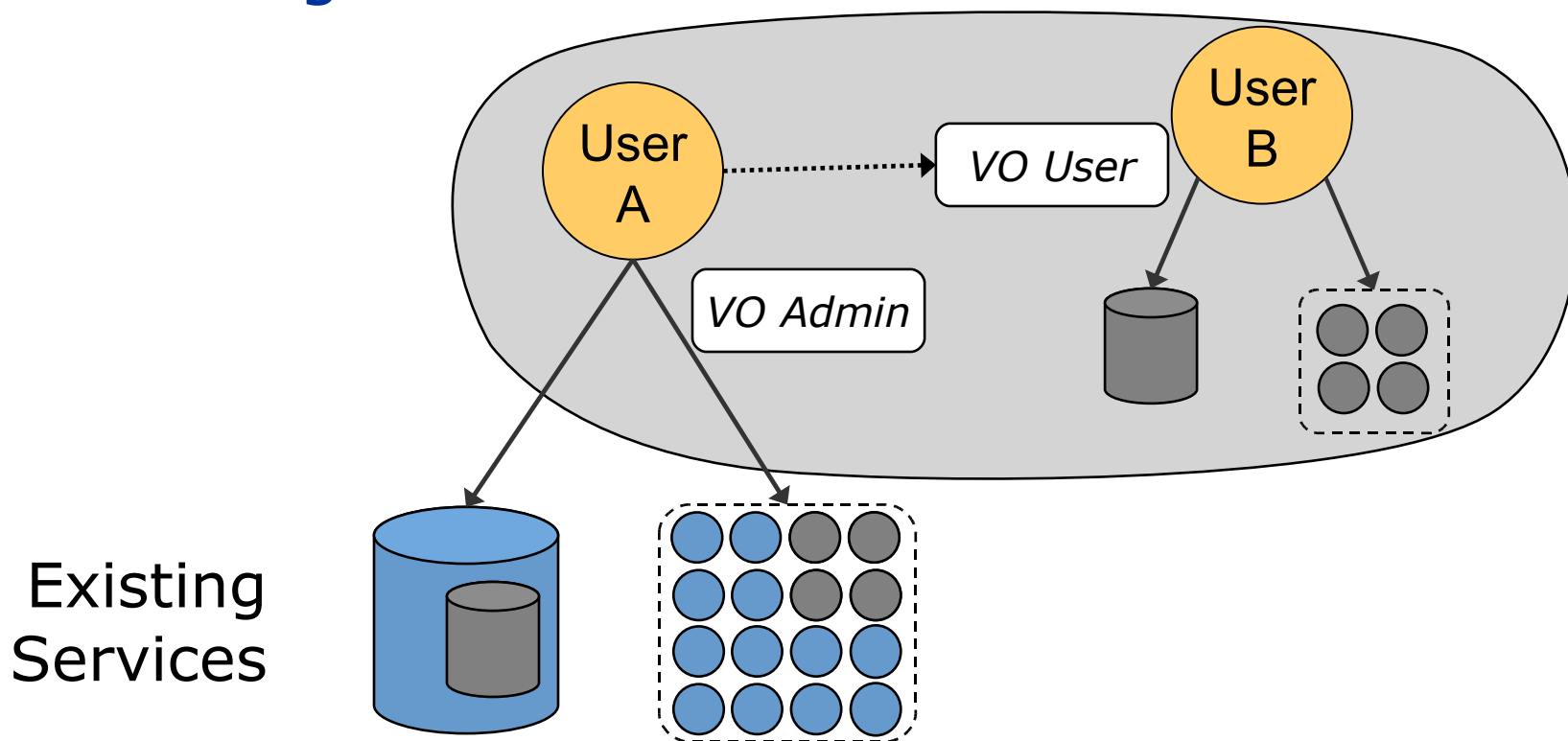
Providing VO Services: (1) Integration from Other Sources

- Negotiate service level agreements
- Delegate and deploy capabilities/services
- Provision to deliver defined capability
- Configure environment
- Host layered functions

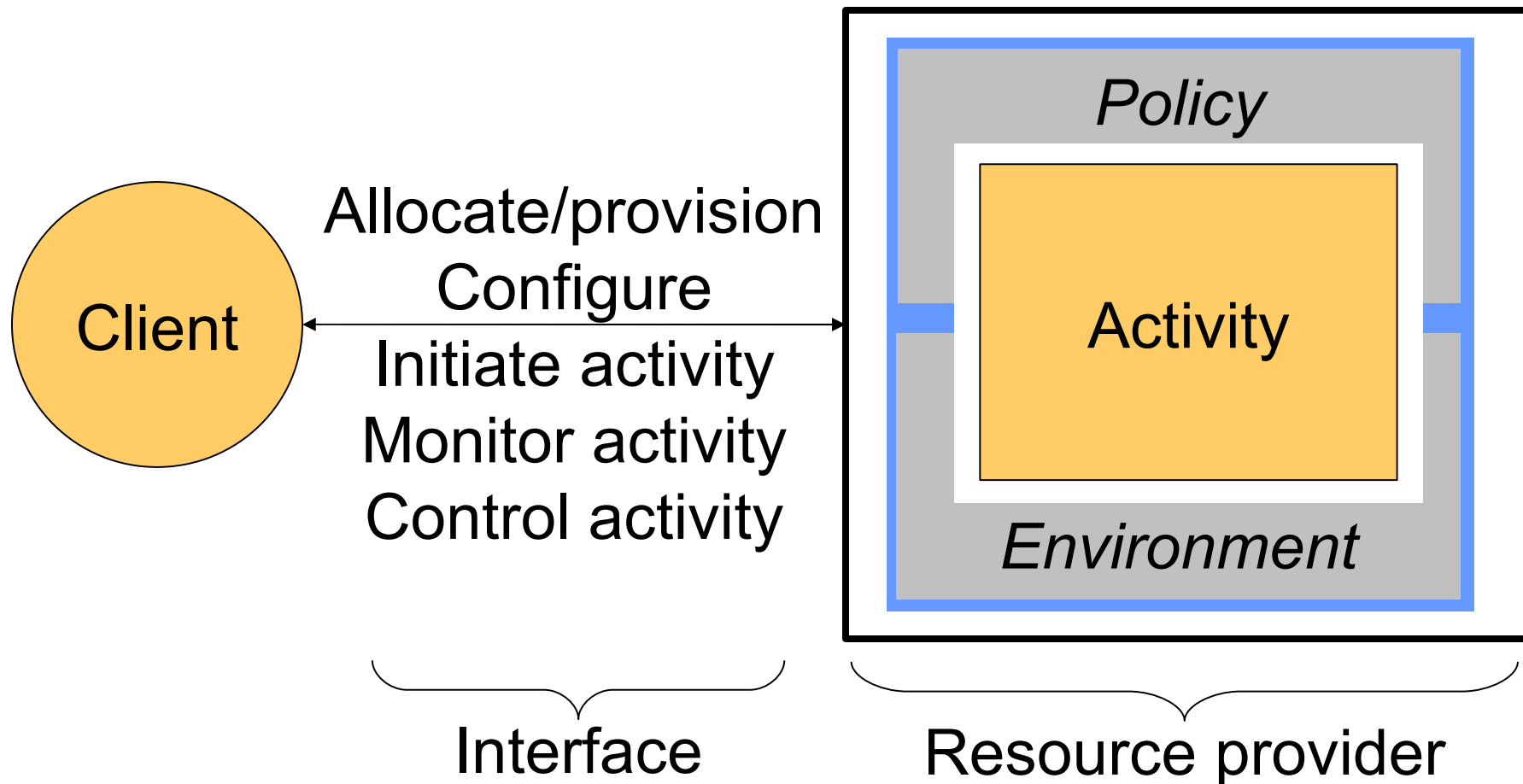


Virtualizing Existing Services into a VO

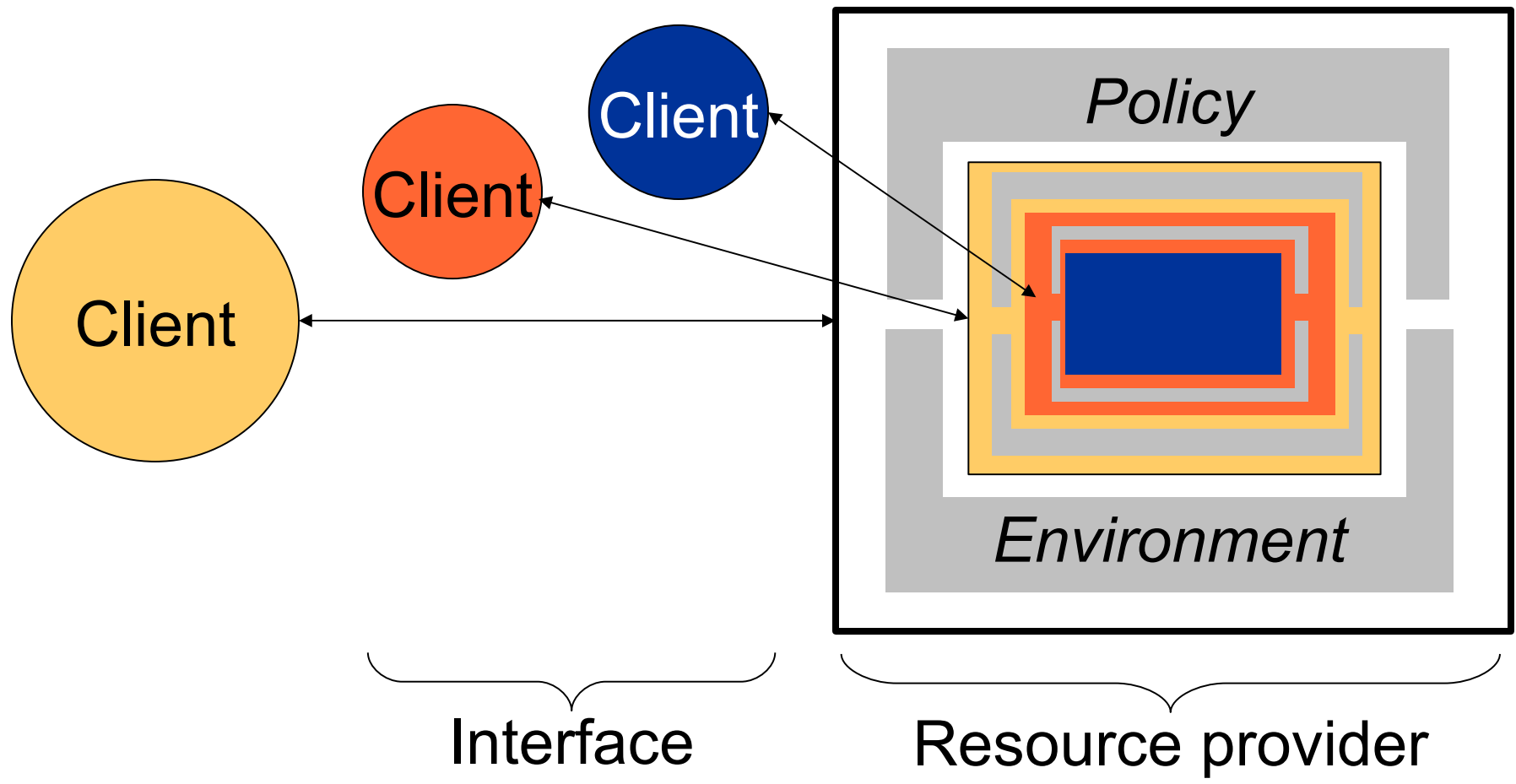
- Establish service agreement with service
 - ◆ E.g., WS-Agreement
- Delegate use to VO user



Deploying New Services



Activities Can Be Nested



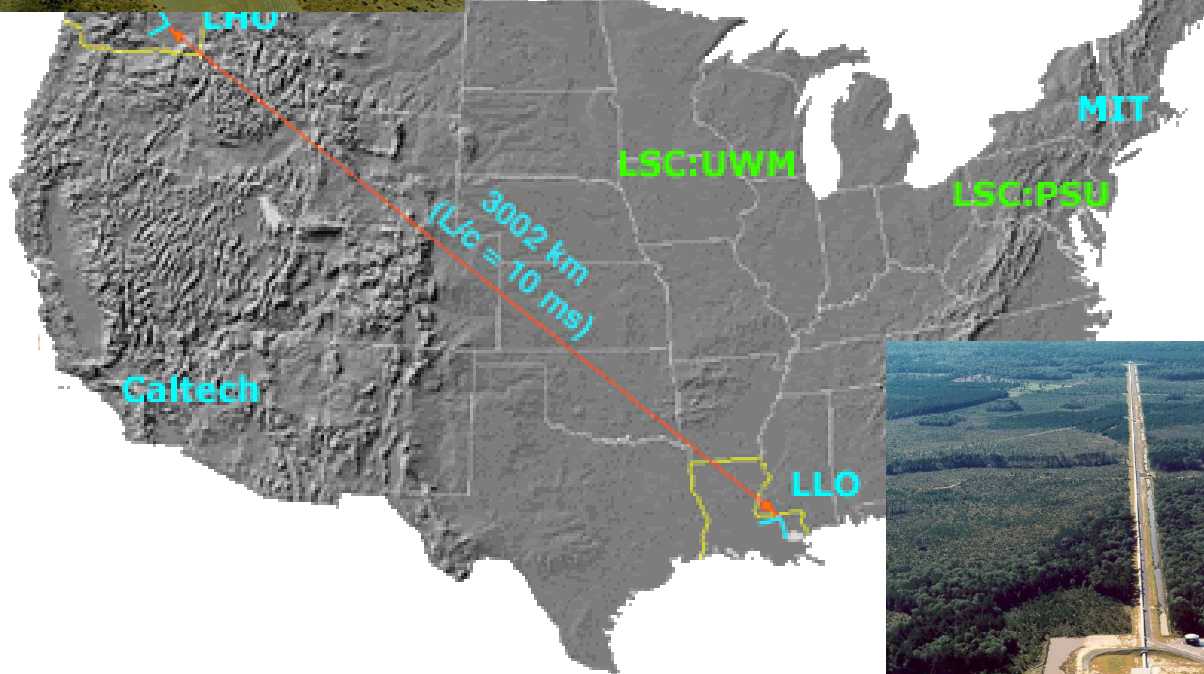
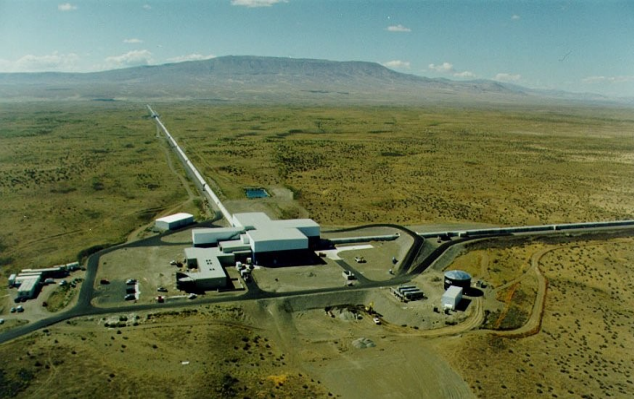
Providing VO Services: (2) Coordination & Composition

- Take a set of provisioned services ...
... & compose to synthesize new behaviors
- This is traditional service composition
 - ◆ But must also be concerned with emergent behaviors, autonomous interactions
 - ◆ See the work of the agent & PlanetLab communities

“Brain vs. Brawn: Why Grids and Agents Need Each Other,”
Foster, Kesselman, Jennings, 2004.

The Globus-Based LIGO Data Grid

LIGO Gravitational Wave Observatory



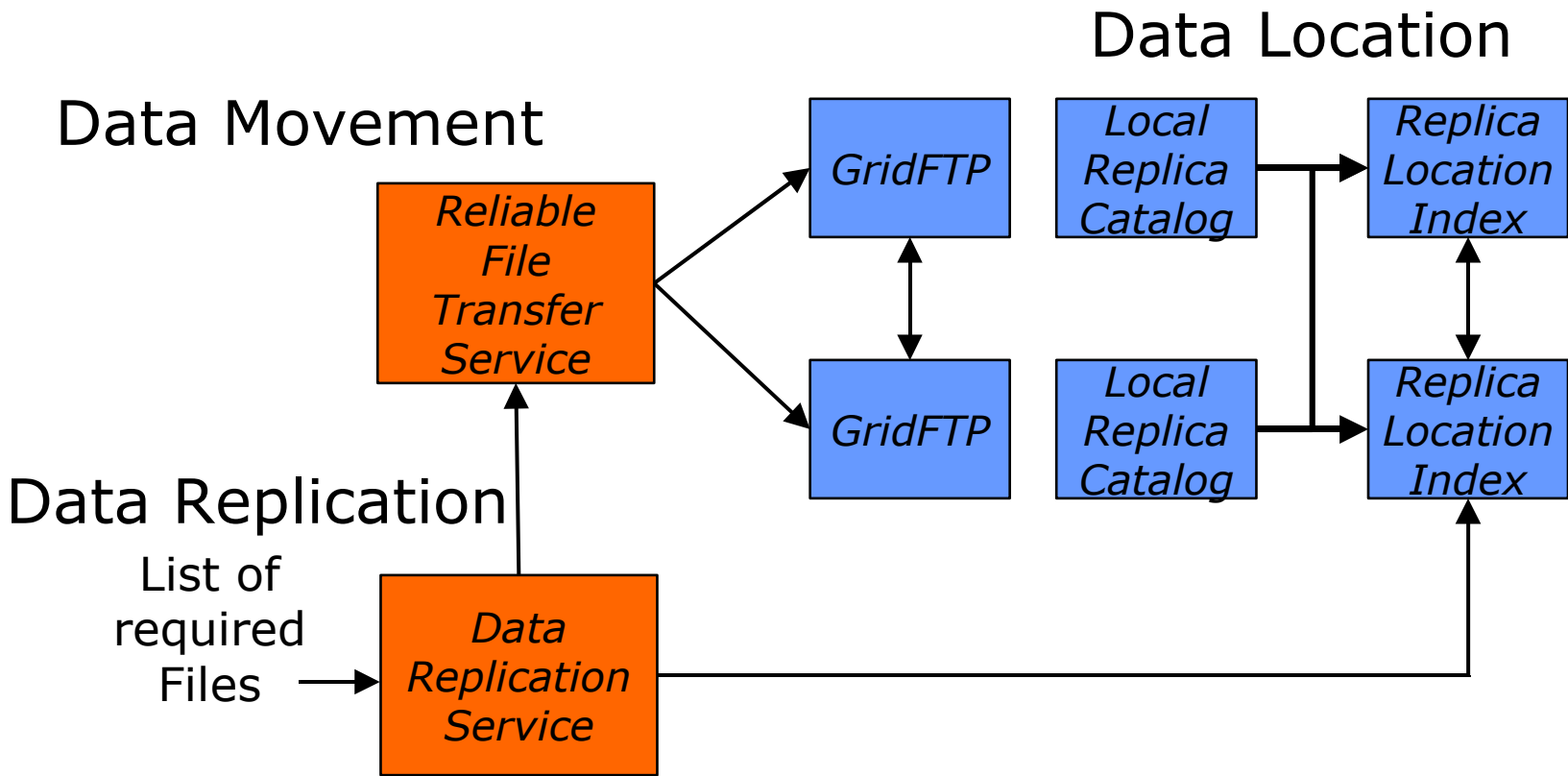
Replicating >1 Terabyte/day to 8 sites
 >40 million replicas so far
 MTBF = 1 month

www.globus.org/solutions



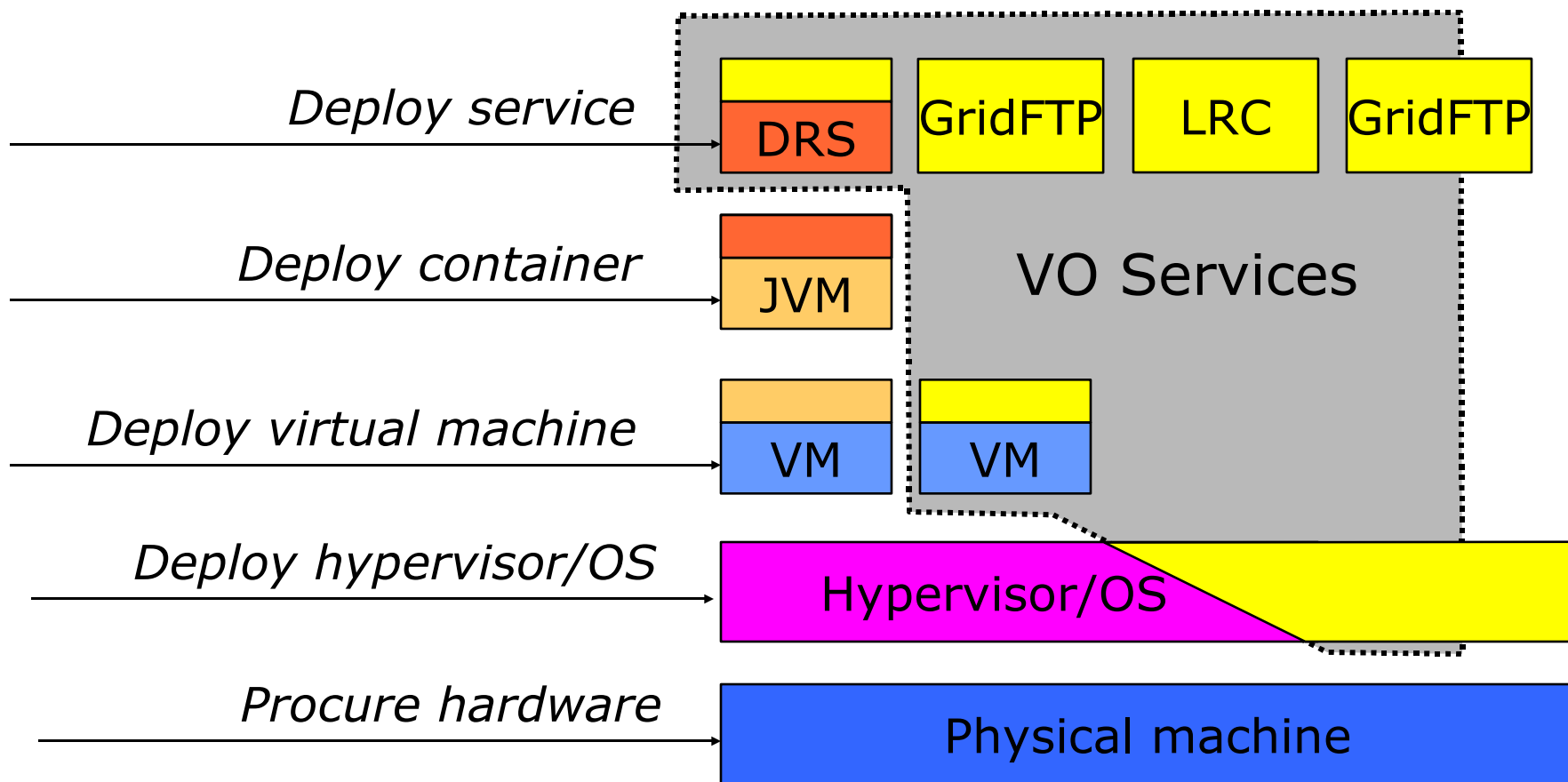
Data Replication Service

- Pull “missing” files to a storage system



“Design and Implementation of a Data Replication Service Based on the Lightweight Data Replicator System,” Chervenak et al., 2005

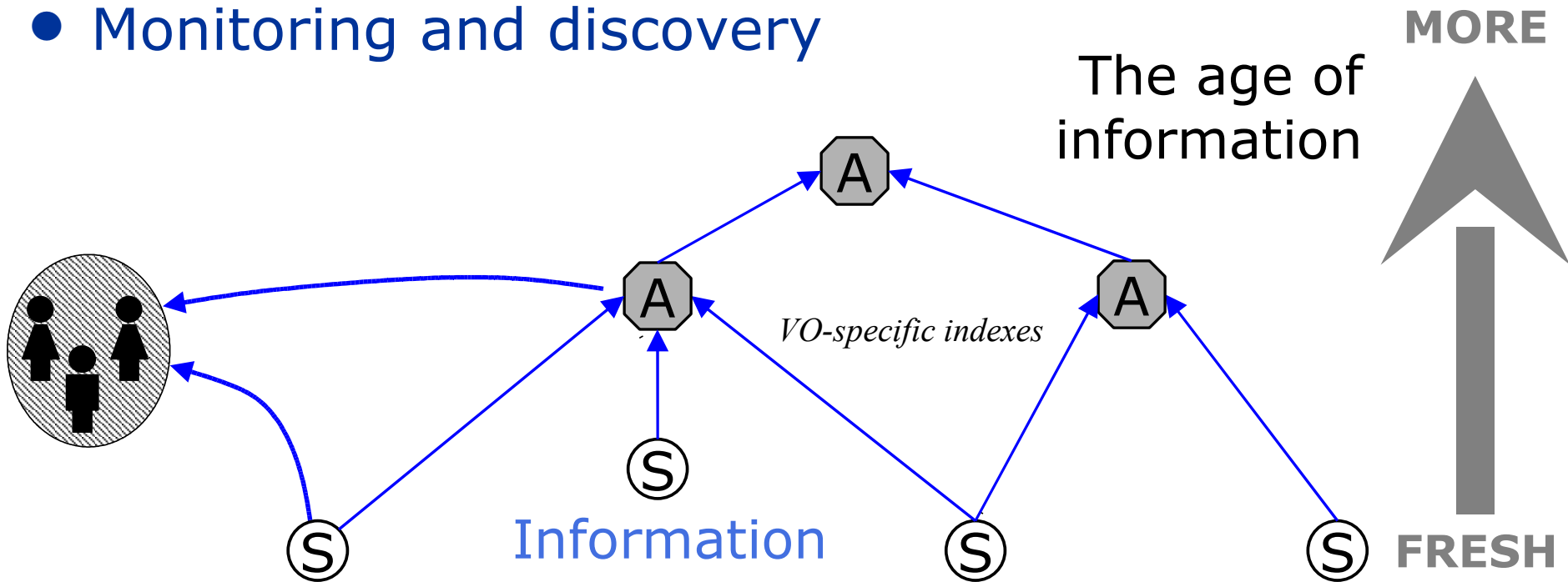
Composing Resources ... Composing Services



Provisioning, management, and monitoring at all levels

Community Commons

- What capabilities are available to VO?
 - ◆ Membership changes, state changes
- Require mechanisms to aggregate and update VO information
- Monitoring and discovery

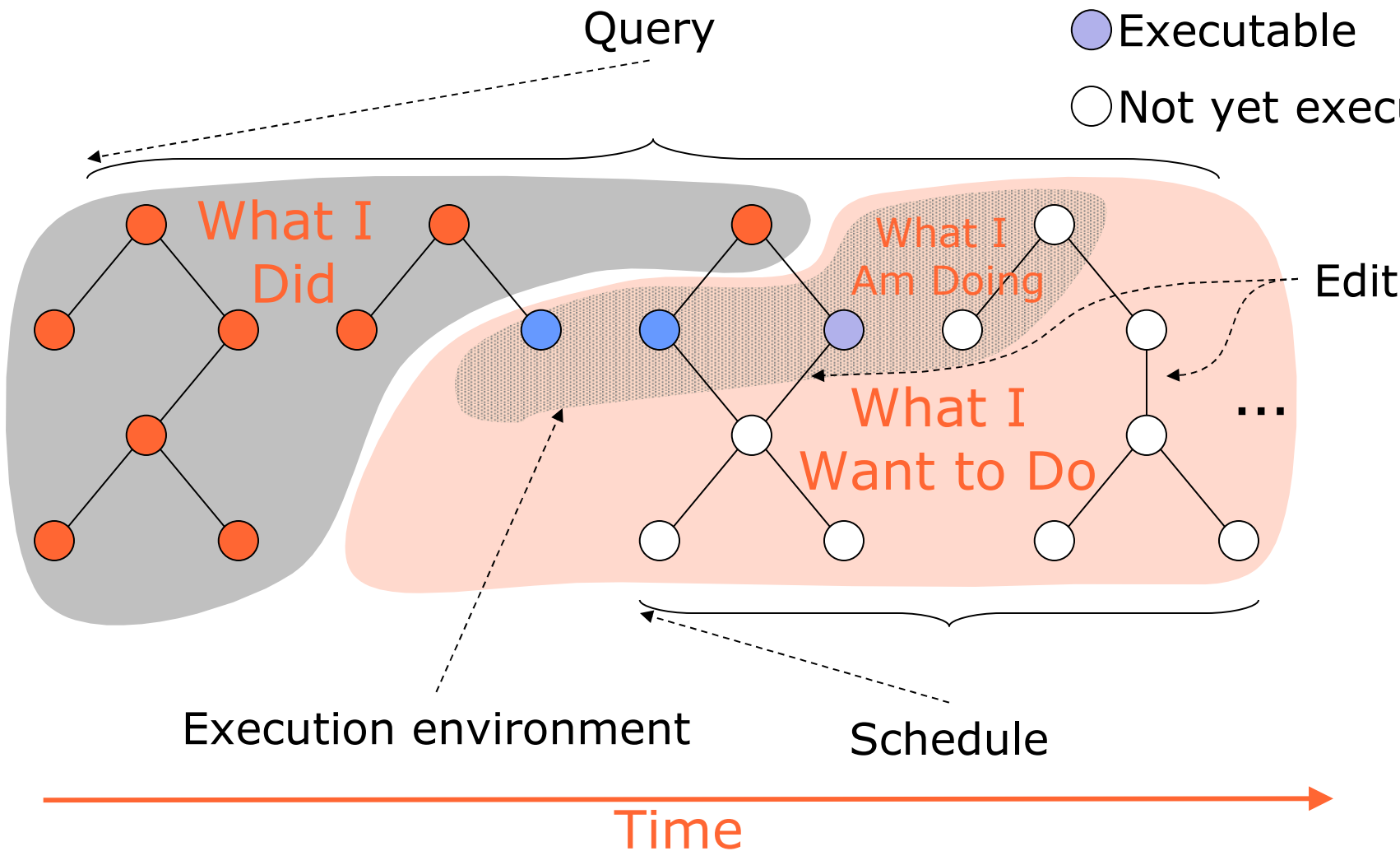


Forming & Operating Scientific Communities

- Define VO membership and roles, & enforce laws and community standards
 - ◆ I.e., policy
- Build, buy, operate, & share community infrastructure
 - ◆ Data, programs, services, computing, storage, instruments
 - ◆ Service-oriented architecture
- **Define and perform collaborative work**
 - ◆ **Use shared infrastructure, roles, & policy**
 - ◆ **Manage community workflow**

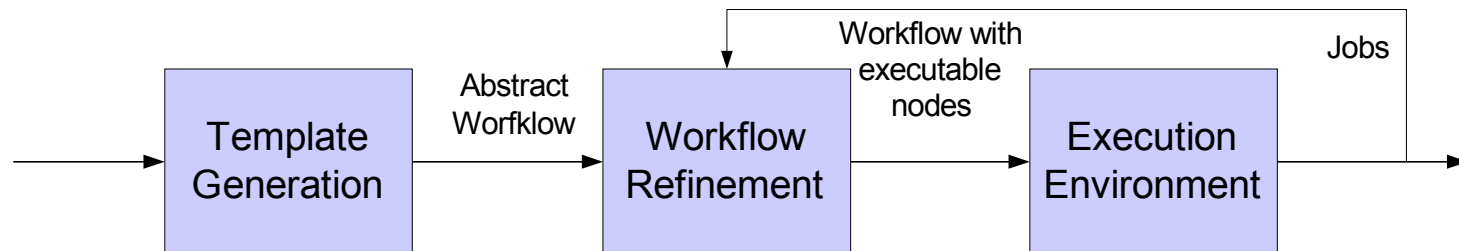
Collaborative Work

- Executed
- Executing
- Executable
- Not yet executable

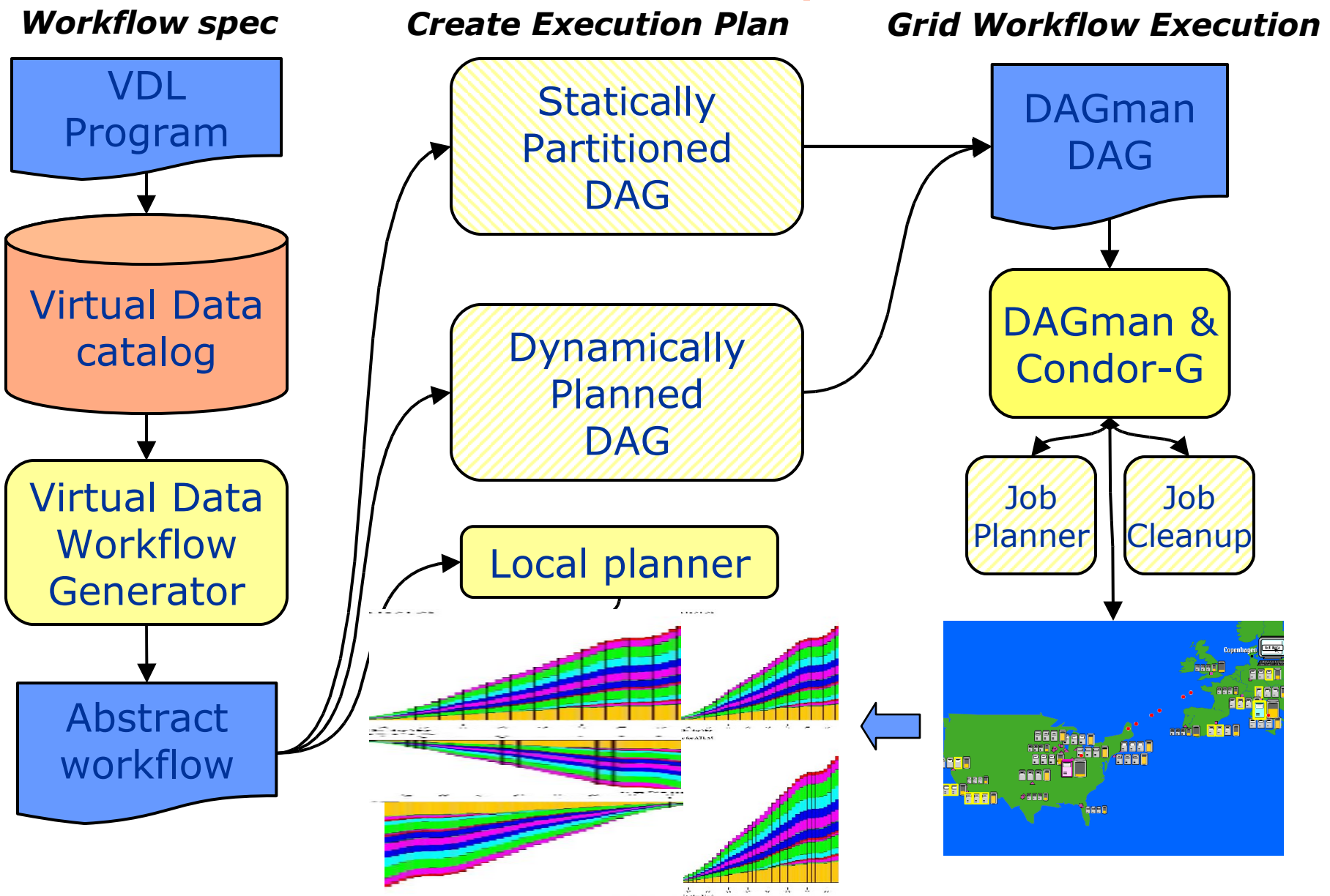


Managing Collaborative Work

- Process as “workflow,” at different scales, e.g.:
 - ◆ Run 3-stage pipeline
 - ◆ Process data flowing from expt over a year
 - ◆ Engage in interactive analysis
- Need to keep track of:
 - ◆ *What I want to do* (will evolve with new knowledge)
 - ◆ *What I am doing now* (evolve with system config.)
 - ◆ *What I did* (persistent; a source of information)



Trident: The GriPhyN Virtual Data System

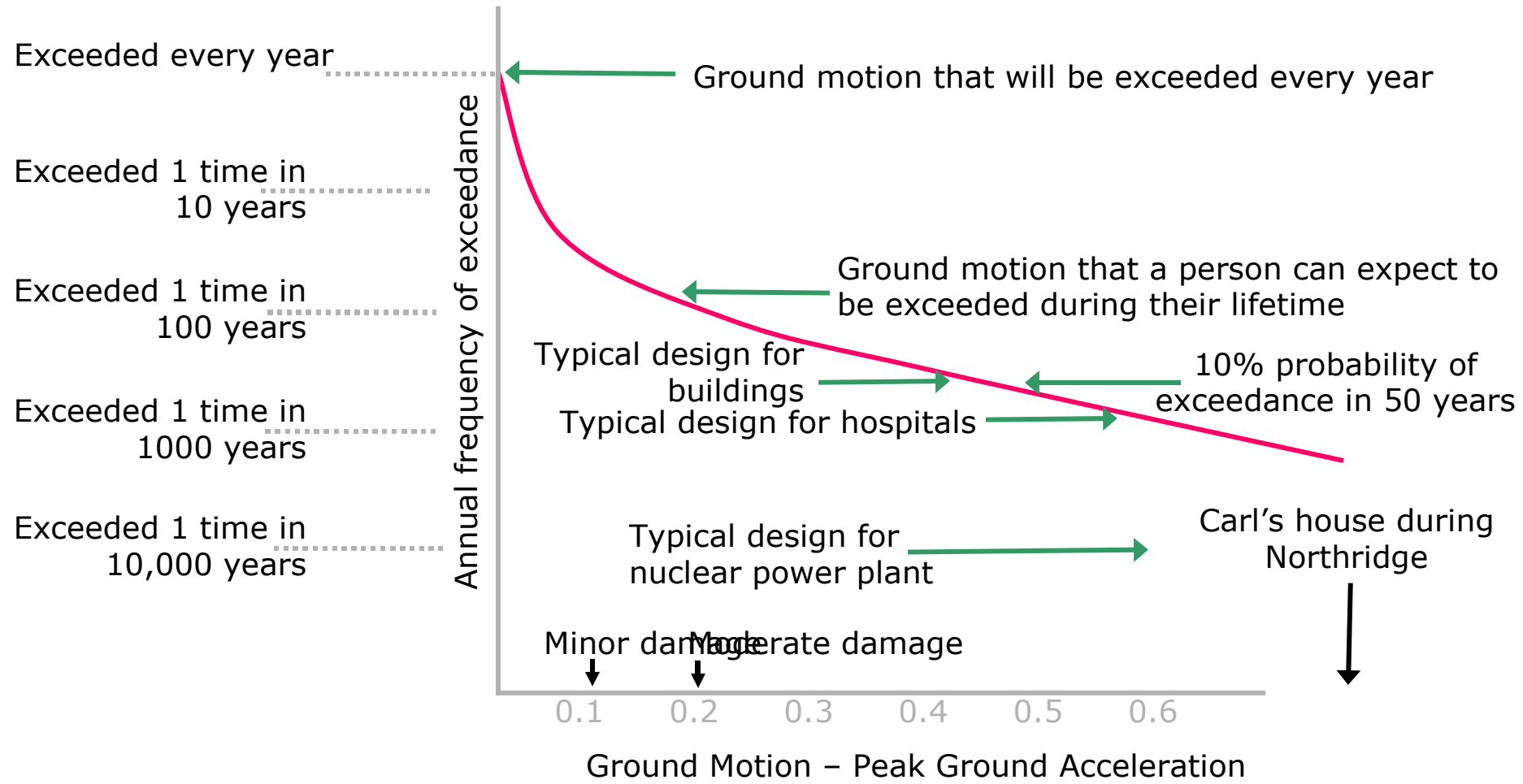


Workflow Generation

- **Given: desired result and constraints**
 - ◆ desired result (high-level, metadata description)
 - ◆ application components
 - ◆ resources in the Grid (dynamic, distributed)
 - ◆ constraints & preferences on solution quality
- **Find: an executable job workflow**
 - ◆ A configuration that generates the desired result
 - ◆ A specification of resources to be used
 - ◆ Sequence of operations: create agreement, move data, request operation
- **May create workflow incrementally as information becomes available**

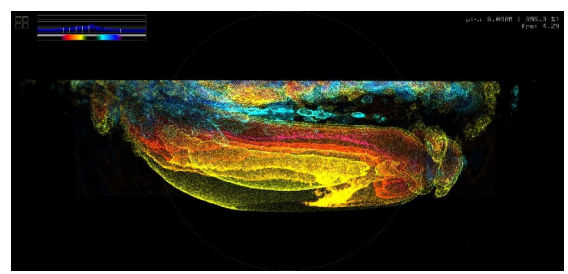
"Mapping Abstract Complex Workflows onto Grid Environments," Deelman, Blythe, Gil, Kesselman, Mehta, Vahi, Arbree, Cavanaugh, Blackburn, Lazzarini, Koranda, 2003.

Seismic Hazard Curve

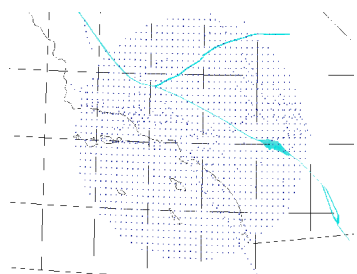


SCEC Cybershake

- Calculate hazard curves by generating synthetic seismograms from estimated rupture forecast

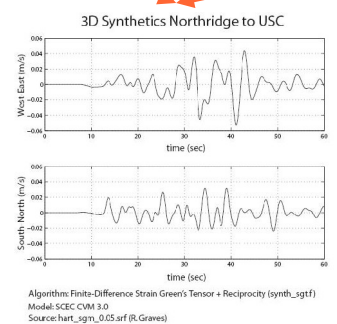
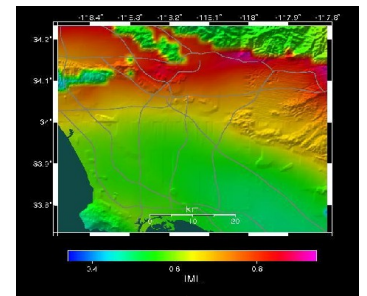


Strain Green Tensor

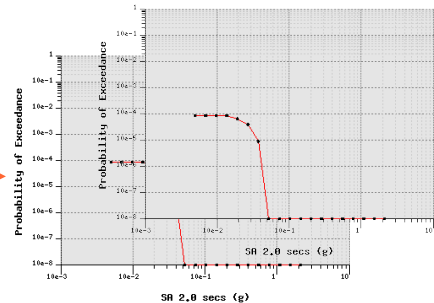


Rupture Forecast

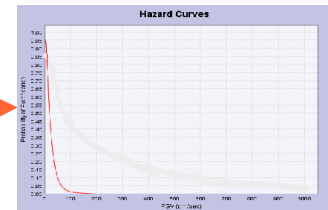
Hazard Map



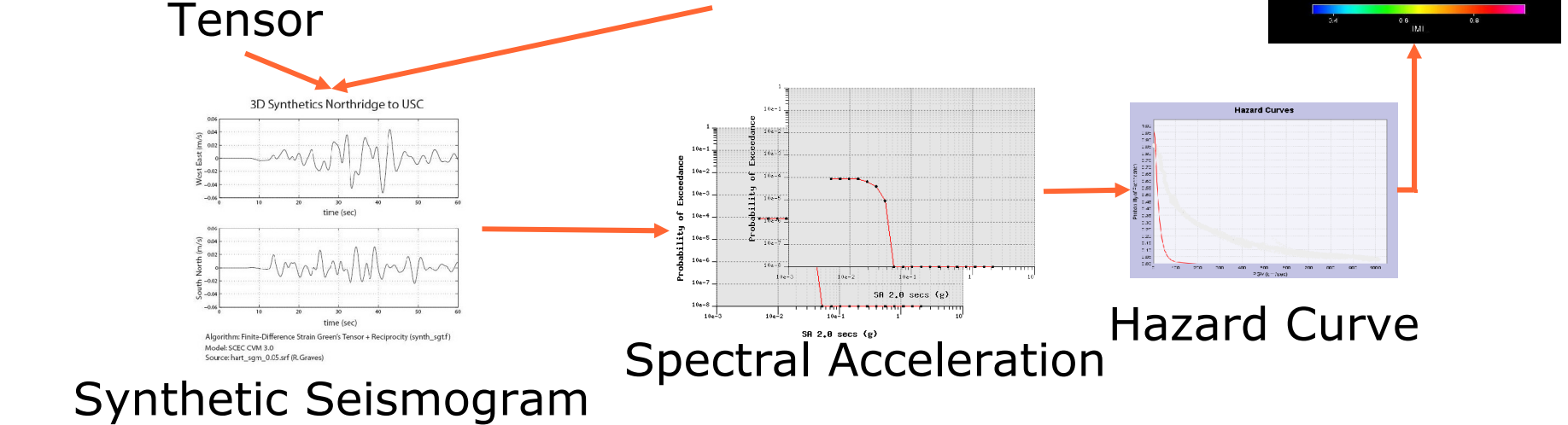
Synthetic Seismogram



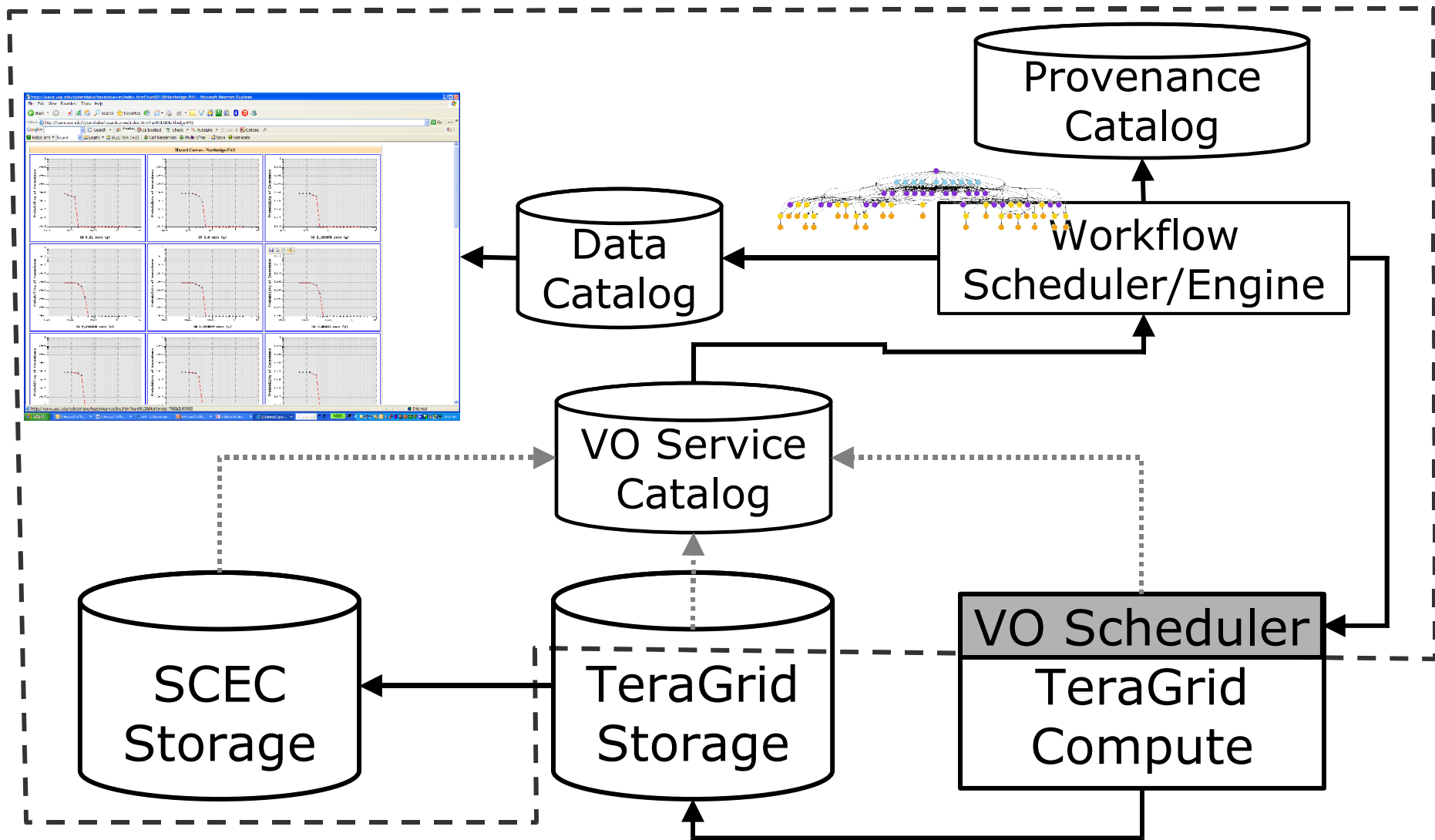
Spectral Acceleration



Hazard Curve



Cybershake on the SCEC VO





Runs on the
TeraGrid in 2005

SCEC is using Pegasus among other tools to generate hazard maps
for the LA area

Summary (1): Community Services

- Community roll, city hall, permits, licensing & police force
 - ◆ Assertions, policy, attribute & authorization services
- Directories, maps
 - ◆ Information services
- City services: power, water, sewer
 - ◆ Deployed services
- Shops, businesses
 - ◆ Composed services
- Day-to-day activities
 - ◆ Workflows, visualization
- Tax board, fees, economic considerations
 - ◆ Barter, planned economy, eventually markets

Summary (2)

- **Community based science** will be the norm
 - ◆ Requires collaborations across sciences— including computer science
- Many different **types of communities**
 - ◆ Differ in coupling, membership, lifetime, size
- Must think **beyond science stovepipes**
 - ◆ Increasingly the community infrastructure will become the scientific observatory
- Scaling requires a **separation of concerns**
 - ◆ Providers of resources, services, content
- Small set of **fundamental mechanisms** required to build communities

The Globus Toolkit

- Background
- Globus Toolkit
- Future directions
- Related tools
- Opportunities for collaboration



On April 29, 2005 the
Globus Alliance released
the finest version of the
Globus Toolkit to date!

Don't take our word for it!

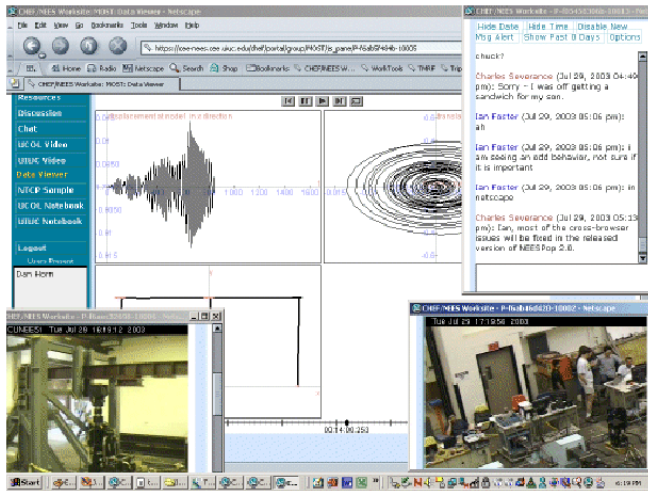
Read the **UK eScience Evaluation of GT4**

www.nesc.ac.uk/technical_papers/UKeS-2005-03.pdf
(Reachable from www.globus.org, under "News")

The Role of the Globus Toolkit

- A collection of solutions to problems that come up frequently when building collaborative distributed applications
- Heterogeneity
 - ◆ A focus, in particular, on overcoming heterogeneity for application developers
- Standards
 - ◆ We capitalize on and encourage use of existing standards (IETF, W3C, OASIS, GGF)
 - ◆ GT also includes reference implementations of new/proposed standards in these organizations

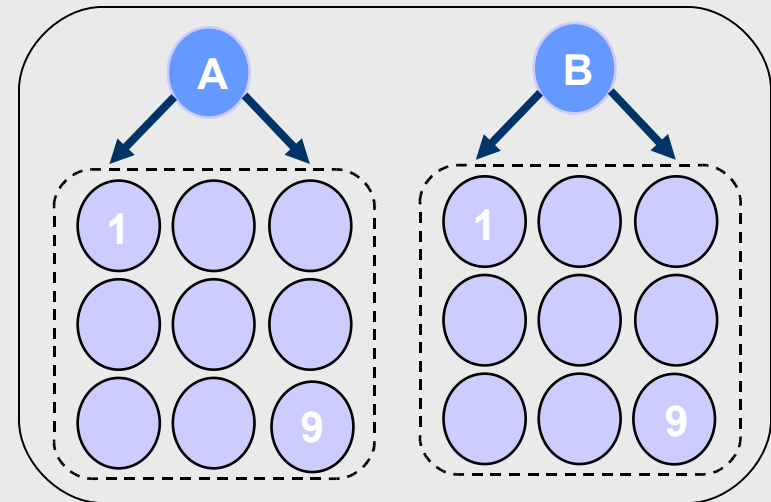
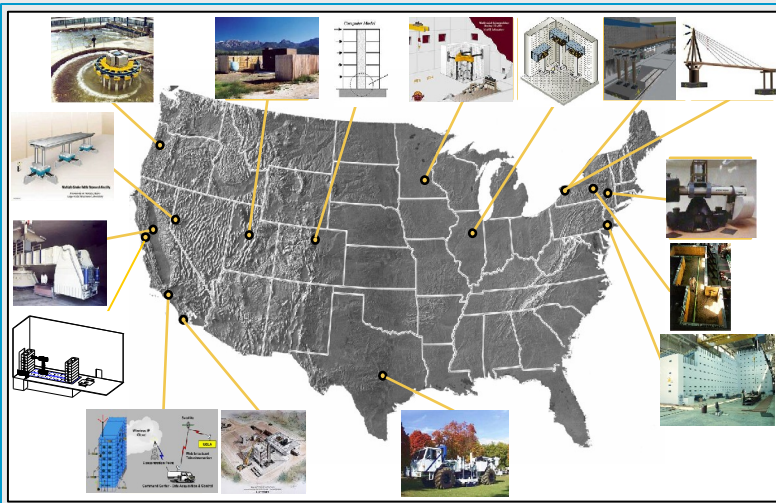
The Application-Infrastructure Gap



**Dynamic
and/or
Distributed
Applications**



Shared Distributed Infrastructure



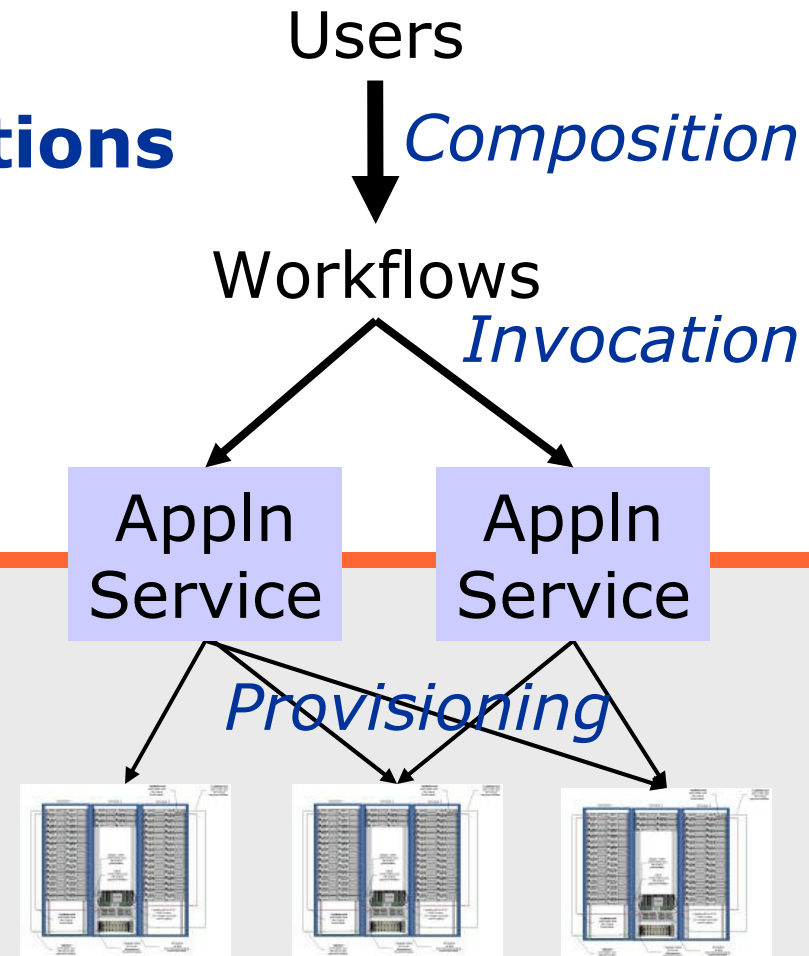
Bridging the Gap: Grid Infrastructure

- Service-oriented **applications**

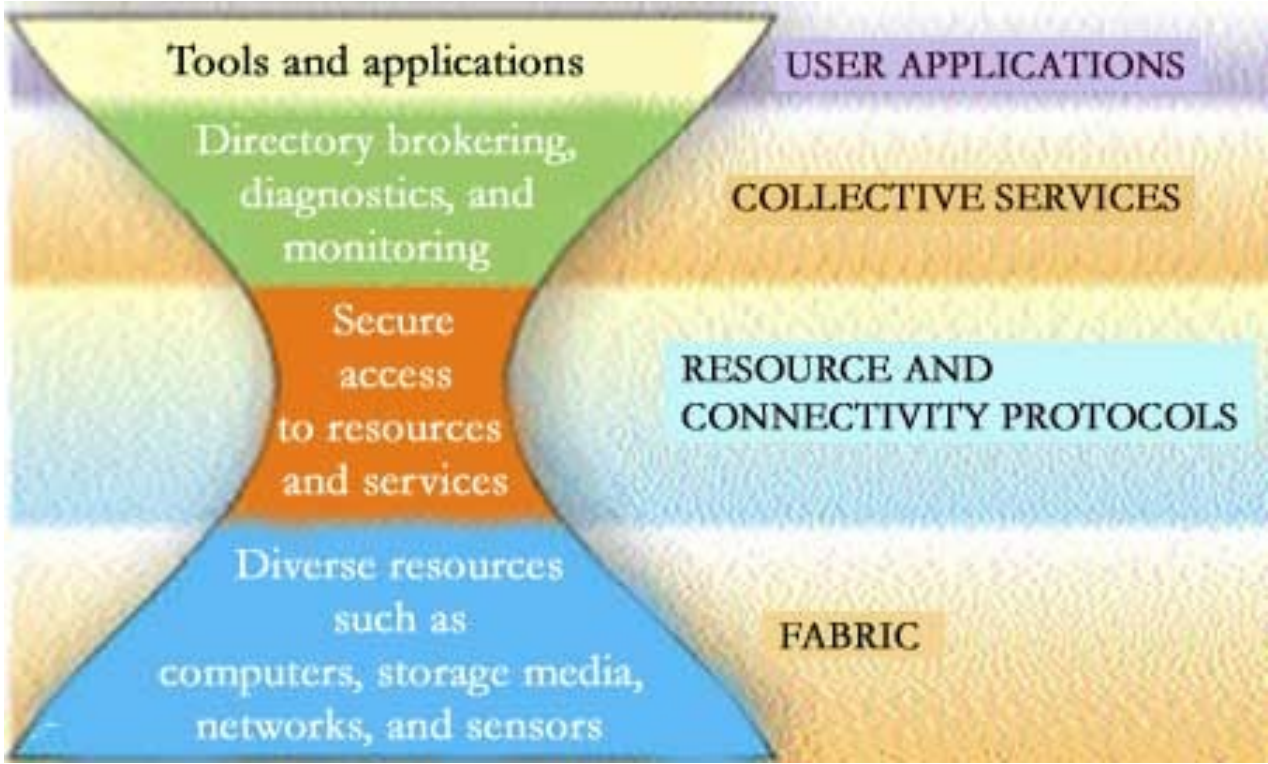
- ◆ Wrap applications as services
- ◆ Compose applications into workflows

- Service-oriented **Grid infrastructure**

- ◆ Provision physical resources to support application workloads



Layers in the Grid



A Typical eScience Use of Globus: Network for Earthquake Eng. Simulation

The central map shows the United States with yellow dots marking locations of research facilities. Surrounding the map are several inset images:

- Top left: A circular shake table facility.
- Top center: A building with a large window.
- Top right: A 'Computer Model' diagram of a multi-story building.
- Middle right: A laboratory with a large yellow machine.
- Far right: A server rack.
- Bottom right: A large industrial table.
- Bottom center: A satellite dish.
- Bottom left: A 'Multiple Shake Table Research Facility' at UCSD.
- Far left: A large white machine.
- Bottom left: A 'Science Communication' diagram.
- Bottom left: A 'Wireless IP Cloud' diagram with a 'UCLA' label.
- Bottom left: A construction site.
- Bottom right: Screenshots of a web browser displaying a chat window and a data viewer with a seismic waveform and a circular pattern.

Links instruments, data, computers, people

The screenshots show a web browser interface with a chat window and a data viewer. The chat window shows a conversation between Charles Severance and Ian Foster. The data viewer shows a seismic waveform and a circular pattern.

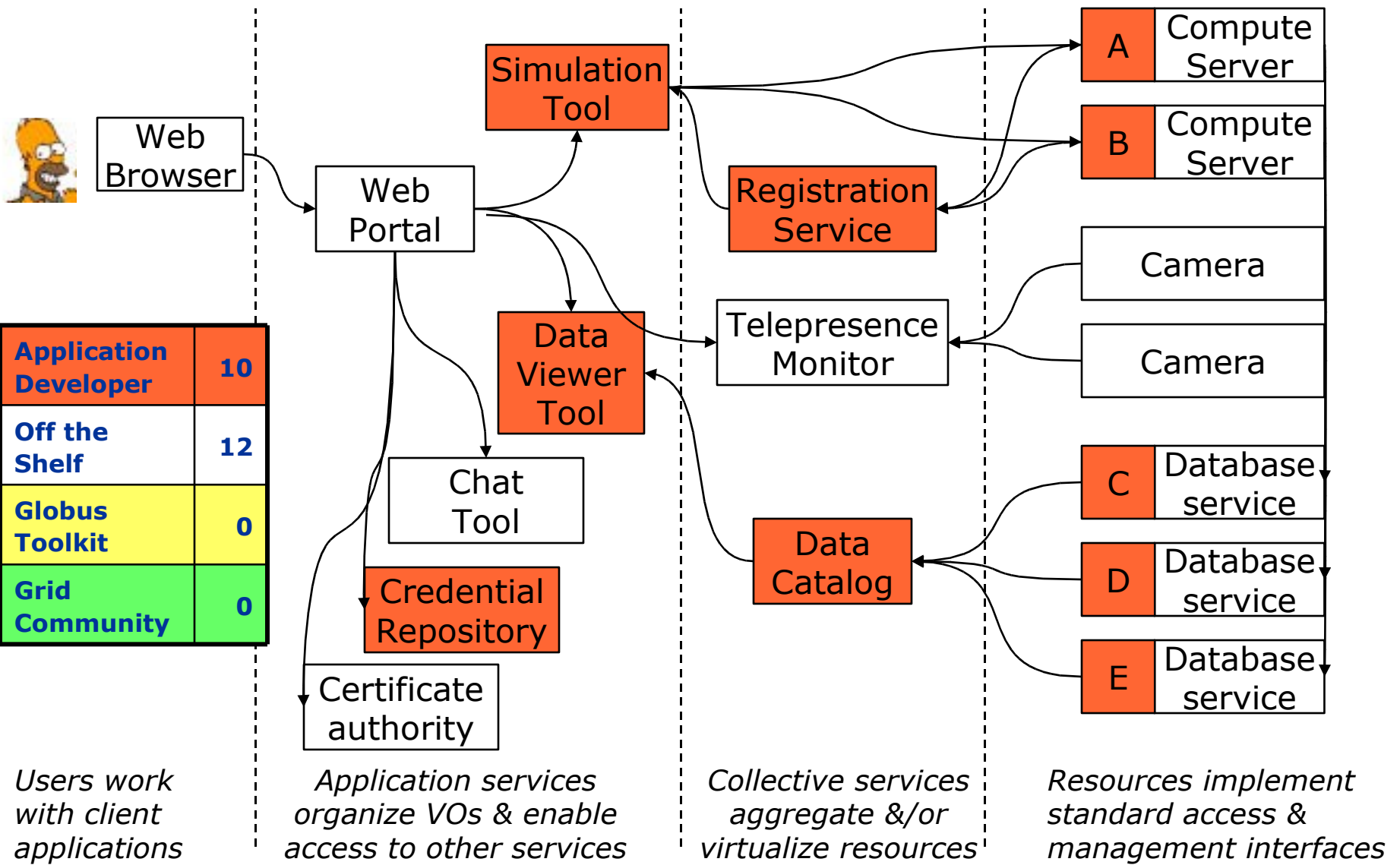
Chat window content:

- Charles Severance (Jul 29, 2003 04:40 pm): Sorry - I was off getting a sandwich for my son.
- Ian Foster (Jul 29, 2003 05:06 pm): ah
- Ian Foster (Jul 29, 2003 05:06 pm): I am seeing an odd behavior, not sure if it is important
- Ian Foster (Jul 29, 2003 05:06 pm): in netscape
- Charles Severance (Jul 29, 2003 05:13 pm): Ian, most of the cross-browser issues will be fixed in the released version of NEEES Pop 2.0.

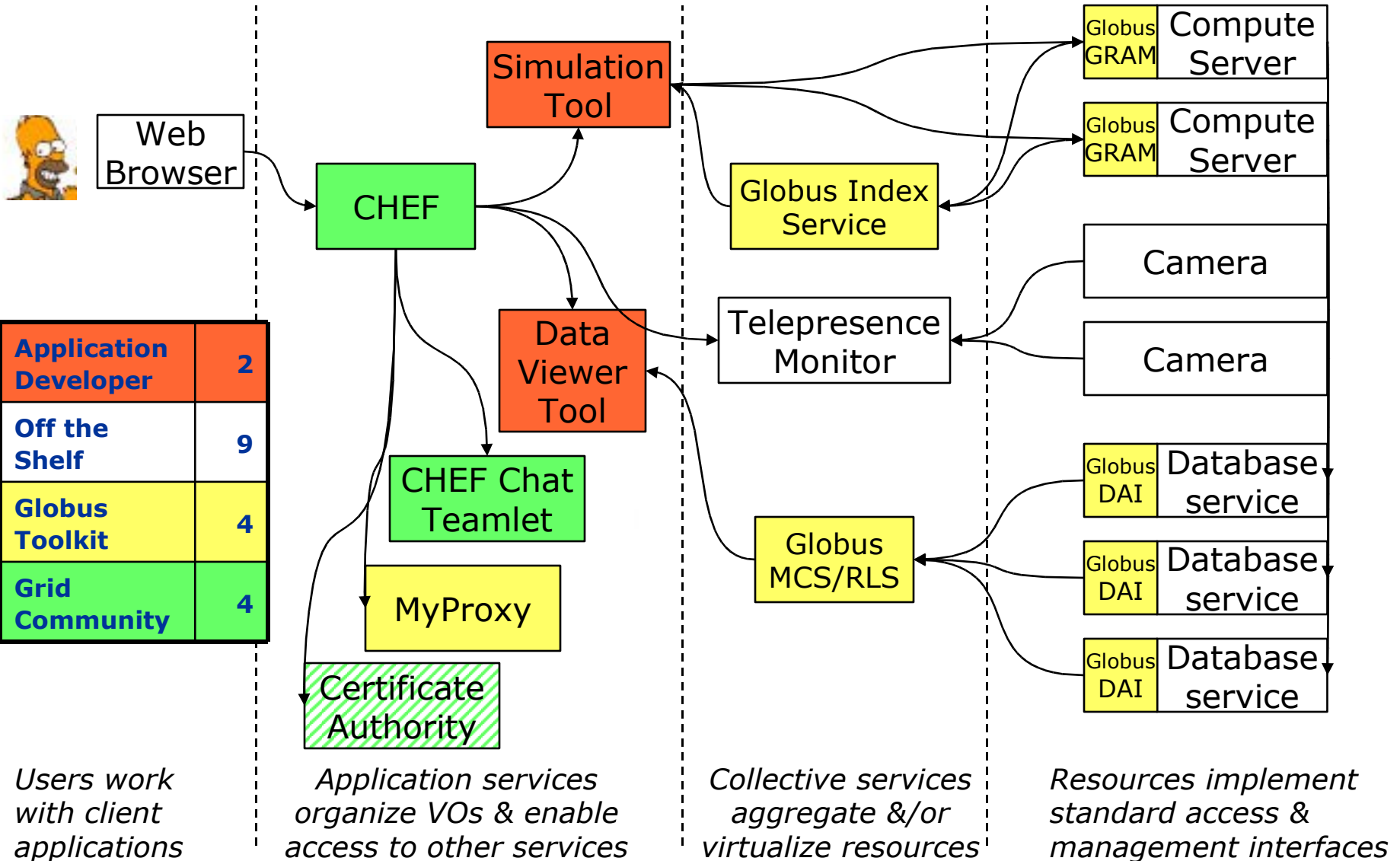
Data viewer content:

- Distances
- Chat
- UCOL Video
- UTUC Video
- Data Viewer
- NTCP Sample
- UCOL Netbook
- UTUC Netbook
- Logout
- User's Profile
- Dam Horn

Without the Globus Toolkit



With the Globus Toolkit



The Globus Toolkit: “Standard Plumbing” for the Grid

- *Not* turnkey solutions, but *building blocks & tools* for application developers & system integrators
 - ◆ Some components (e.g., file transfer) go farther than others (e.g., remote job submission) toward end-user relevance
- Easier to reuse than to reinvent
 - ◆ Compatibility with other Grid systems comes for free
- Today the majority of the GT public interfaces are usable by application developers and system integrators
 - ◆ Relatively few end-user interfaces
 - ◆ In general, not intended for direct use by end users (scientists, engineers, marketing specialists)

Globus is Grid Infrastructure

- Software for Grid infrastructure
 - ◆ Service enable new & existing resources
 - ◆ E.g., GRAM on computer, GridFTP on storage system, custom application service
 - ◆ Uniform abstractions & mechanisms
- Tools to build applications that exploit Grid infrastructure
 - ◆ Registries, security, data management, ...
- Open source & open standards
 - ◆ Each empowers the other
- Enabler of a rich tool & service ecosystem

An eBusiness Use of Globus: SAP Demonstration @ GlobusWorld

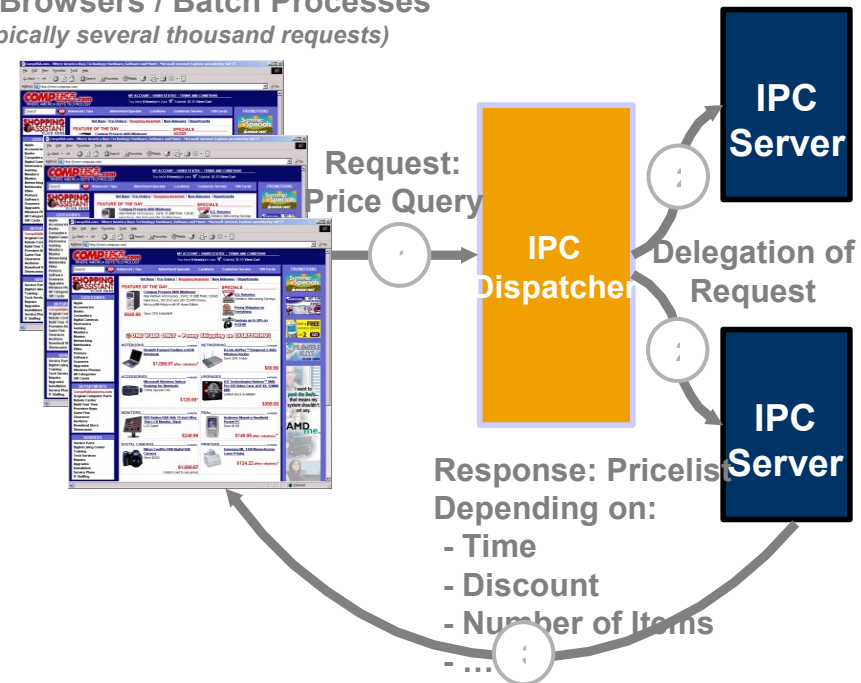
- 3 Globus-enabled applns:

- ◆ CRM: Internet Pricing Configurator (IPC)
- ◆ CRM: Workforce Management (WFM)
- ◆ SCM: Advanced Planner & Optimizer (APO)

- Applications modified to:

- ◆ Adjust to varying demand & resources
- ◆ Use Globus to discover & provision resources

Web Browsers / Batch Processes
(typically several thousand requests)



SAP AG R/3 Internet Pricing
& Configurator (IPC)

The Globus Toolkit is a Collection of Components

- A set of loosely-coupled components, with:
 - ◆ Services and clients
 - ◆ Libraries
 - ◆ Development tools
- GT components are used to build Grid-based applications and services
 - ◆ GT can be viewed as a Grid SDK
- GT components can be categorized across two different dimensions
 - ◆ By broad domain area
 - ◆ By protocol support

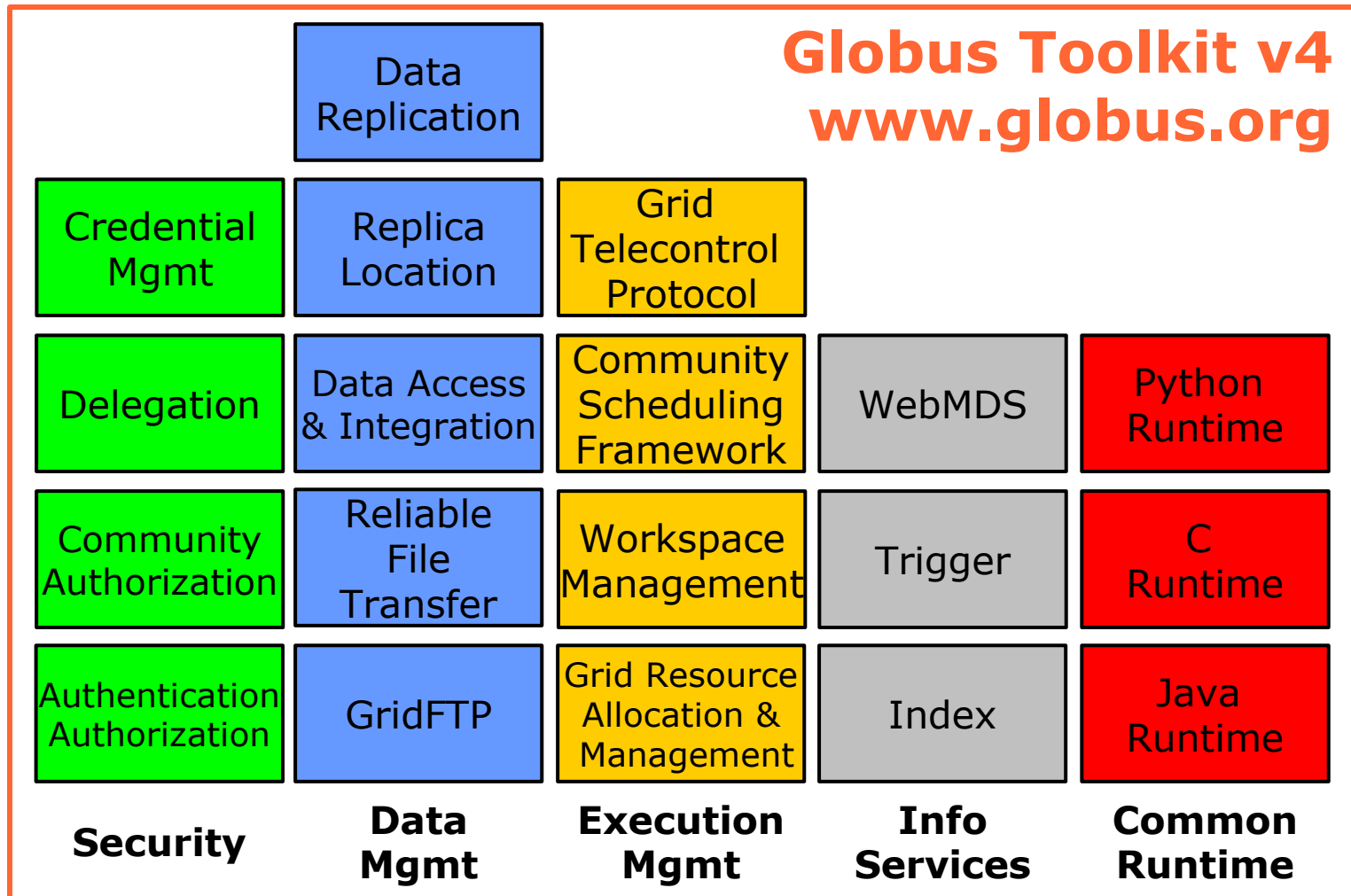
Our Goals for GT4

- Usability, reliability, scalability, ...
 - ◆ Web service components have quality equal or superior to pre-WS components
 - ◆ Documentation at acceptable quality level
- Consistency with latest standards (WS-*, WSRF, WS-N, etc.) and Apache platform
 - ◆ WS-I Basic Profile compliant
 - ◆ WS-I Basic Security Profile compliant
- New components, platforms, languages
 - ◆ And links to larger Globus ecosystem

GT Domain Areas

- Core runtime
 - ◆ Infrastructure for building new services
- Security
 - ◆ Apply uniform policy across distinct systems
- Execution management
 - ◆ Provision, deploy, & manage services
- Data management
 - ◆ Discover, transfer, & access large data
- Monitoring
 - ◆ Discover & monitor dynamic services

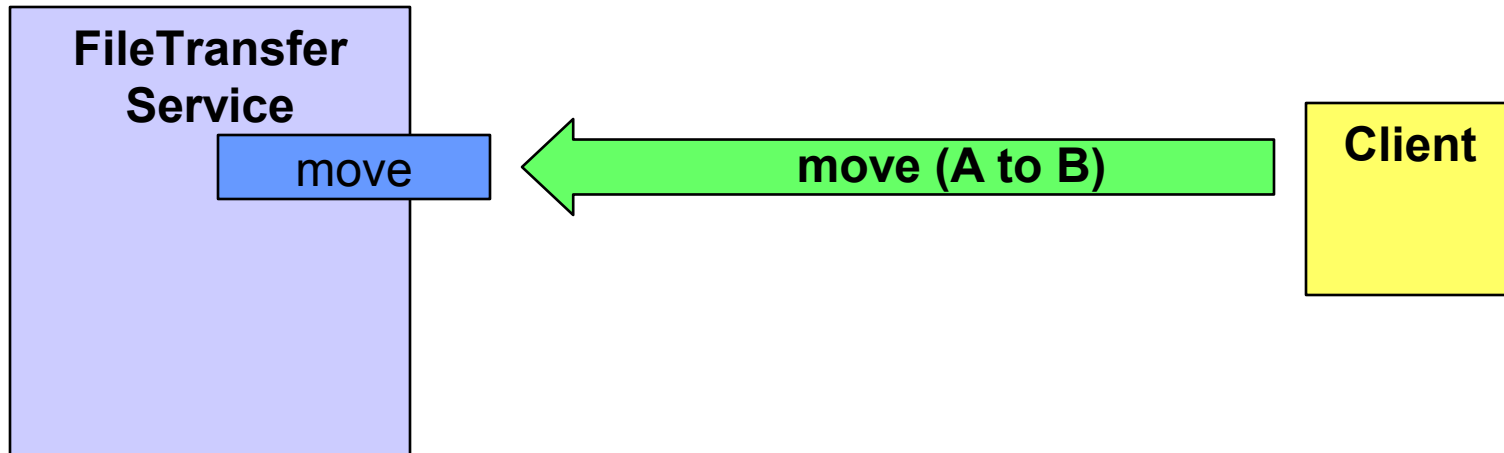
Globus Toolkit: Open Source Grid Infrastructure



GT Protocols

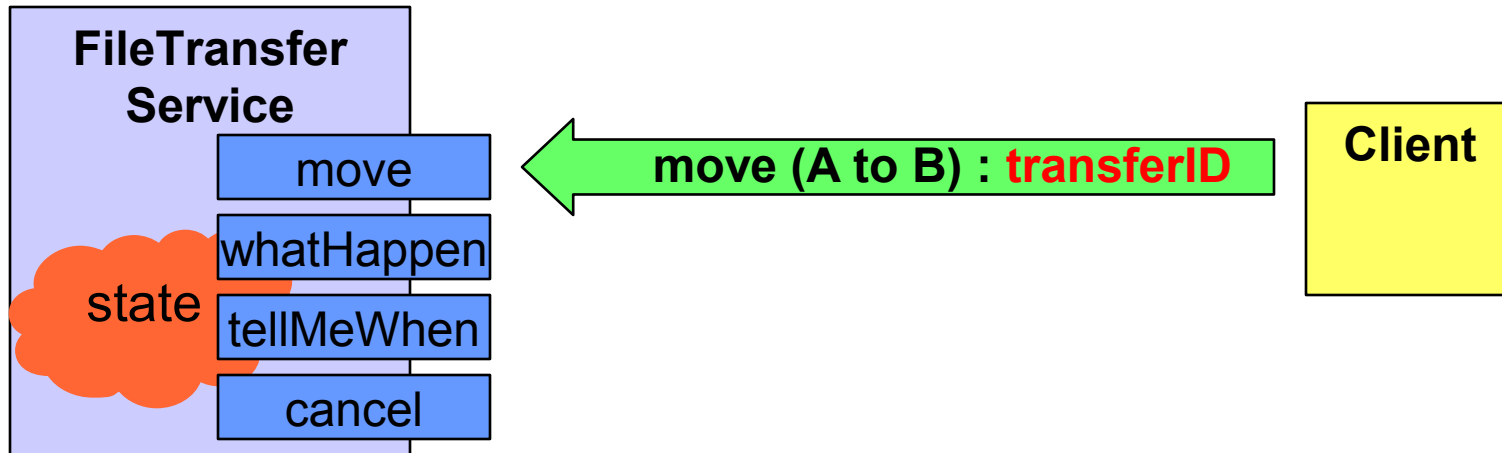
- Web service protocols
 - ◆ WSDL, SOAP
 - ◆ WS Addressing, WSRF, WSN
 - ◆ WS Security, SAML, XACML
 - ◆ WS-Interoperability profile
- Non Web service protocols
 - ◆ Standards-based, such as GridFTP
 - ◆ Custom

“Stateless” vs. “Stateful” Services



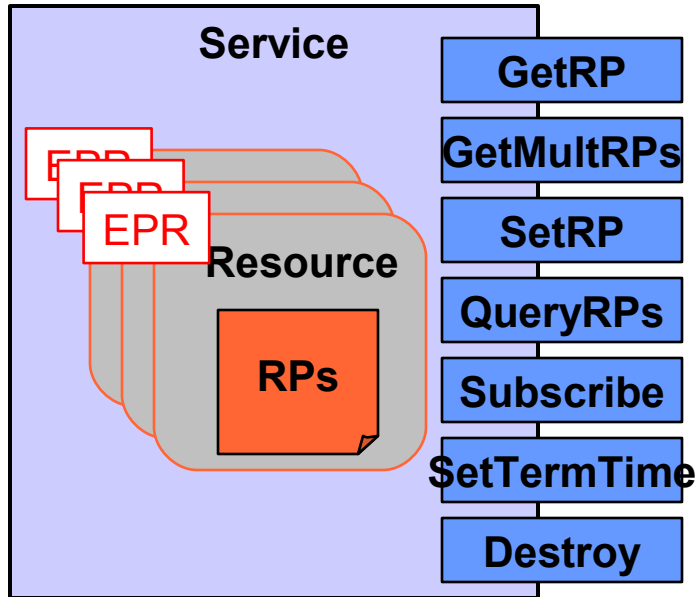
- Without state, how does client:
 - ◆ Determine what happened (success/failure)?
 - ◆ Find out how many files completed?
 - ◆ Receive updates when interesting events arise?
 - ◆ Terminate a request?
- Few useful services are truly “stateless”, but WS interfaces alone do not provide built-in support for state

FileTransferService (without WSRF)



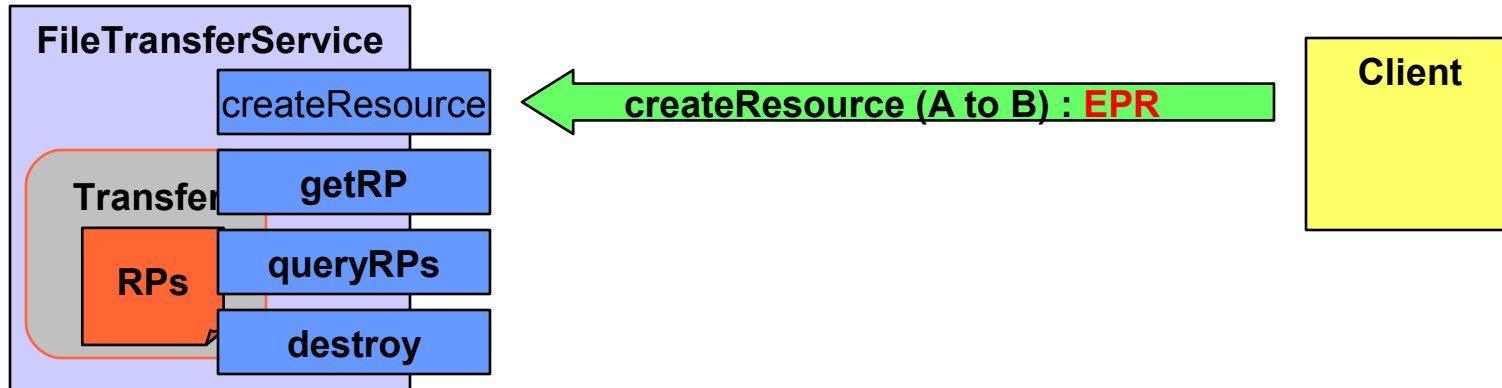
- Developer reinvents wheel for each new service
 - ◆ Custom management and identification of state: **transferID**
 - ◆ Custom operations to inspect state synchronously (**whatHappen**) and asynchronously (**tellMeWhen**)
 - ◆ Custom lifetime operation (**cancel**)

WSRF in a Nutshell



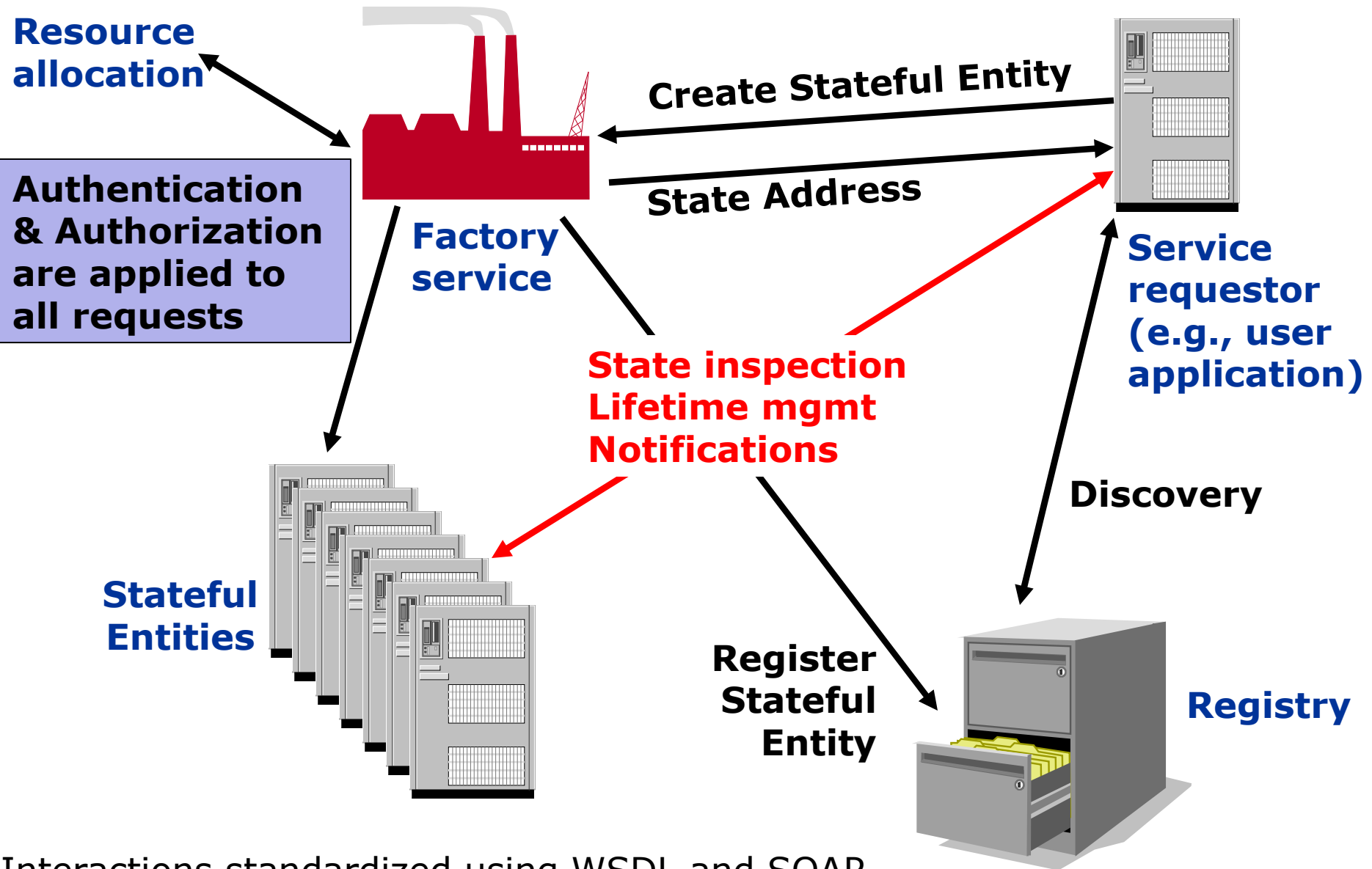
- Service
- State representation
 - ◆ Resource
 - ◆ Resource Property
- State identification
 - ◆ Endpoint Reference
- State Interfaces
 - ◆ GetRP, QueryRPs, GetMultipleRPs, SetRP
- Lifetime Interfaces
 - ◆ SetTerminationTime
 - ◆ ImmediateDestruction
- Notification Interfaces
 - ◆ Subscribe
 - ◆ Notify
- ServiceGroups

FileTransferService (w/ WSRF)

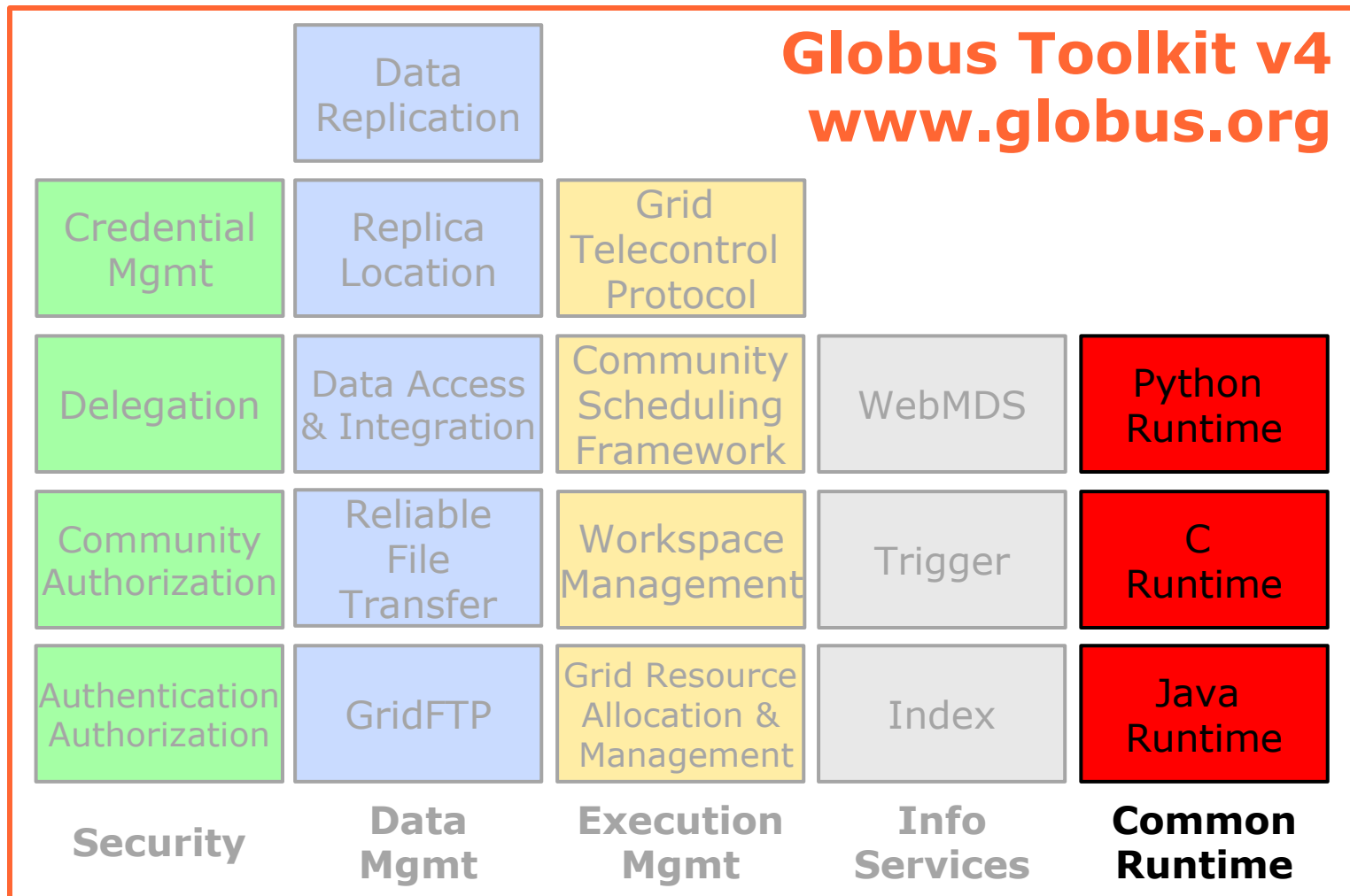


- Developer specifies custom method to createResource and leaves the rest to WSRF standards:
 - ◆ State exposed as Resource + Resource Properties and identified by Endpoint Reference (EPR)
 - ◆ State inspected by standard interfaces (GetRP, QueryRPs)
 - ◆ Lifetime management by standard interfaces (Destroy)

Modeling State in Web Services



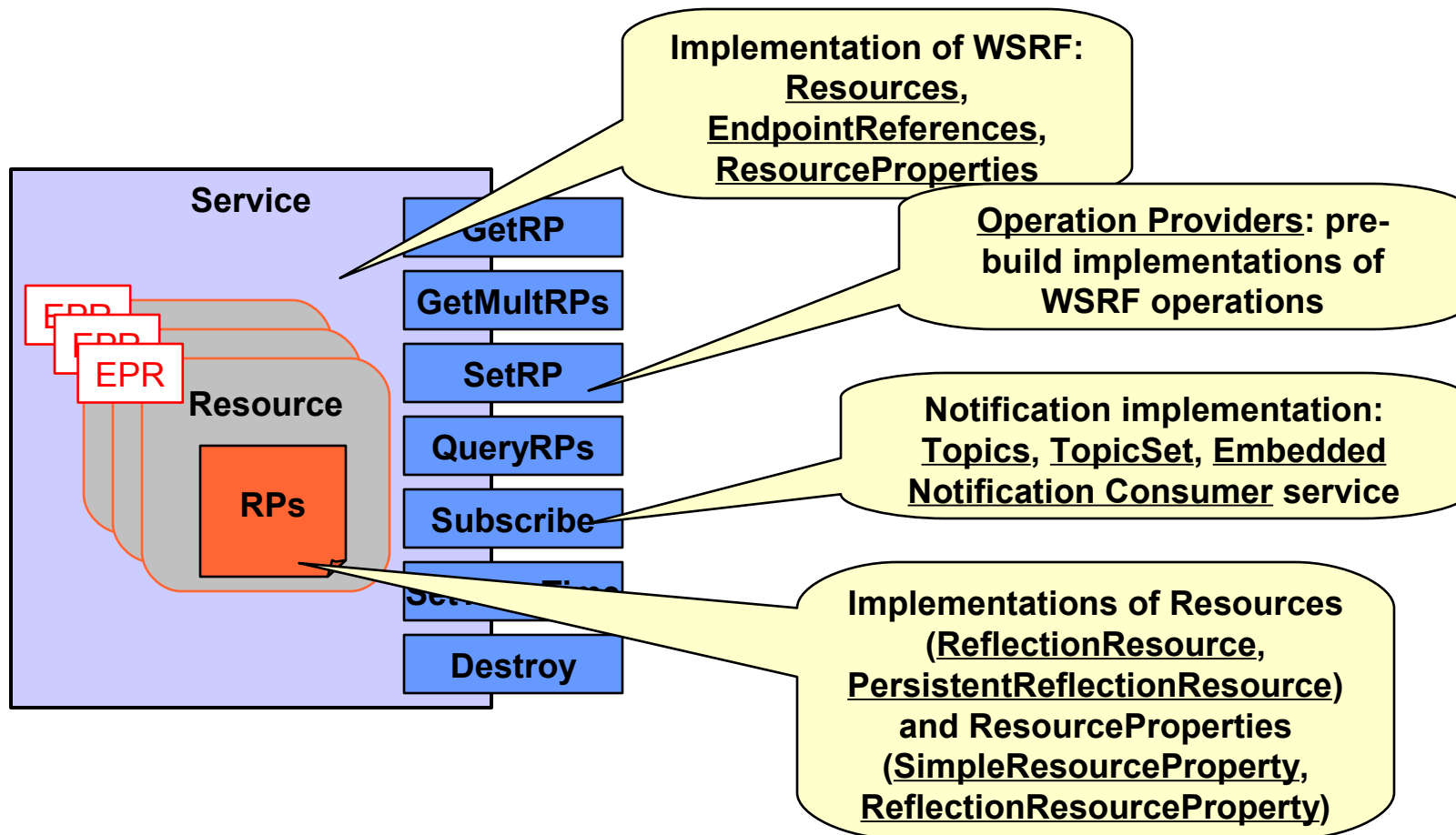
Globus Toolkit: Open Source Grid Infrastructure



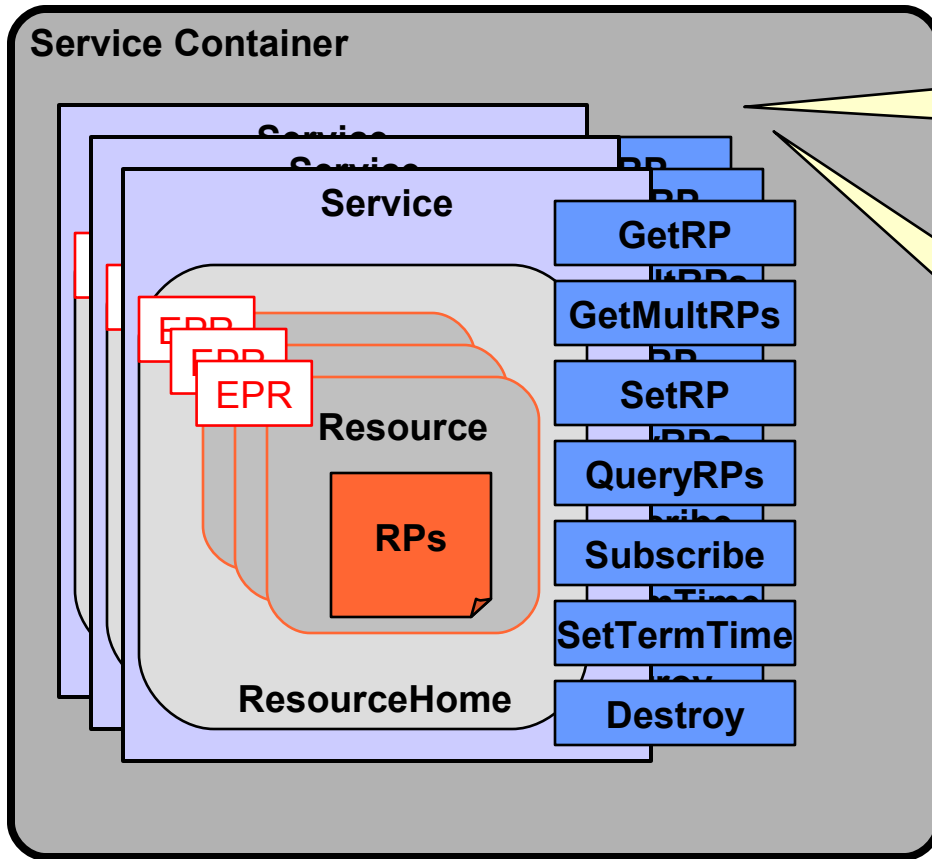
GT4 Web Services Runtime

- Supports both GT (GRAM, RFT, Delegation, etc.) & user-developed services
- Redesign to enhance scalability, modularity, performance, usability
- Leverages existing WS standards
 - ◆ WS-I Basic Profile: WSDL, SOAP, etc.
 - ◆ WS-Security, WS-Addressing
- Adds support for emerging WS standards
 - ◆ WS-Resource Framework, WS-Notification
- Java, Python, & C hosting environments
 - ◆ Java is standard Apache

GT4 WS Core in a Nutshell



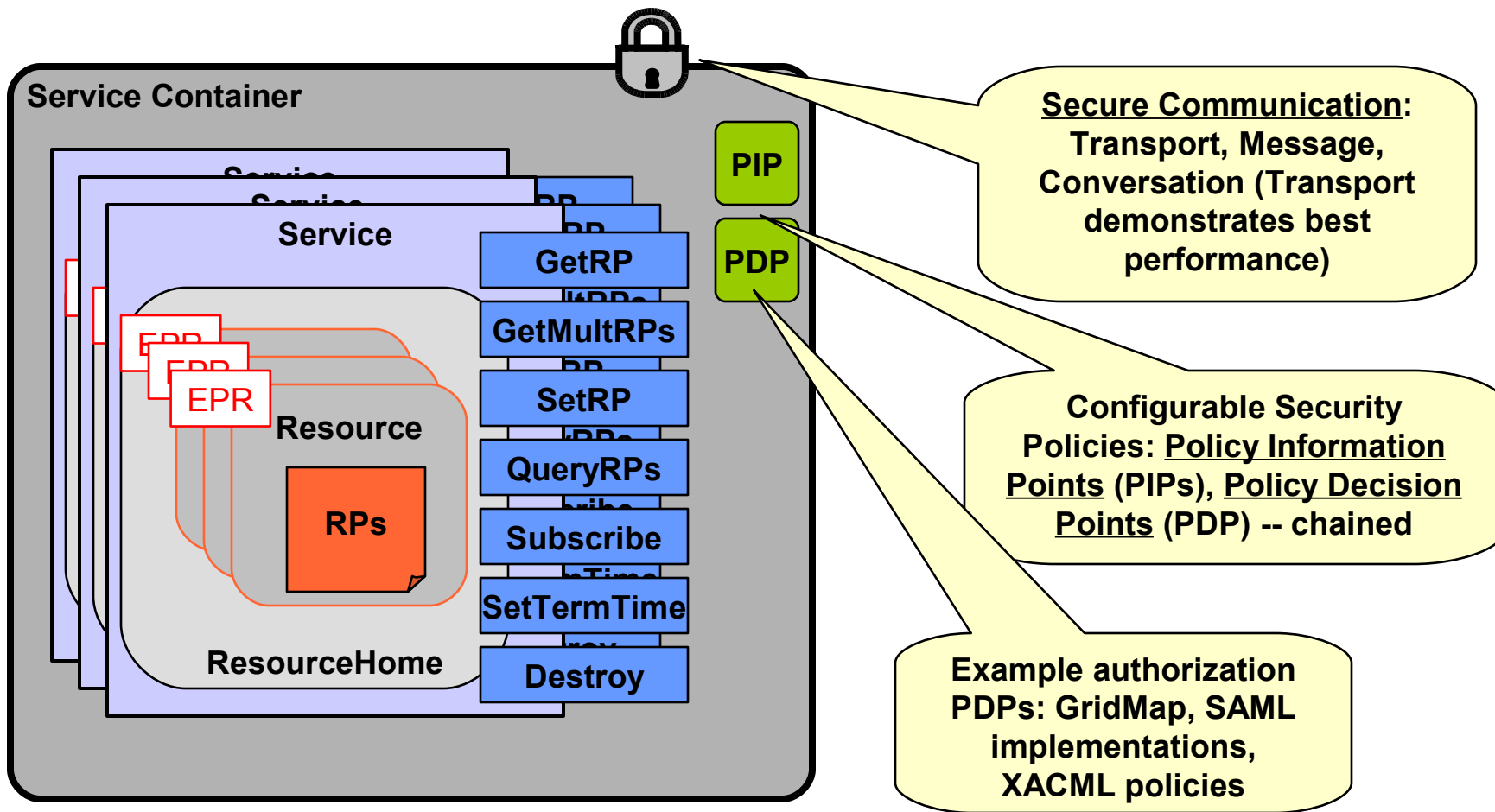
GT4 WS Core in a Nutshell



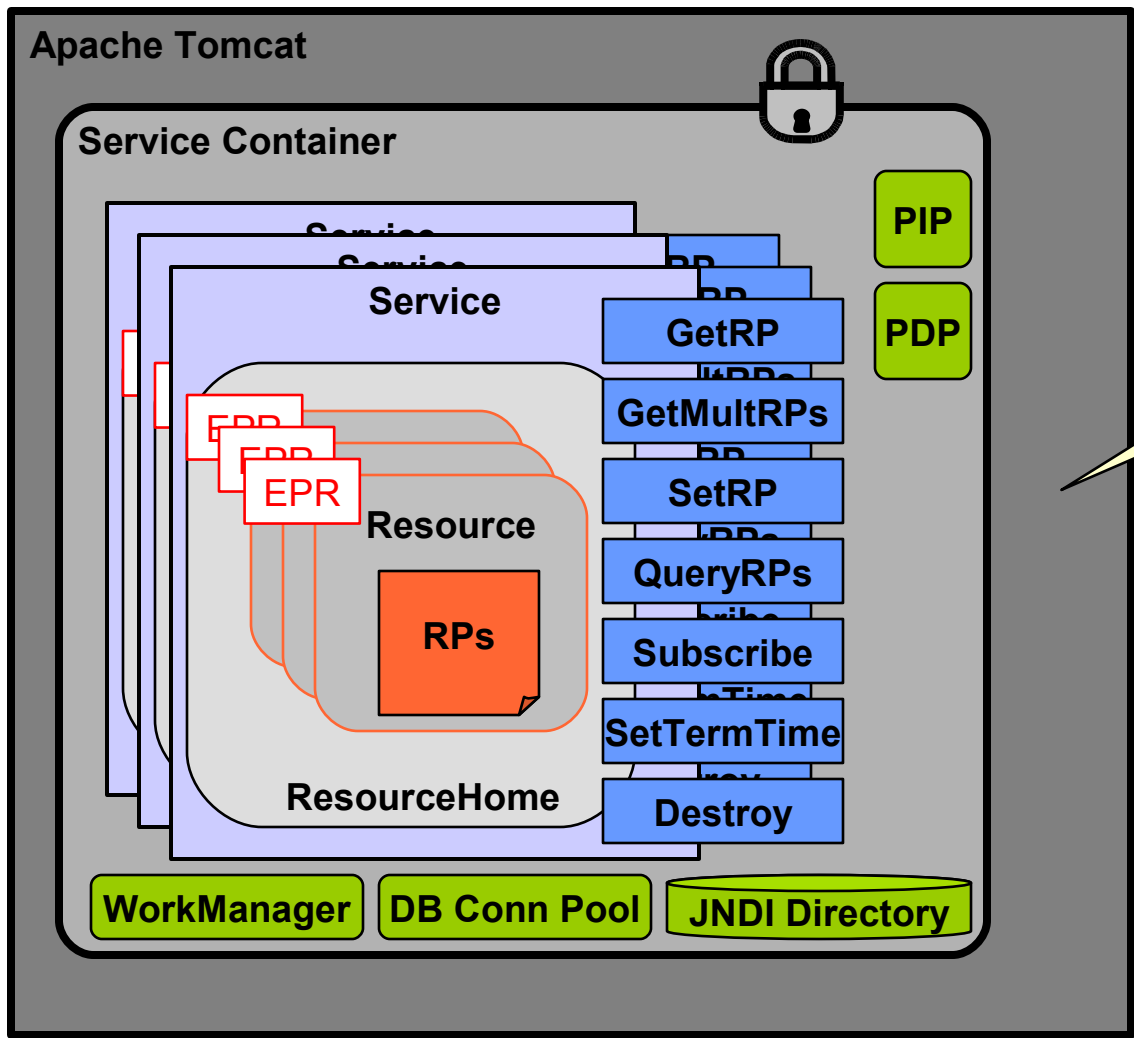
Service Container: host multiple services in container; one JVM process

...more details: based on AXIS service container, processes SOAP messages, ResourceContext extension.

GT4 WS Core in a Nutshell

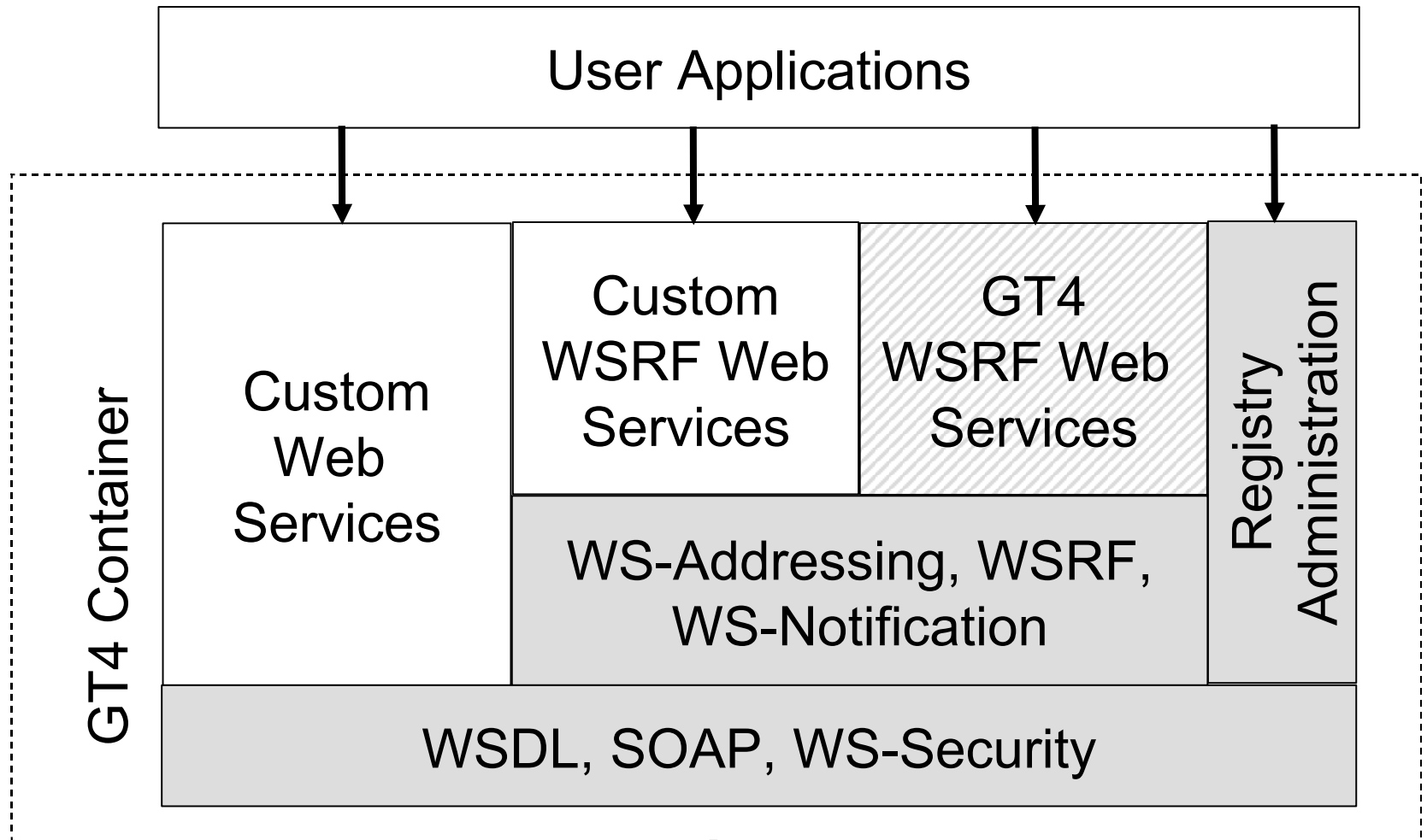


GT4 WS Core in a Nutshell



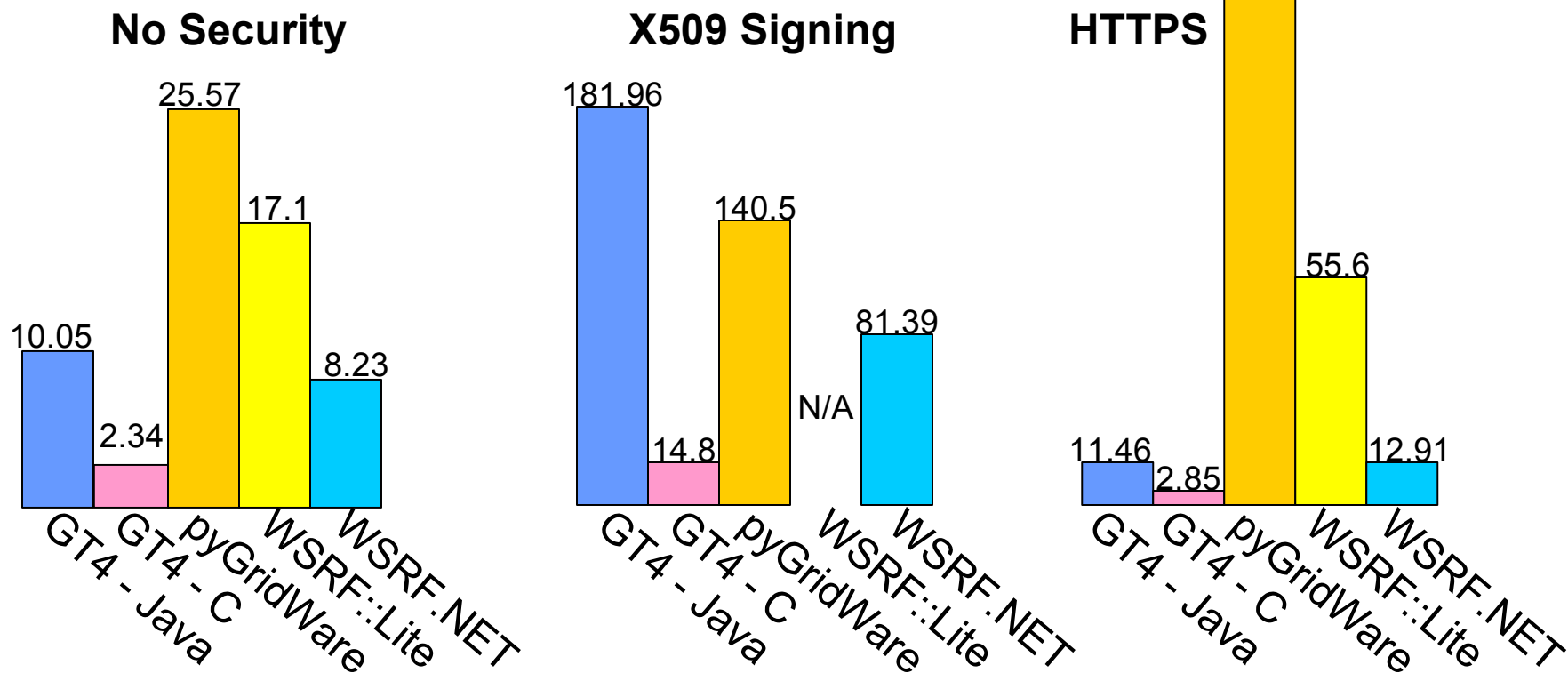
Deploy Service Container "standalone" or within Apache Tomcat

GT4 Web Services Runtime



GetRP Test

Distributed client and service on same LAN
(times in milliseconds)





GT4 WS Core Performance

(1) Message-level security (times in milliseconds)

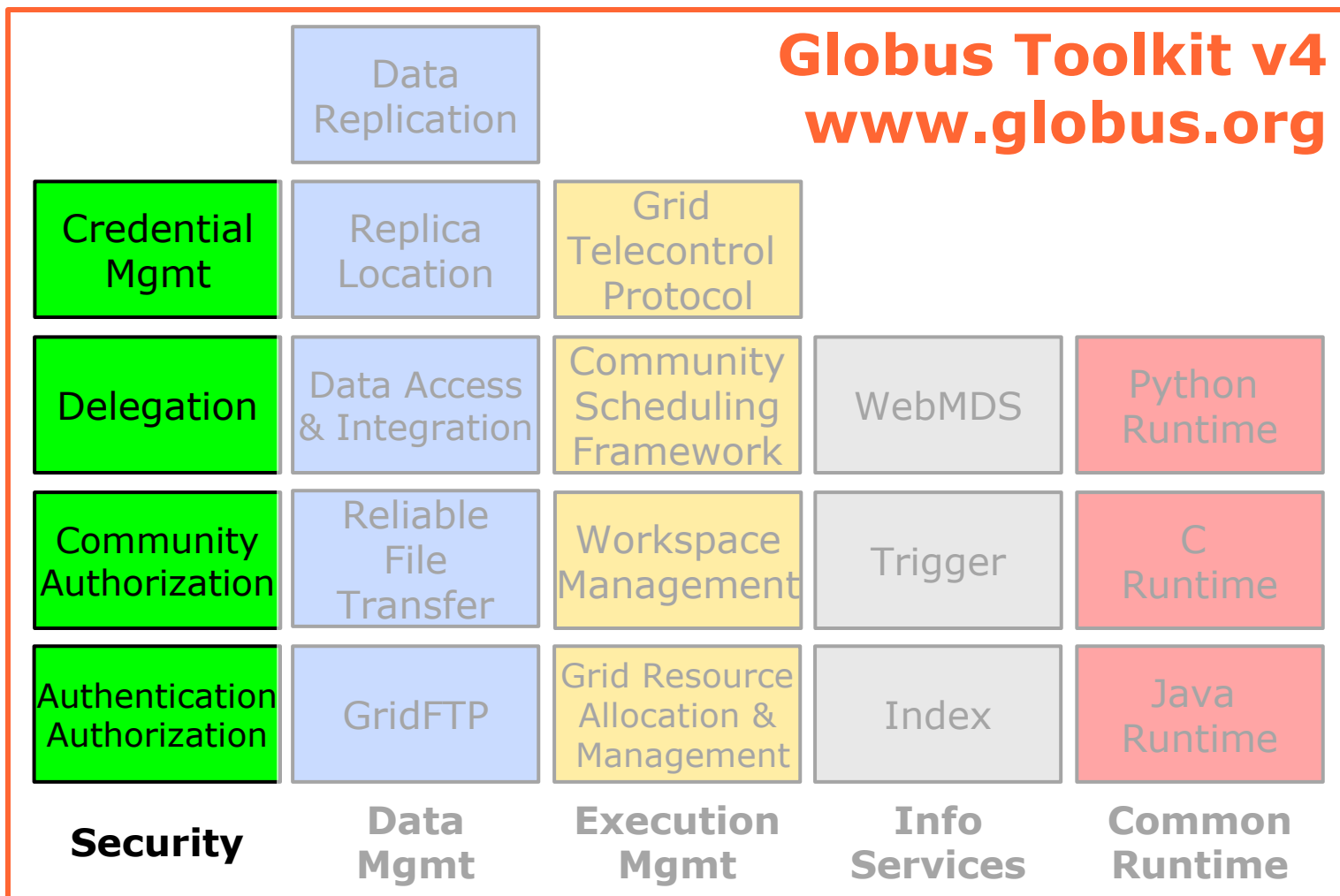
	GT4 Java	GT4 C	GT4 Python	WSRF.NET
GetRP	181.96	14.77	140.50	81.39
SetRP	182.04	14.99	142.21	82.48
CreateR	188.46	14.98	132.26	96.22
DestroyR	182.03	15.76	136.12	86.89
Notify	219.51	N/A	244.93	101.57

(2) Transport-level security (times in milliseconds)

	GT4 Java	GT4 C	GT4 Python	WSRF.NET
getRP	11.46	2.85	149.67	12.91
setRP	11.47	2.86	150.79	12.3
createR	18.00	2.82	132.60	20.84
destroyR	14.92	2.71	149.21	16.05
Notify	29.26	9.67	169.07	45.0

“WSRF/WSNs Compared,” HPDC 2005.

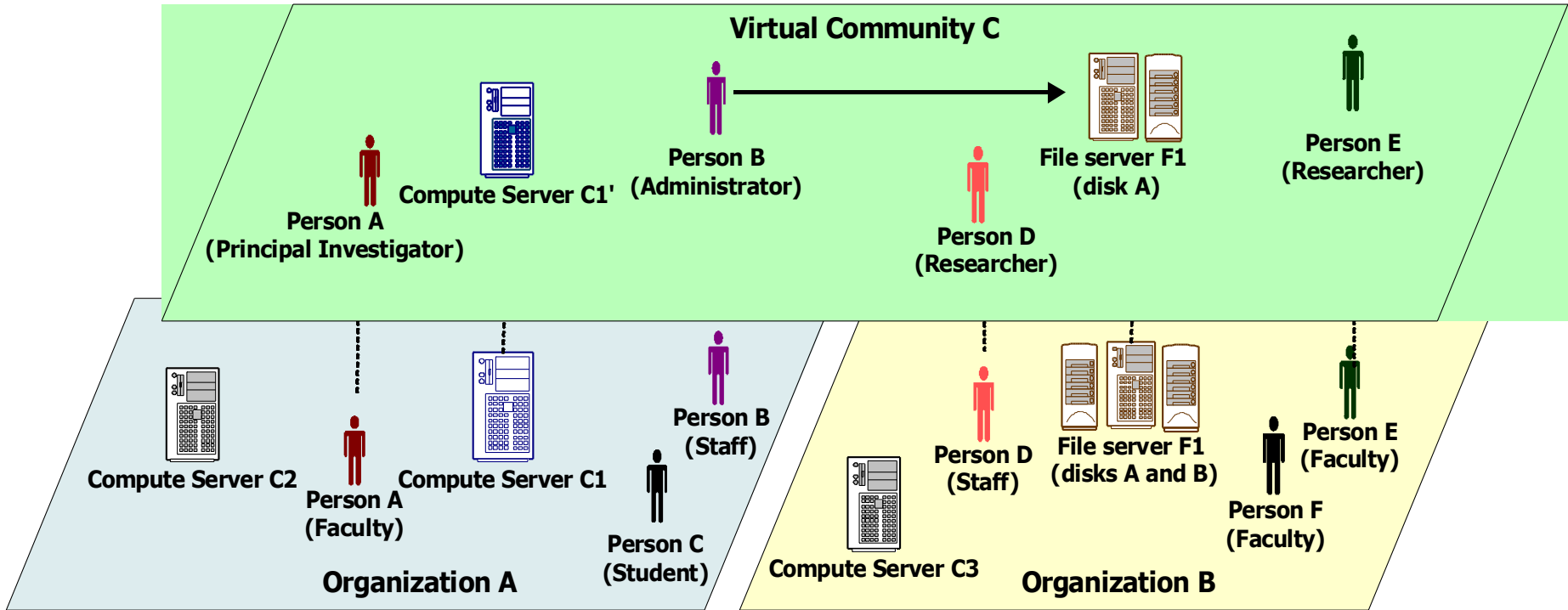
Globus Toolkit: Open Source Grid Infrastructure



Globus Security

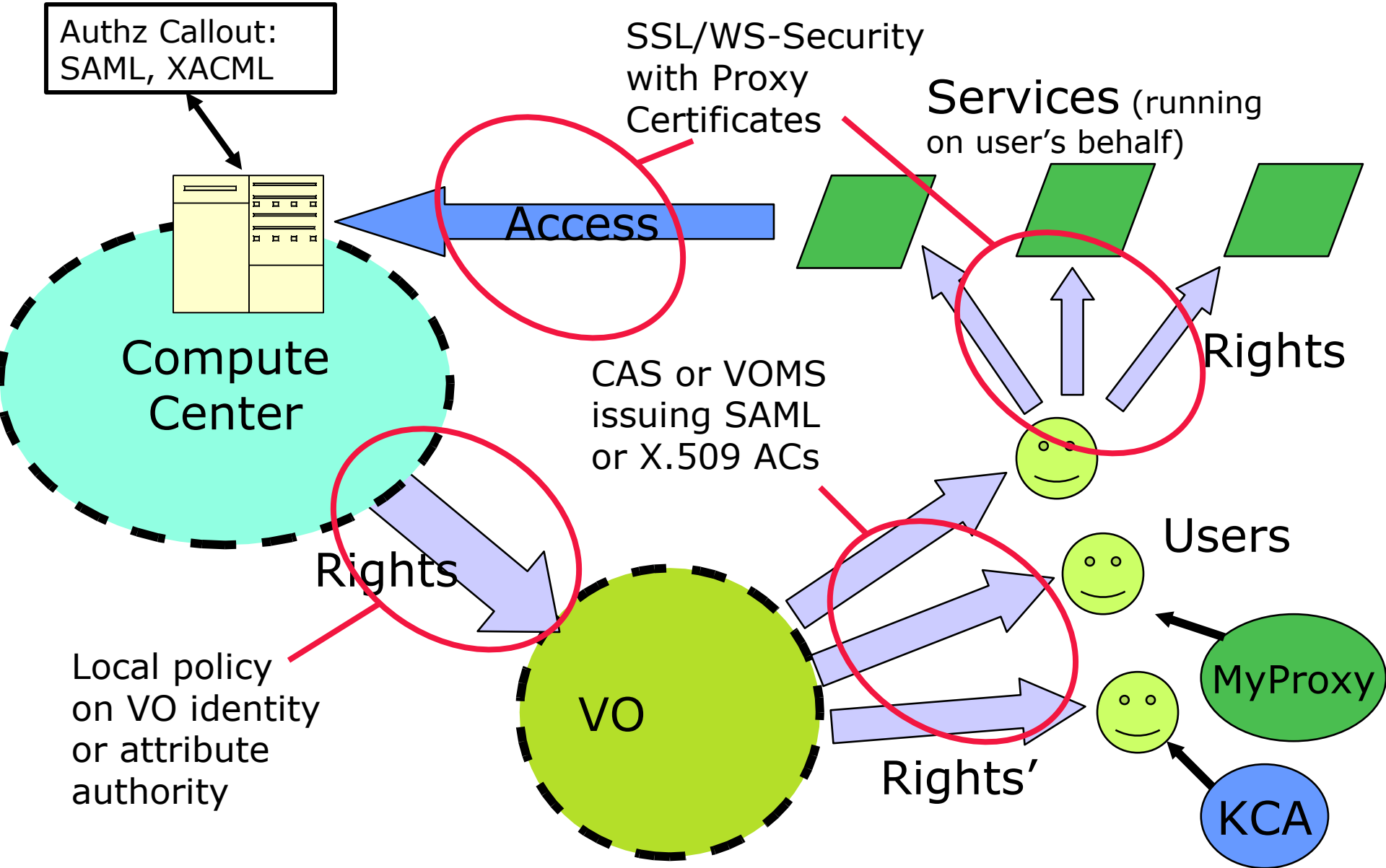
- Control access to shared services
 - ◆ Address autonomous management, e.g., different policy in different work-groups
- Support multi-user collaborations
 - ◆ Federate through mutually trusted services
 - ◆ Local policy authorities rule
- Allow users and application communities to set up dynamic trust domains
 - ◆ Personal/VO collection of resources working together based on trust of user/VO

Virtual Organization (VO) Concept



- VO for each application or workload
- Carve out and configure resources for a particular use and set of users

GT4 Security



GT4 Security

- Public-key-based authentication
- Extensible authorization framework based on Web services standards
 - ◆ SAML-based authorization callout
 - As specified in GGF OGSA-Authz WG
 - ◆ Integrated policy decision engine
 - XACML policy language, per-operation policies, pluggable
- Credential management service
 - ◆ MyProxy (One time password support)
- Community Authorization Service
- Standalone delegation service

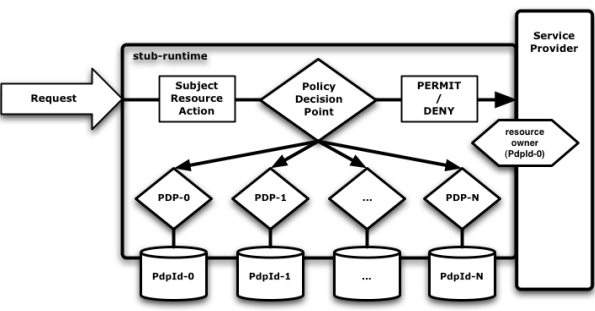
GT4's Use of Security Standards

	Message-level Security w/X.509 Credentials	Message-level Security w/Username and Passwords	Transport-level Security w/X.509 Credentials
Authorization	SAML and grid-mapfile	grid-mapfile	SAML and grid-mapfile
Delegation	X.509 Proxy Certificates/ WS-Trust		X.509 Proxy Certificates/ WS-Trust
Authentication	X.509 End Entity Certificates	Username/ Password	X.509 End Entity Certificates
Message Protection	WS-Security WS-SecureConversation	WS-Security	TLS
Message format	SOAP	SOAP	SOAP
	Supported, but slow	Supported, but insecure	Fastest, so default

GT-XACML Integration

- eXtensible Access Control Markup Language
 - ◆ OASIS standard, open source implementations
- XACML: sophisticated policy language
- Globus Toolkit ships with XACML runtime
 - ◆ Included in every client and server built on GT
 - ◆ Turned-on through configuration
- ... that can be called transparently from runtime and/or explicitly from application ...
- ... and we use the XACML-“model” for our Authz Processing Framework

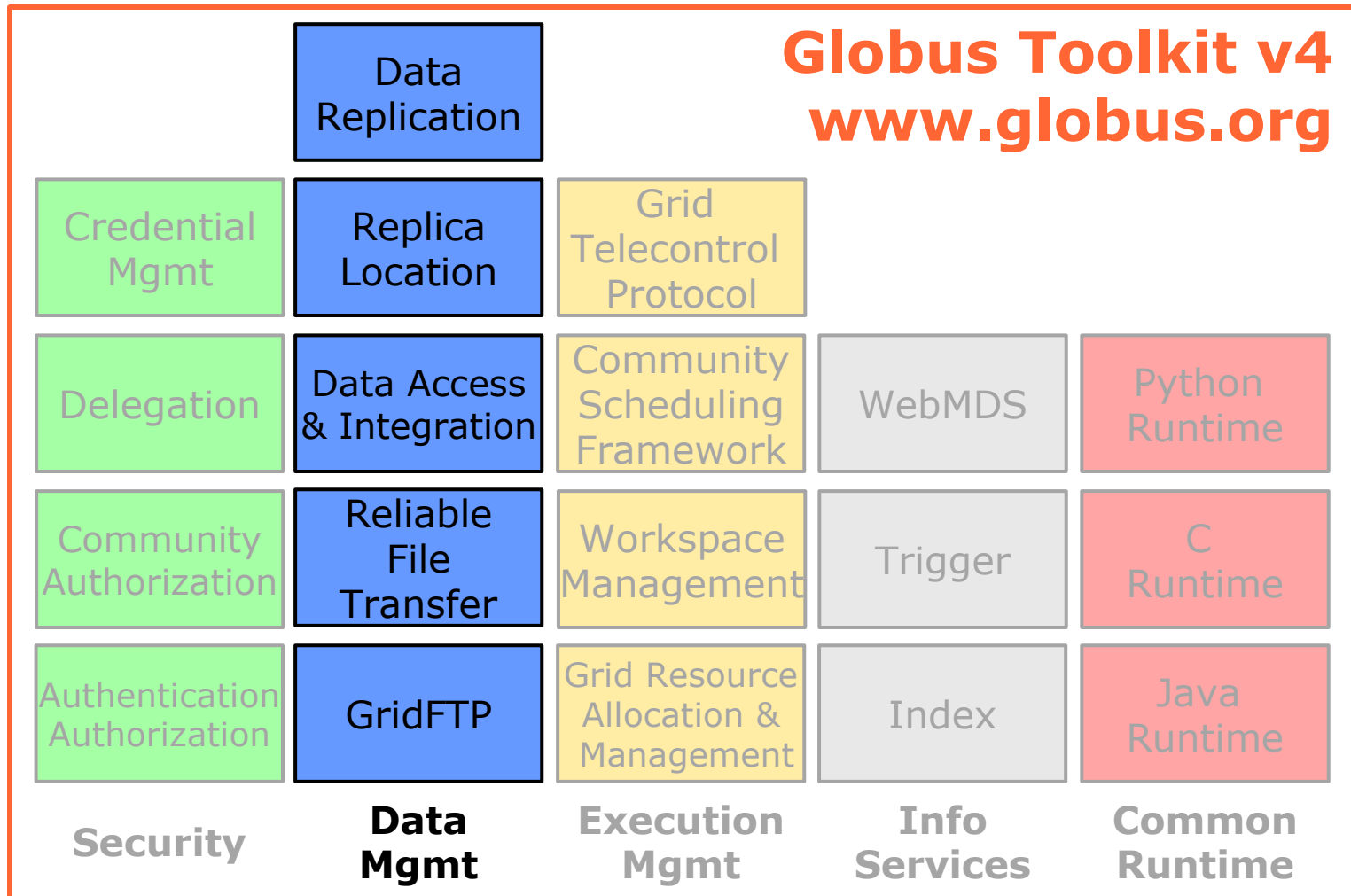
GT Authorization Framework



Other Security Services Include ...

- **MyProxy**
 - ◆ Simplified credential management
 - ◆ Web portal integration
 - ◆ Single-sign-on support
- **KCA & kx.509**
 - ◆ Bridging into/out-of Kerberos domains
- **SimpleCA**
 - ◆ Online credential generation
- **PERMIS**
 - ◆ Authorization service callout

Globus Toolkit: Open Source Grid Infrastructure

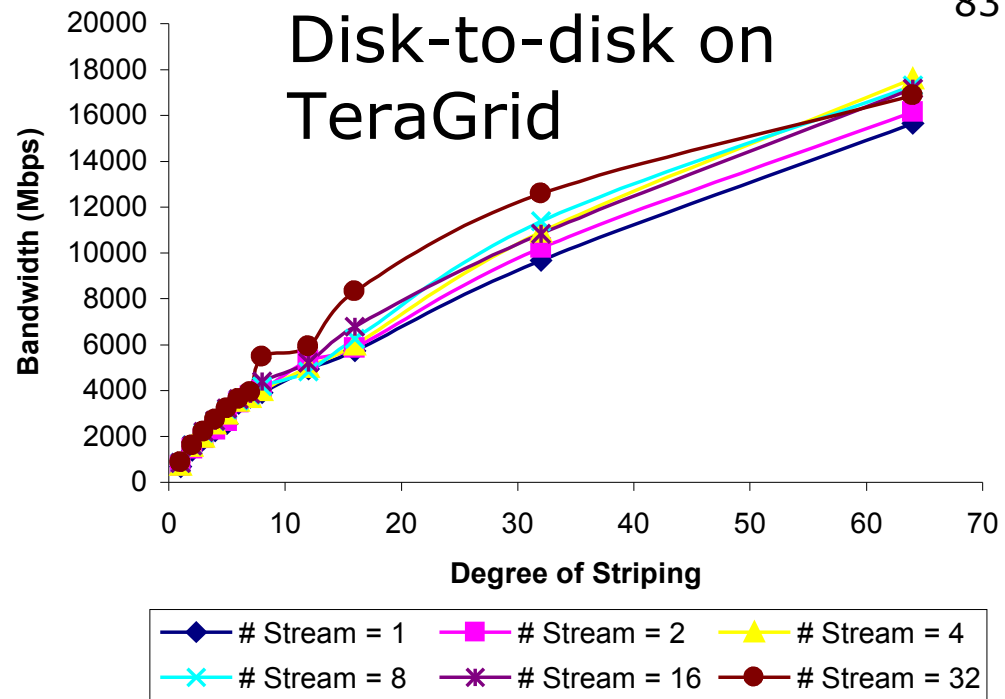


GT4 Data Management

- **Stage/move** large data to/from nodes
 - ◆ GridFTP, Reliable File Transfer (RFT)
 - ◆ Alone, and integrated with GRAM
- **Locate** data of interest
 - ◆ Replica Location Service (RLS)
- **Replicate** data for performance/reliability
 - ◆ Distributed Replication Service (DRS)
- Provide **access** to diverse data sources
 - ◆ File systems, parallel file systems, hierarchical storage: GridFTP
 - ◆ Databases: OGSA DAI

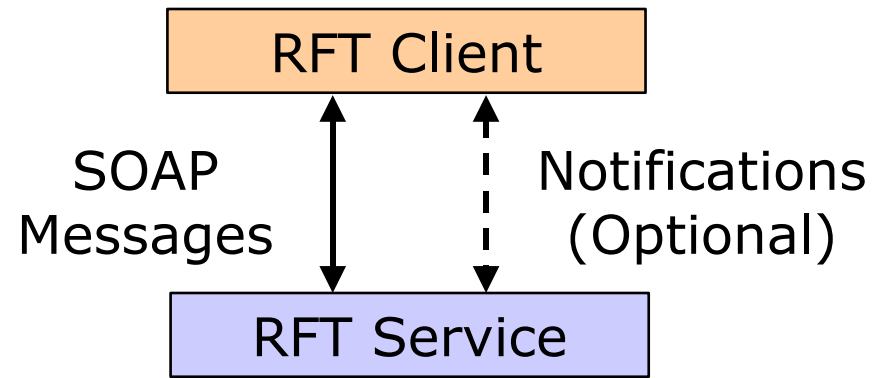
GridFTP in GT4

- 100% Globus code
 - ◆ No licensing issues
 - ◆ Stable, extensible
- IPv6 Support
- XIO for different transports
- Striping → multi-Gb/sec wide area transport
 - ◆ 27 Gbit/s on 30 Gbit/s link
- Pluggable
 - ◆ Front-end: e.g., future WS control channel
 - ◆ Back-end: e.g., HPSS, cluster file systems
 - ◆ Transfer: e.g., UDP, NetBLT transport



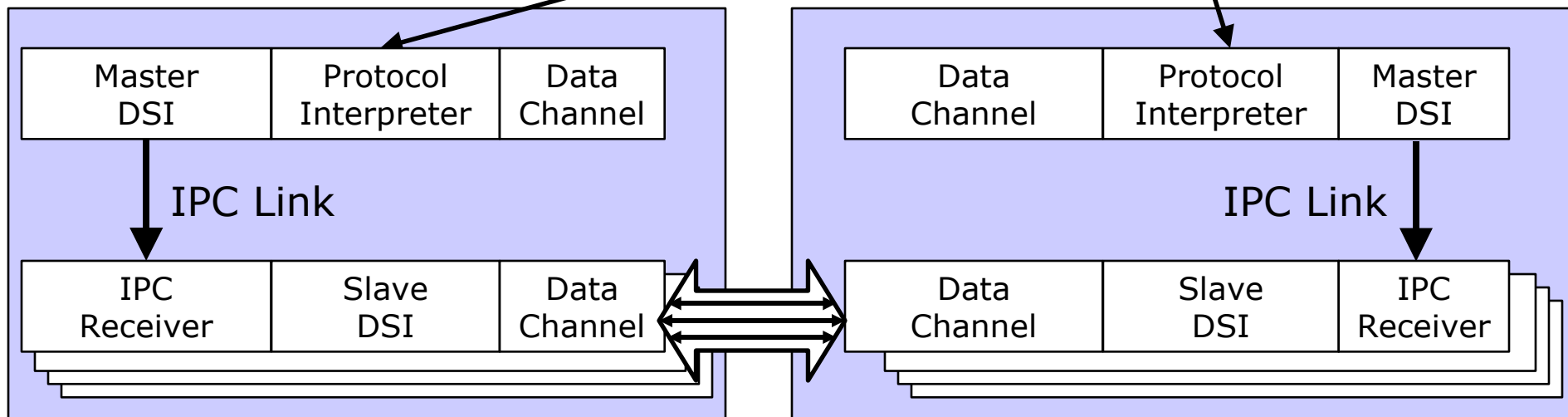
Reliable File Transfer: Third Party Transfer

- Fire-and-forget transfer
- Web services interface
- Many files & directories
- Integrated failure recovery
- Has transferred 900K files



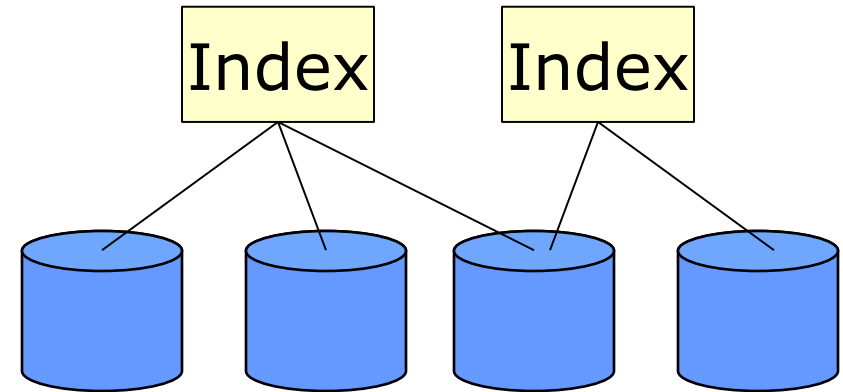
GridFTP Server

GridFTP Server



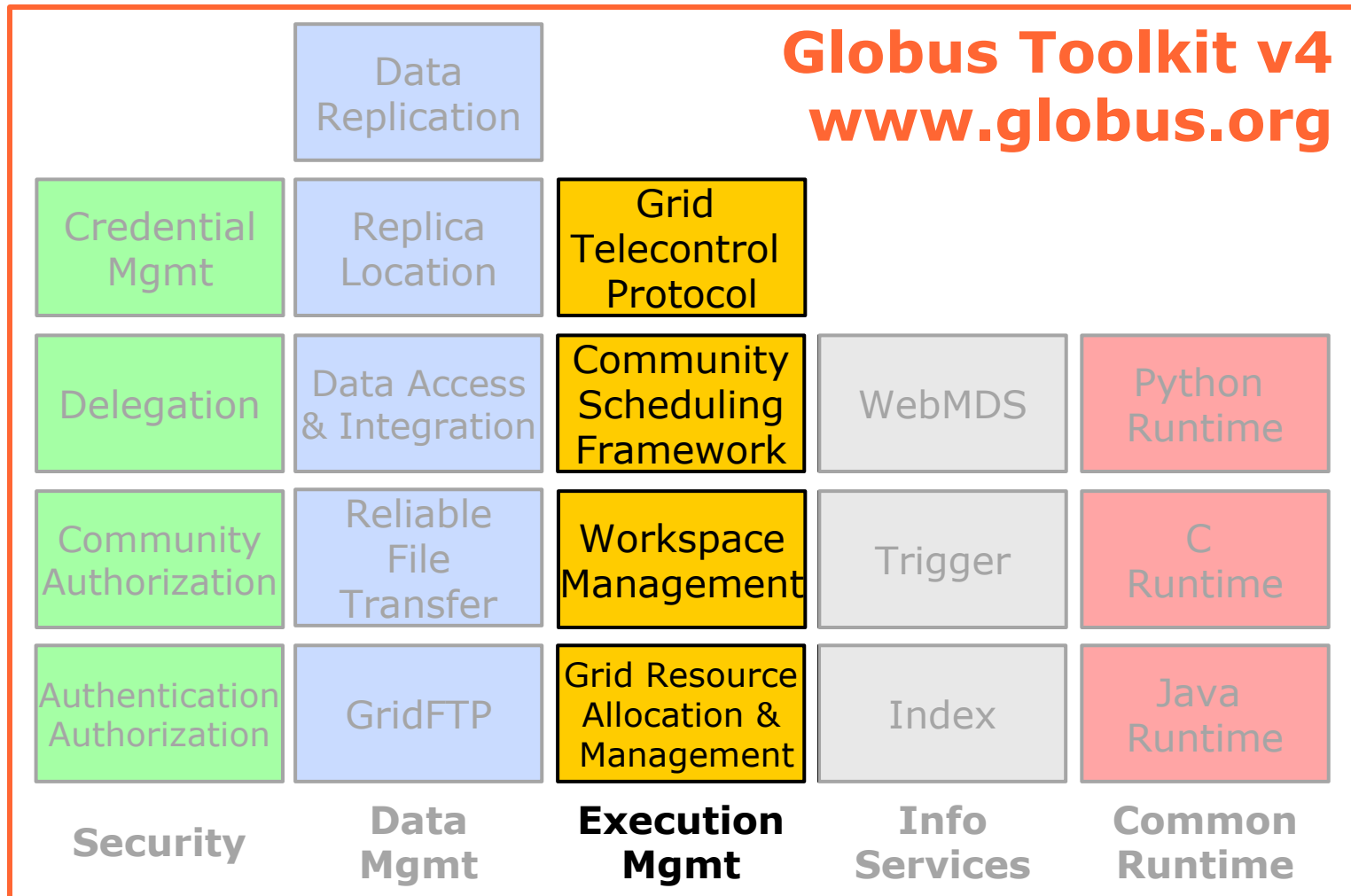
Replica Location Service

- Identify location of files via logical to physical name map
- Distributed indexing of names, fault tolerant update protocols
- GT4 version scalable & stable
- Managing ~40 million files across ~10 sites



Local DB	Update send (secs)	Bloom filter (secs)	Bloom filter (bits)
10K	<1	2	1 M
1 M	2	24	10 M
5 M	7	175	50 M

Globus Toolkit: Open Source Grid Infrastructure



Execution Management (GRAM)

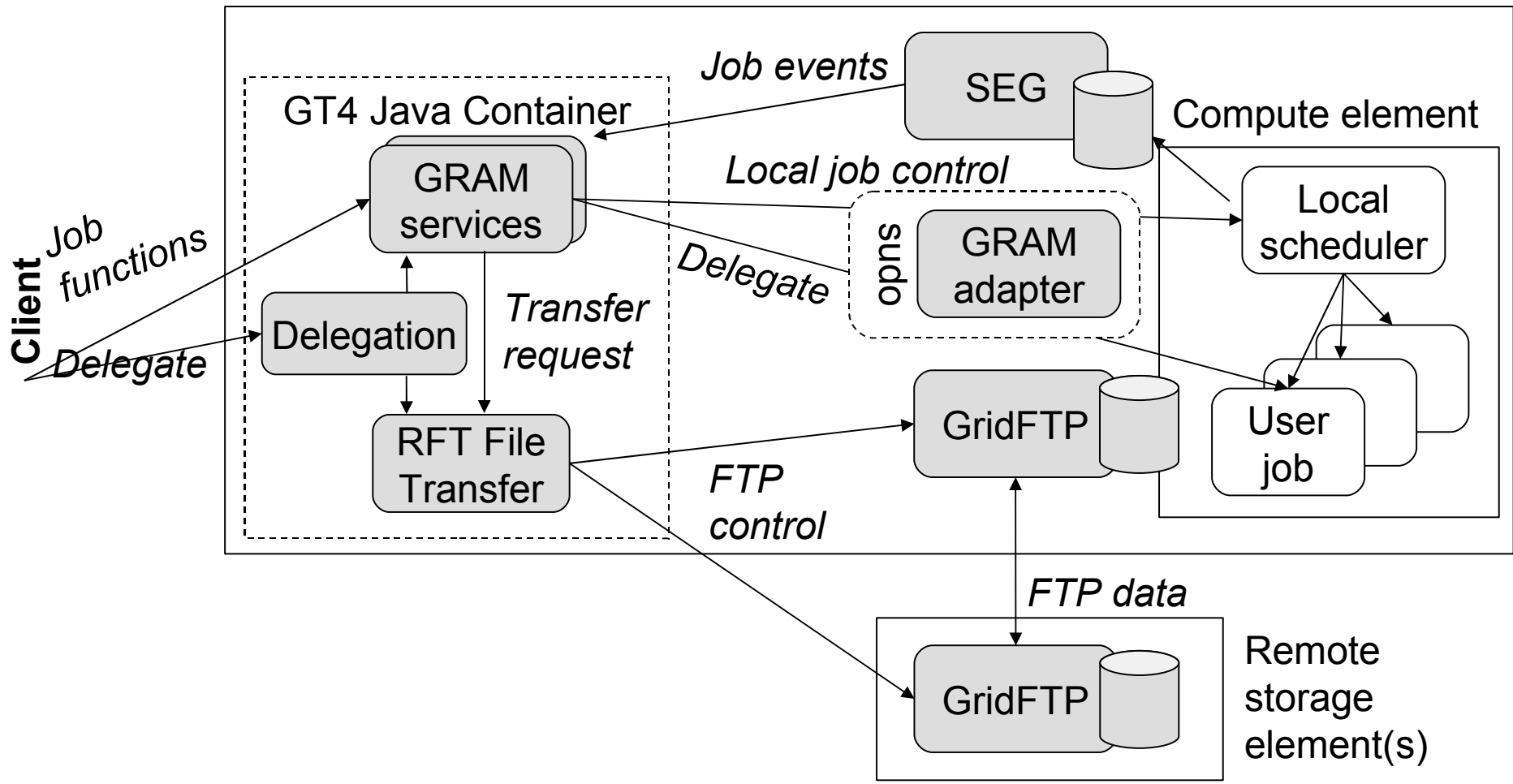
- Common WS interface to schedulers
 - ◆ Unix, Condor, LSF, PBS, SGE, ...
- More generally: interface for process execution management
 - ◆ Lay down execution environment
 - ◆ Stage data
 - ◆ Monitor & manage lifecycle
 - ◆ Kill it, clean up
- A basis for application-driven provisioning

GT4 WS GRAM

- 2nd-generation WS implementation optimized for performance, flexibility, stability, scalability
- Streamlined critical path
 - ◆ Use only what you need
- Flexible credential management
 - ◆ Credential cache & delegation service
- GridFTP & RFT used for data operations
 - ◆ Data staging & streaming output
 - ◆ Eliminates redundant GASS code

GT4 WS GRAM Architecture

Service host(s) and compute element(s)

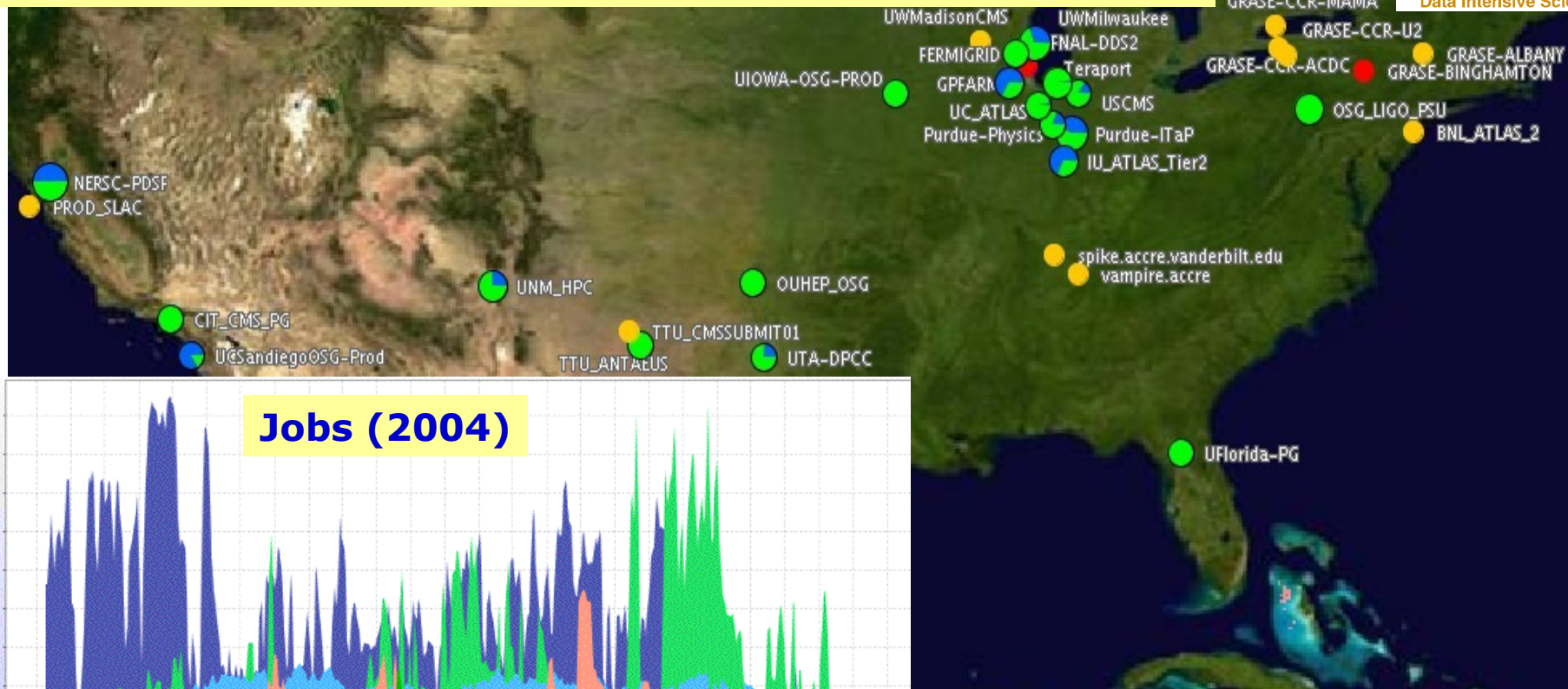
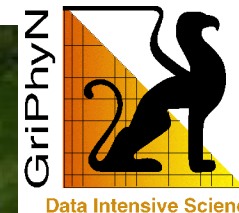
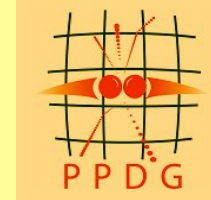


WS GRAM Performance

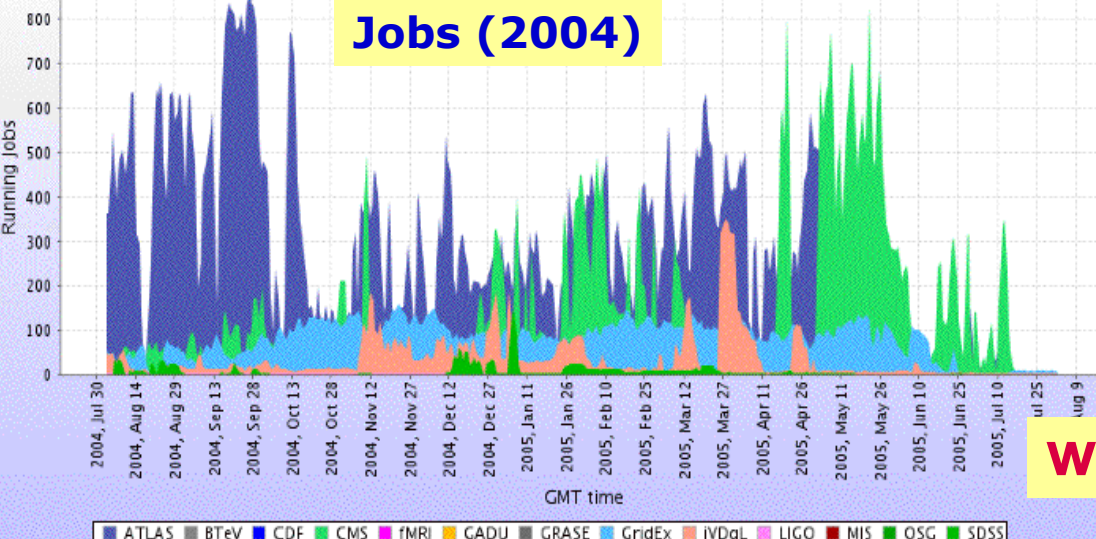
- Time to submit a basic GRAM job
 - ◆ Pre-WS GRAM: < 1 second
 - ◆ WS GRAM: 2 seconds
- Concurrent jobs
 - ◆ Pre-WS GRAM: 300 jobs
 - ◆ WS GRAM: 32,000 jobs
- Various studies are underway to test latest software

Open Science Grid

- 50 sites (15,000 CPUs) & growing
- 400 to >1000 concurrent jobs
- Many applications + CS experiments; includes long-running production operations
- Up since October 2003; few FTEs central ops



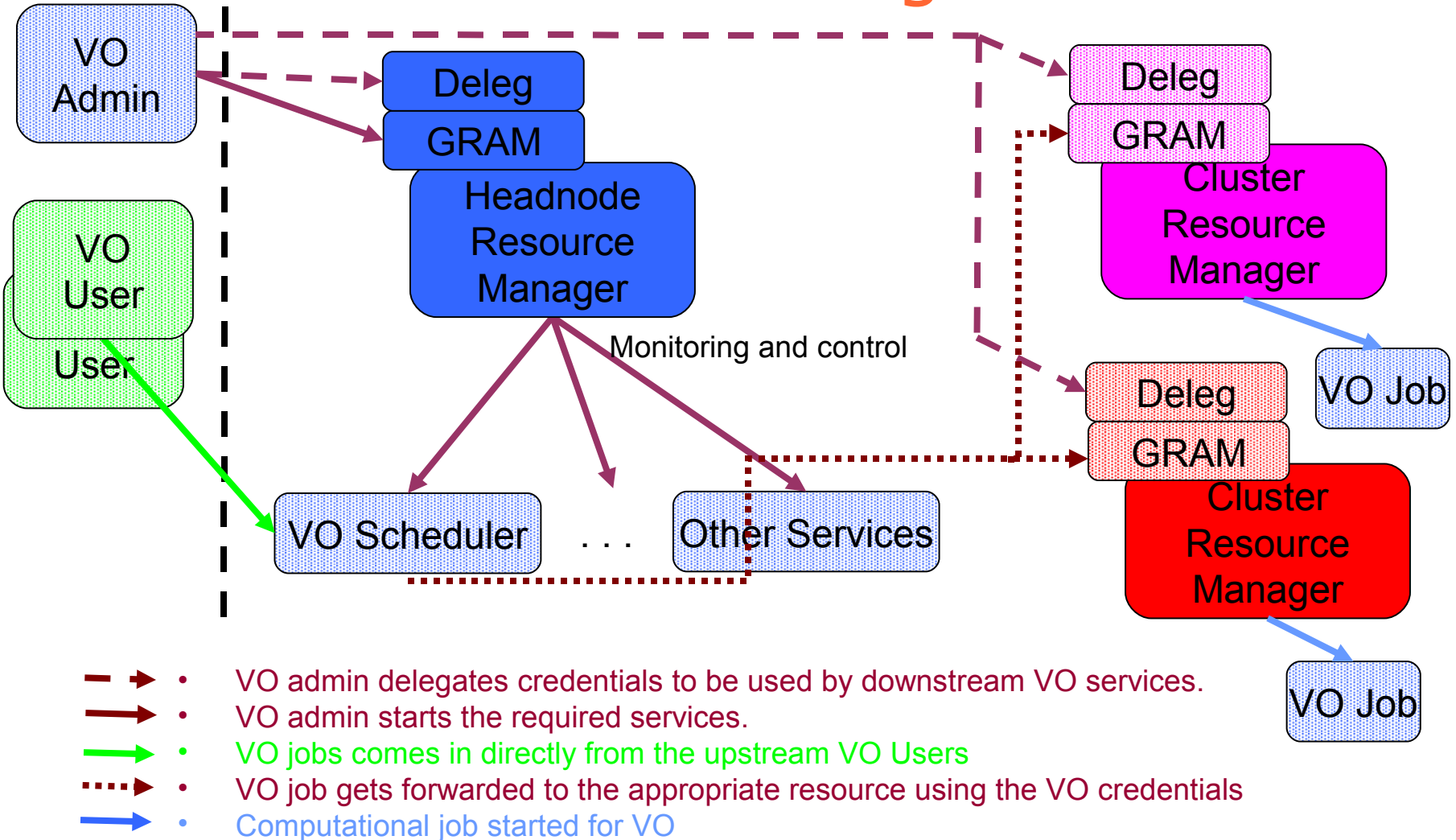
Jobs (2004)



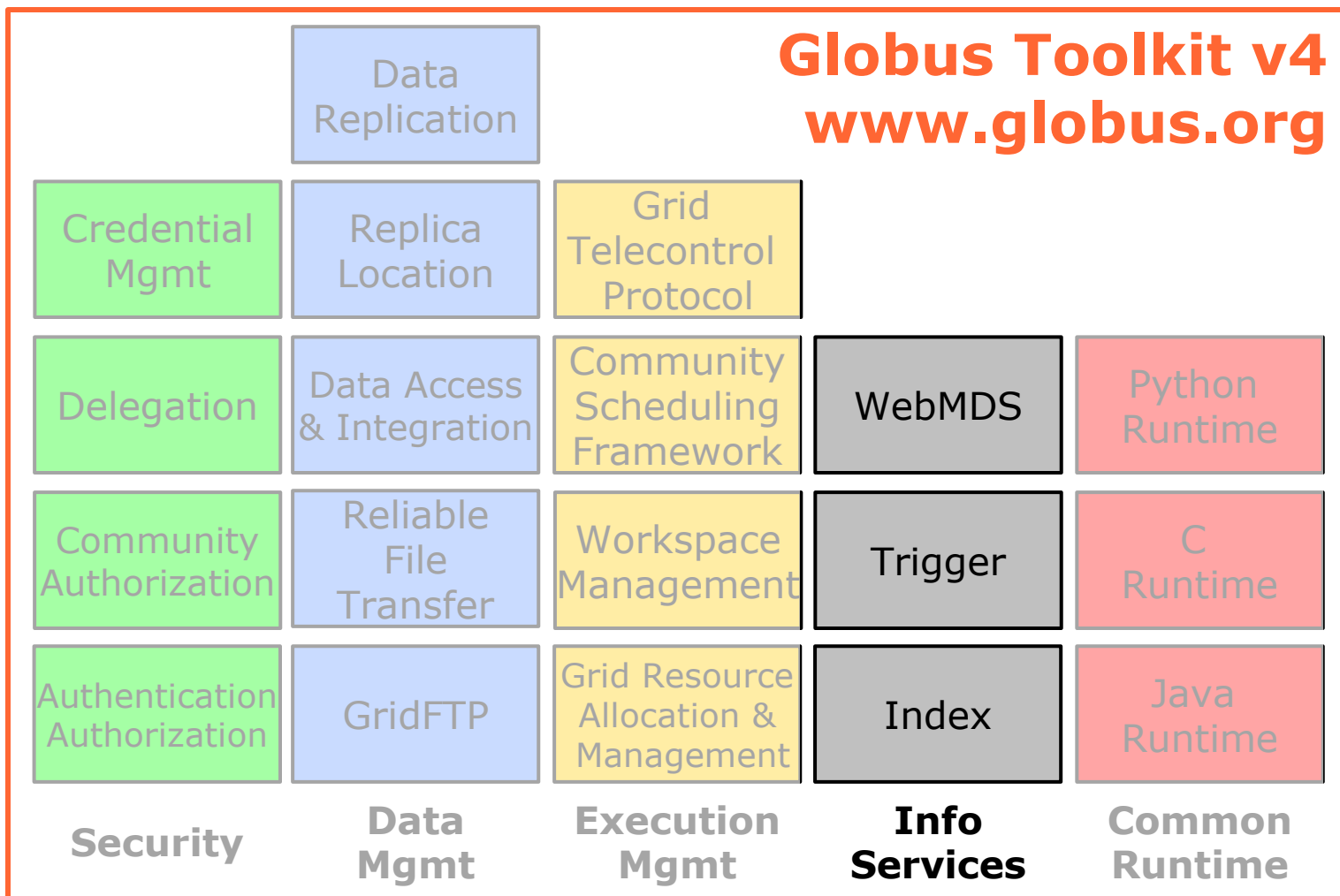
www.opensciencegrid.org

Embedded Resource Management

Client-side



Globus Toolkit: Open Source Grid Infrastructure



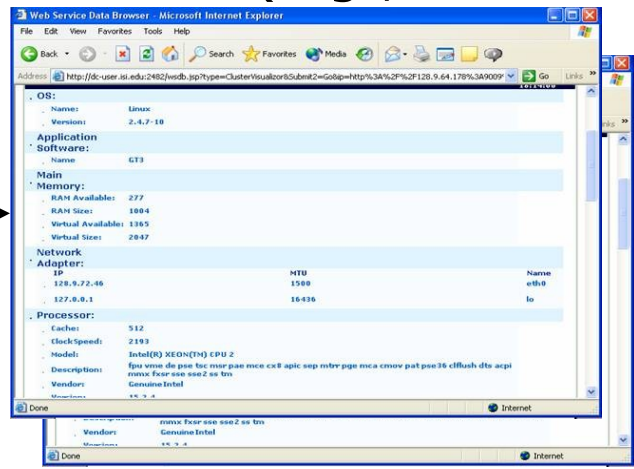
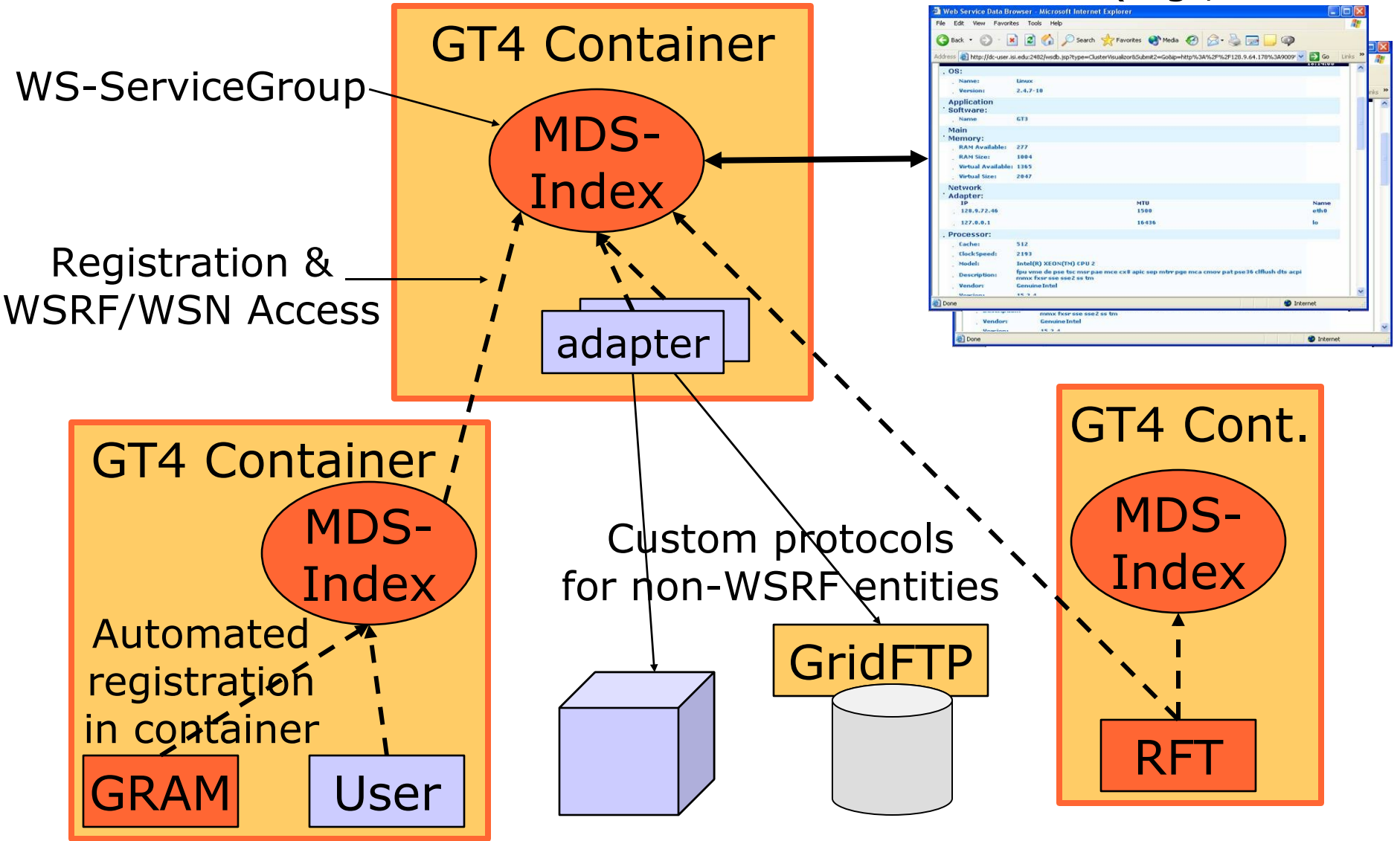
Monitoring and Discovery

- “Every service should be monitorable and discoverable using common mechanisms”
 - ◆ WSRF/WSN provides those mechanisms
- A common **aggregator** framework for collecting information from services, thus:
 - ◆ MDS-Index: Xpath queries, with caching
 - ◆ MDS-Trigger: perform action on condition
 - ◆ (MDS-Archiver: Xpath on historical data)
- Deep integration with Globus containers & services: every GT4 service is discoverable
 - ◆ GRAM, RFT, GridFTP, CAS, ...

GT4

Monitoring & Discovery

Clients
(e.g., WebMDS)



Information Providers

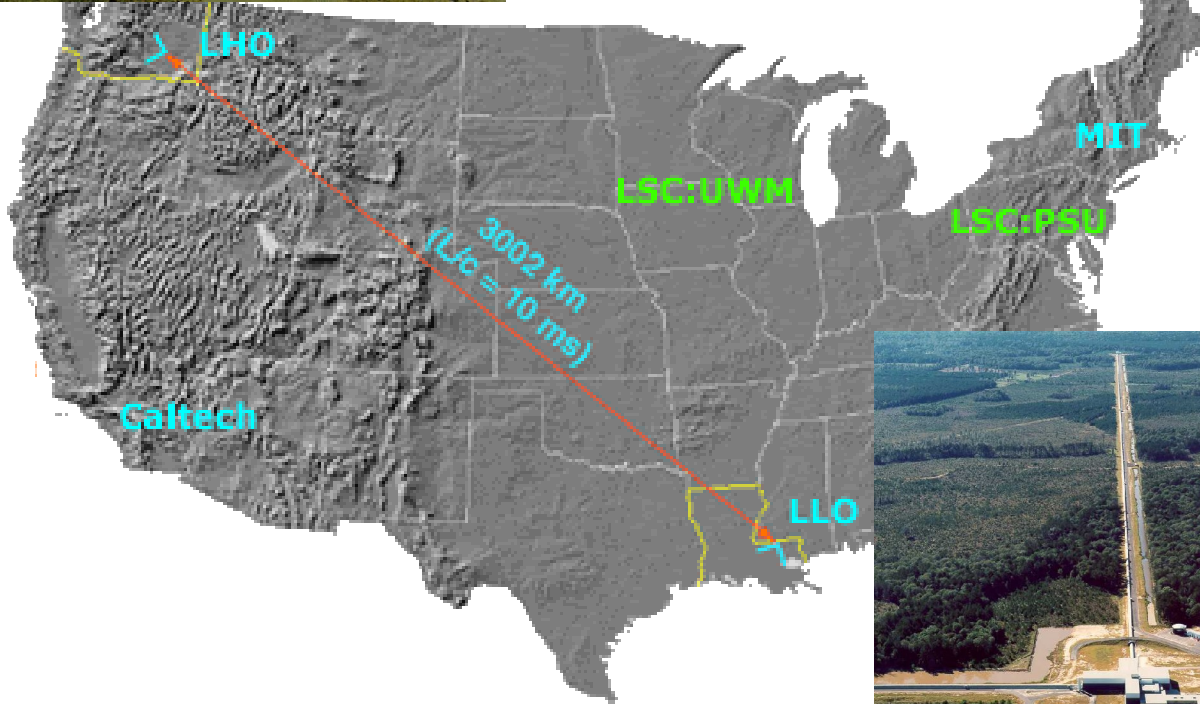
- **GT4 information providers** collect information from some system and make it accessible as WSRF resource properties
- Growing number of information providers
 - ◆ Nagios, SGE, LSF, PBS
- Many opportunities to build additional ones
 - ◆ E.g., network monitoring, storage systems, various sensors

Putting it Together: Data Replication Service

- New capabilities can be build by:
 - ◆ Creating new services using GT4 containers
 - ◆ Composing/combining existing service

Reliable Wide Area Data Replication

LIGO Gravitational Wave Observatory



Replicating >1 Terabyte/day to 8 sites
 >30 million replicas so far
 MTBF = 1 month

www.globus.org/solutions



The Data Replication Service

- Tech Preview in GT4.0
- Based on the publication component of the Lightweight Data Replicator system
 - ◆ Developed by Scott Koranda from U. Wisconsin at Milwaukee
- Function to locally replicate a set of files
 - ◆ User identifies a desired files
 - ◆ DRS uses RLS to discover file locations
 - ◆ Use RFT to create local replicas
 - ◆ Registers new replicas in RLS

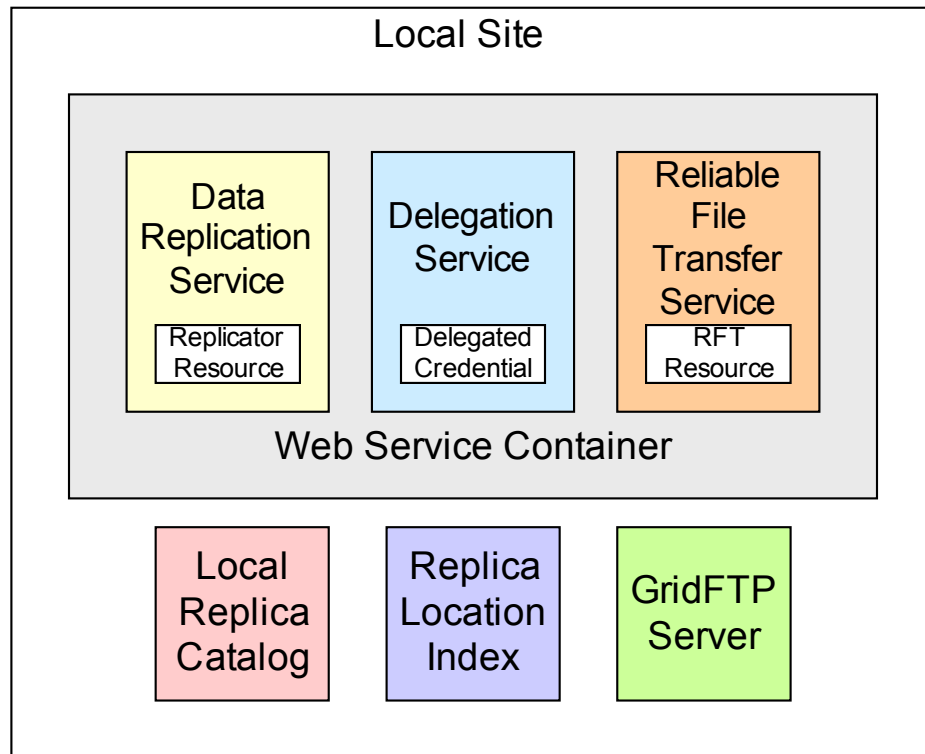
Motivation for DRS

- Need for higher-level data management services that integrate lower-level Grid functionality
 - ◆ Efficient data transfer (GridFTP, RFT)
 - ◆ Replica registration and discovery (RLS)
 - ◆ Eventually validation of replicas, etc.
- Goal is to generalize the custom data management systems developed by several application communities
- Eventually to provide a suite of configurable high-level data management services
- DRS is the first of these services

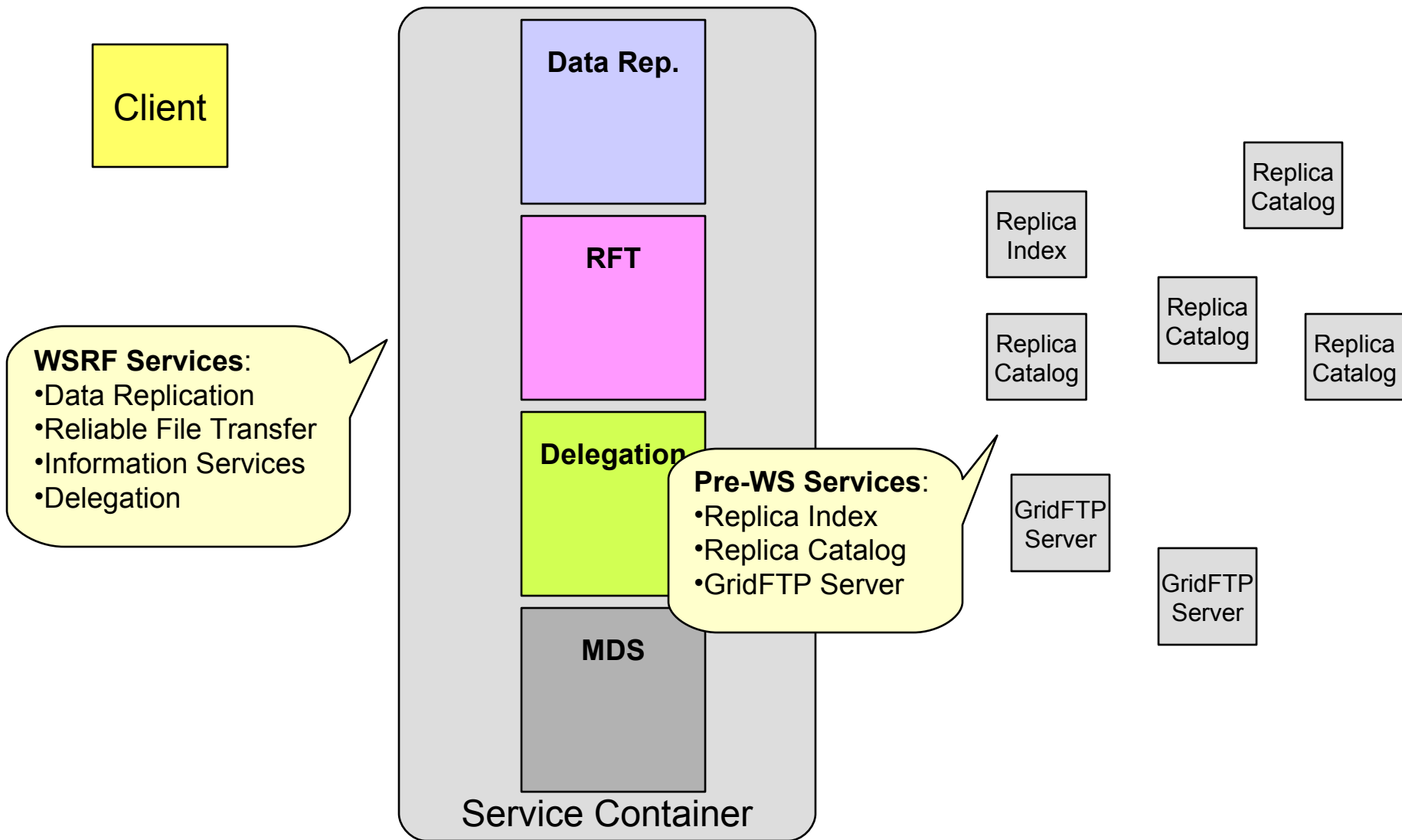
Relationship to Other Globus Services

At requesting site, deploy:

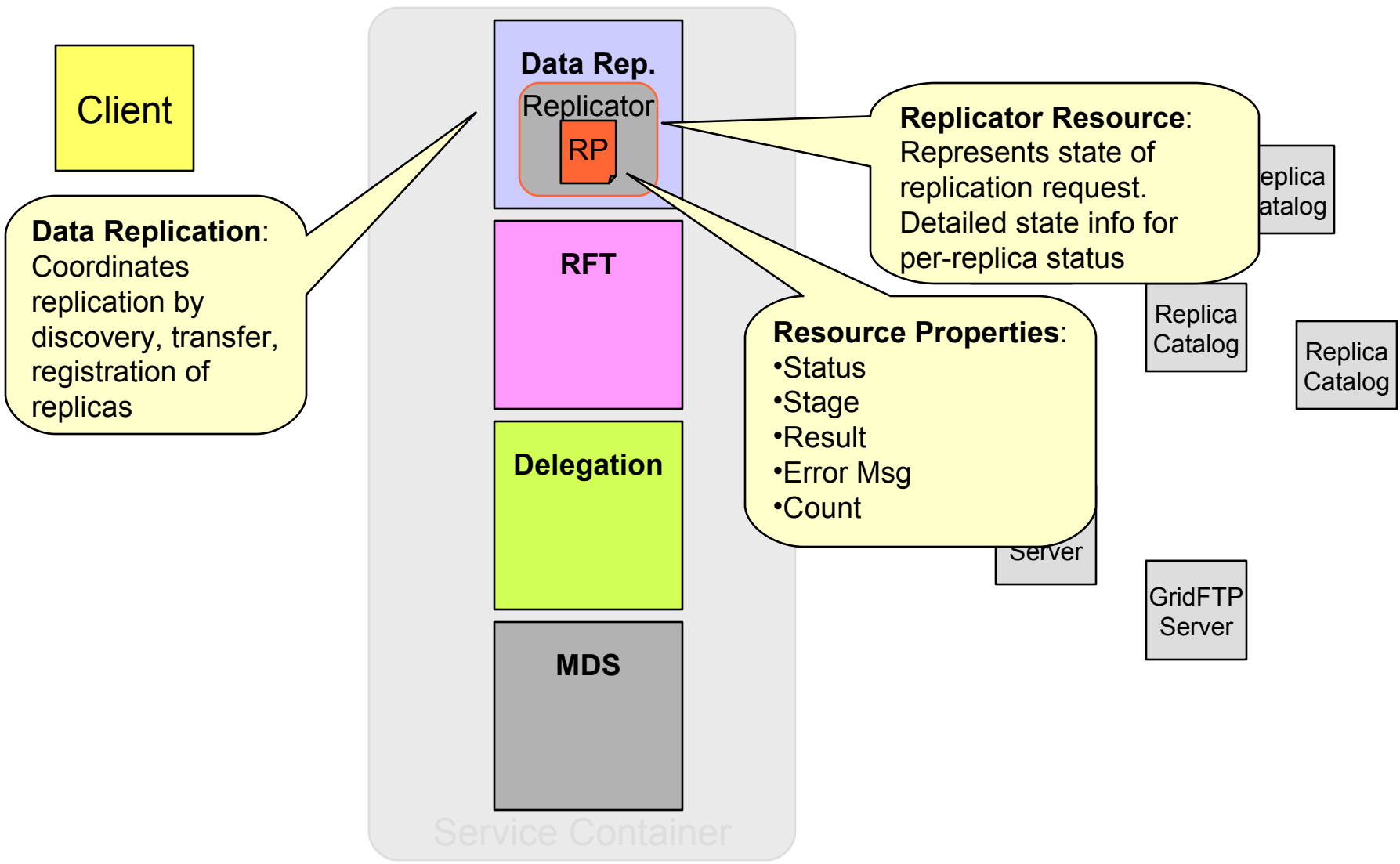
- **WS-RF Services**
 - ◆ Data Replication Service
 - ◆ Delegation Service
 - ◆ Reliable File Transfer Service
- **Pre WS-RF Components**
 - ◆ Replica Location Service (Local Replica Catalog and Replica Location Index)
 - ◆ GridFTP Server



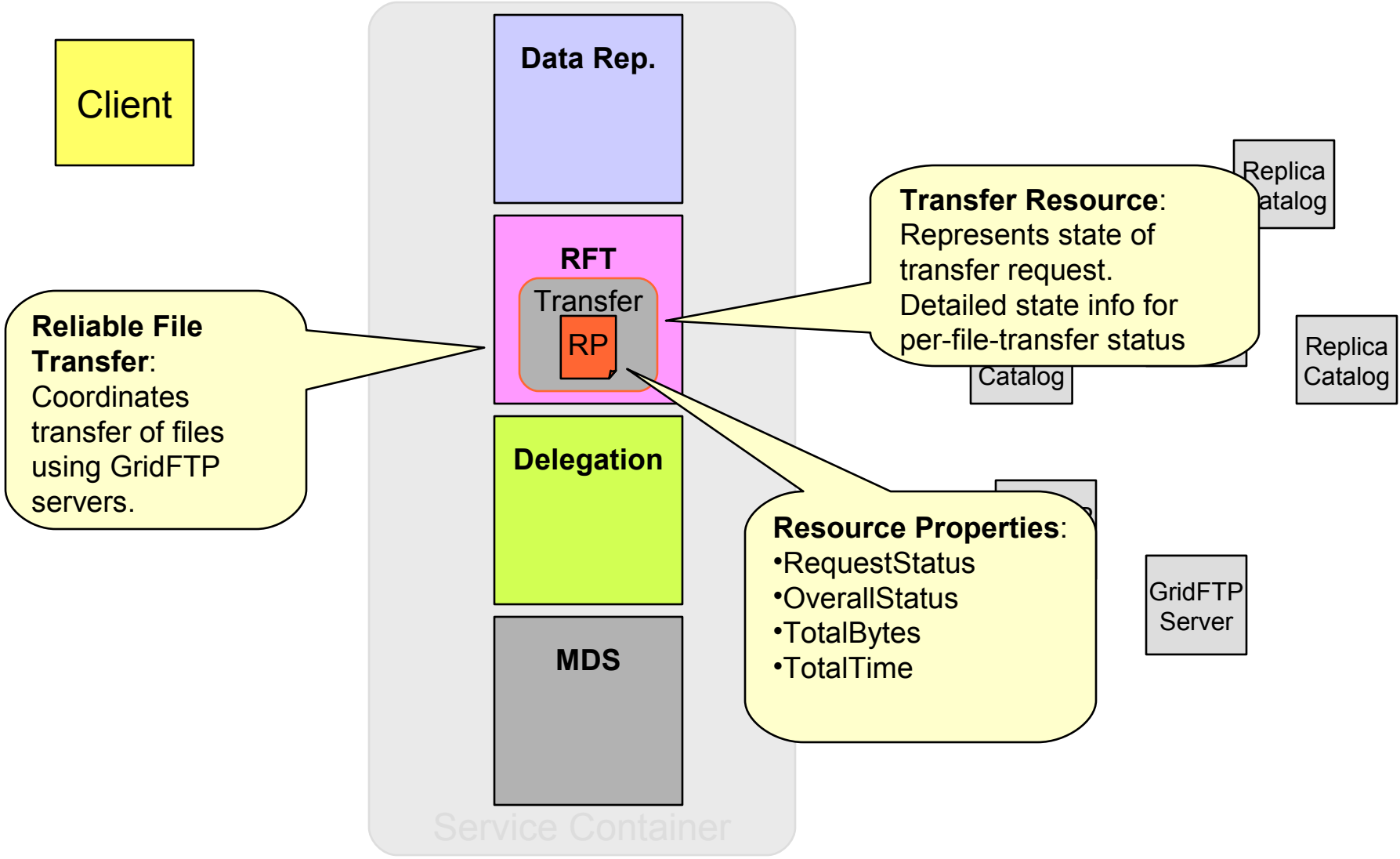
Service Composition Overview



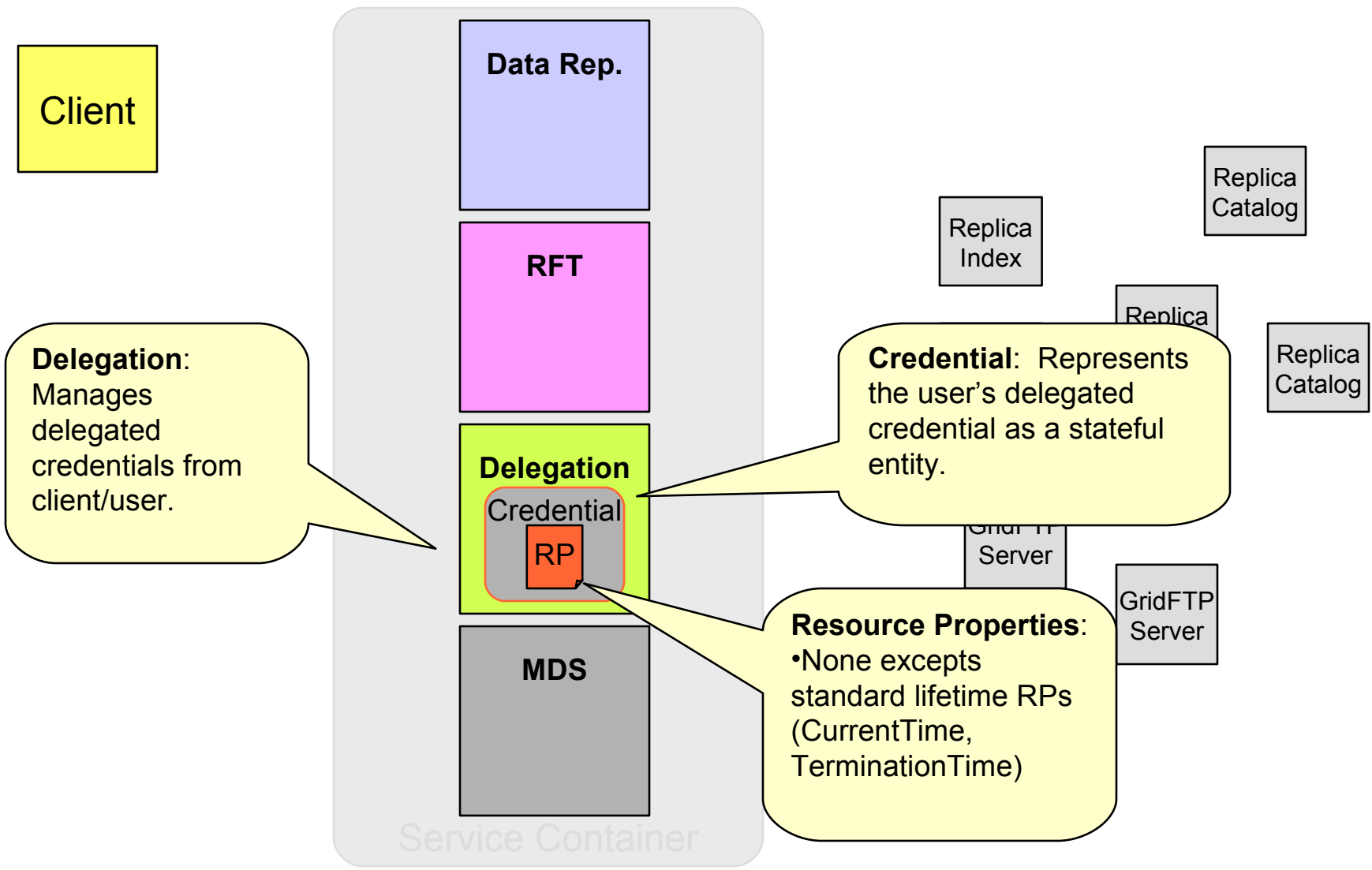
Data Replication Service



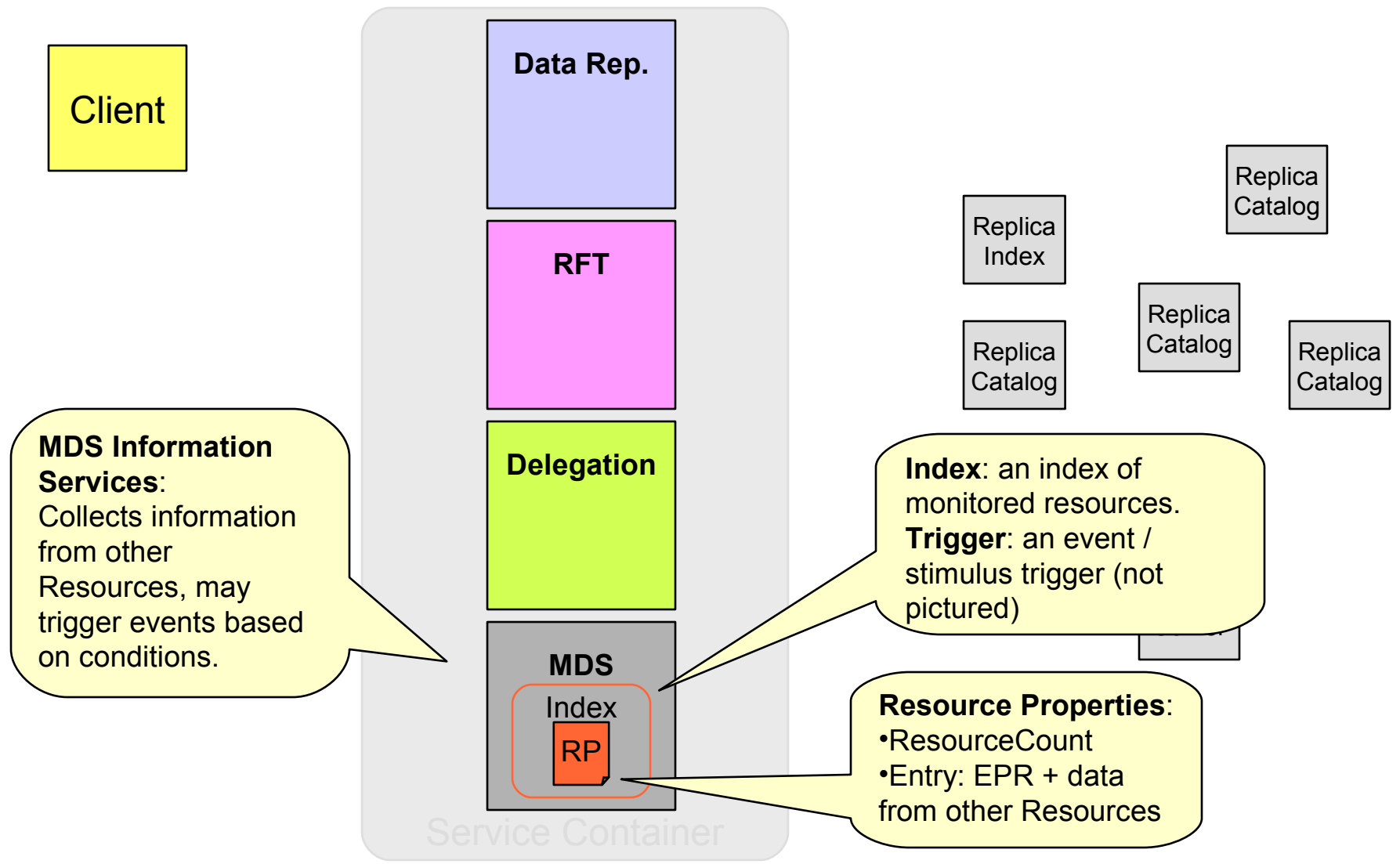
Reliable File Transfer Service



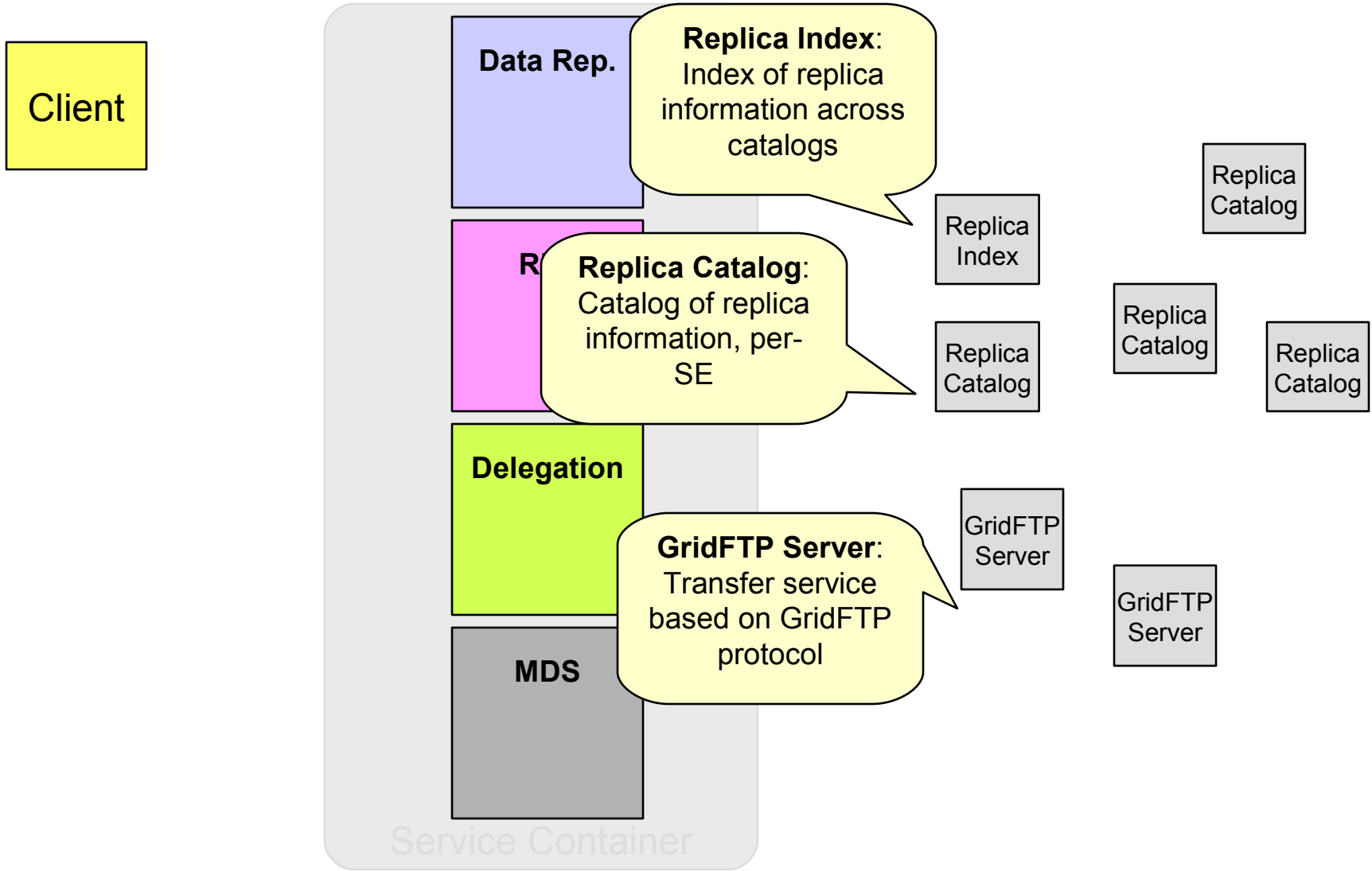
Delegation Service



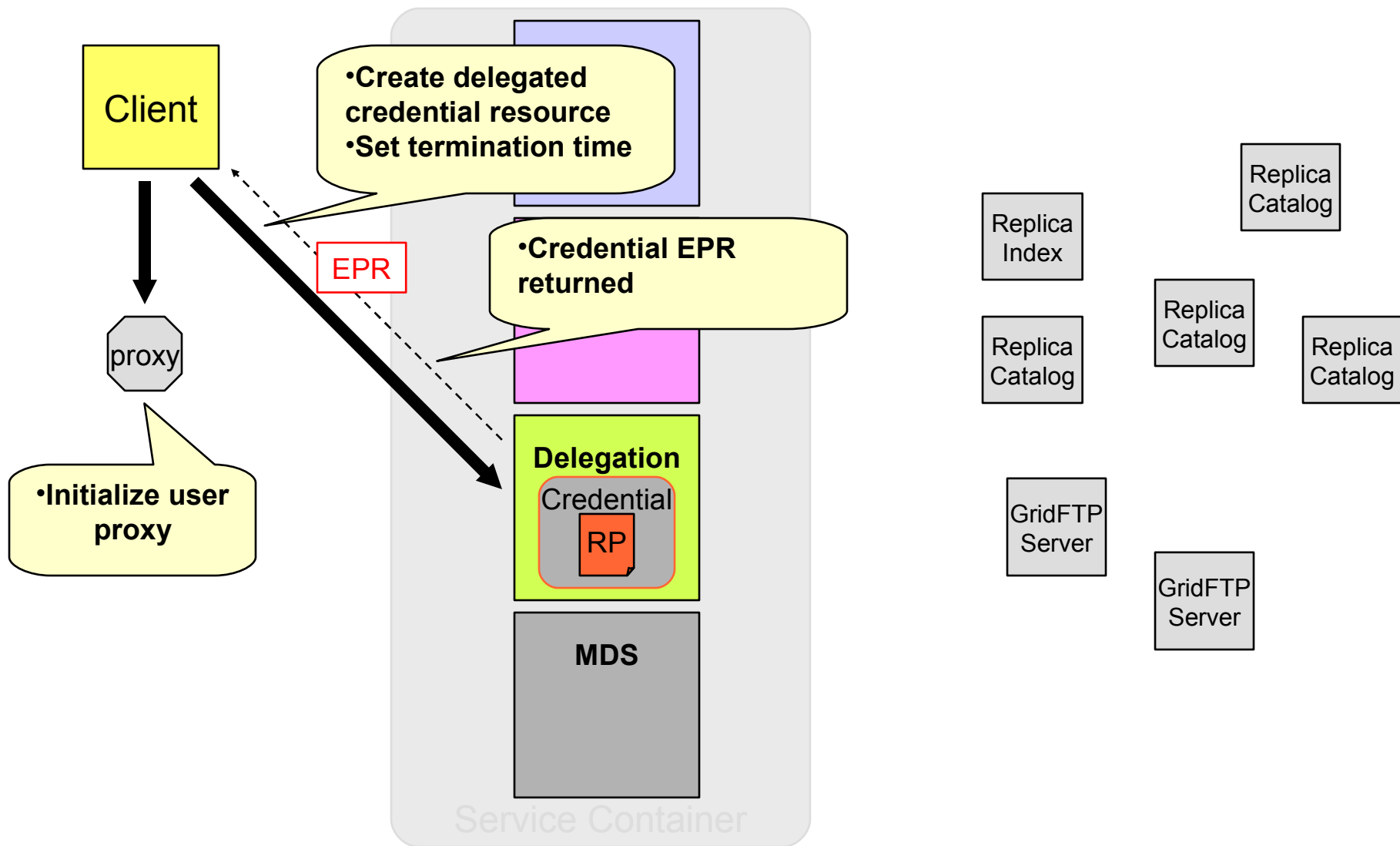
Information Services (MDS)



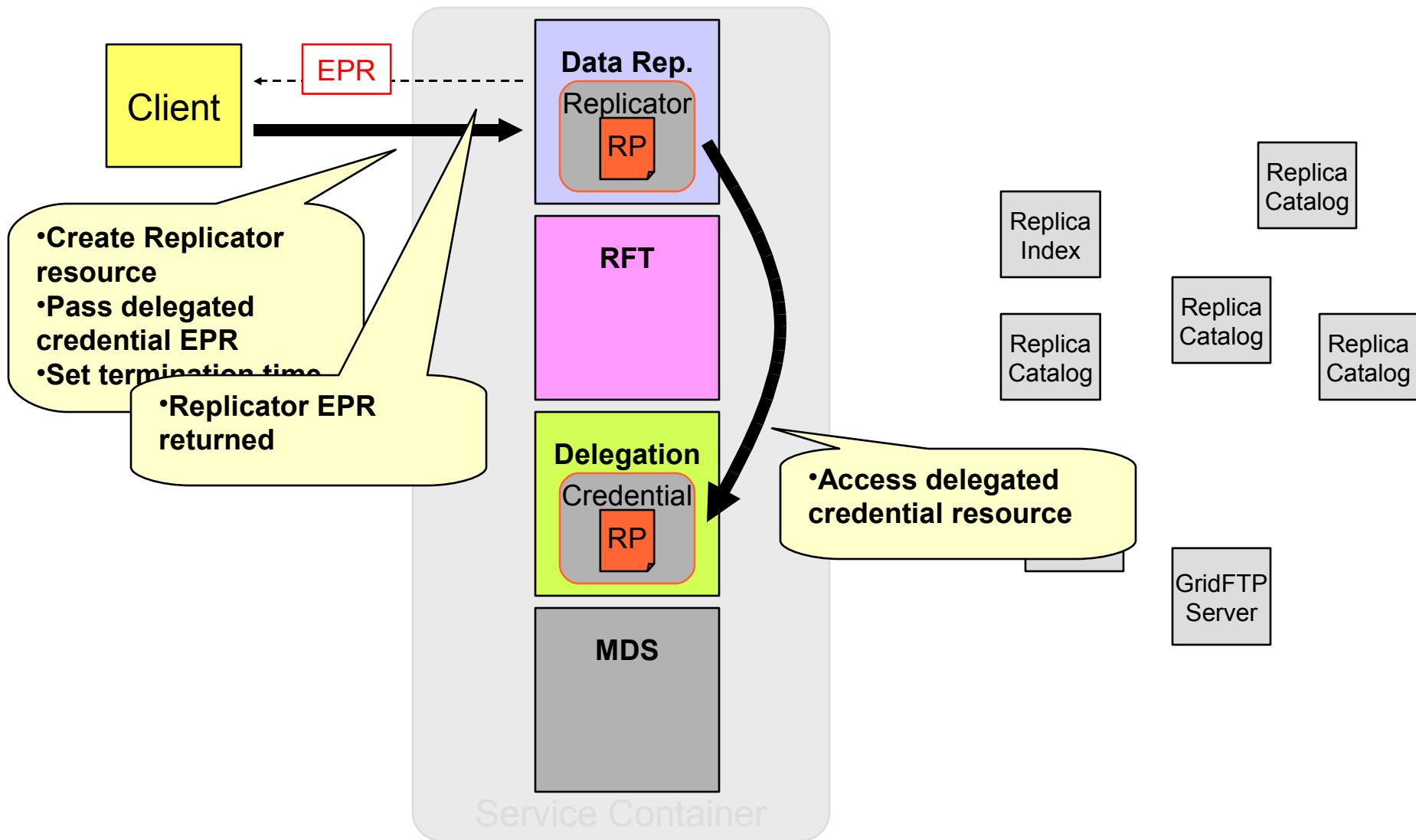
Pre-WS Services



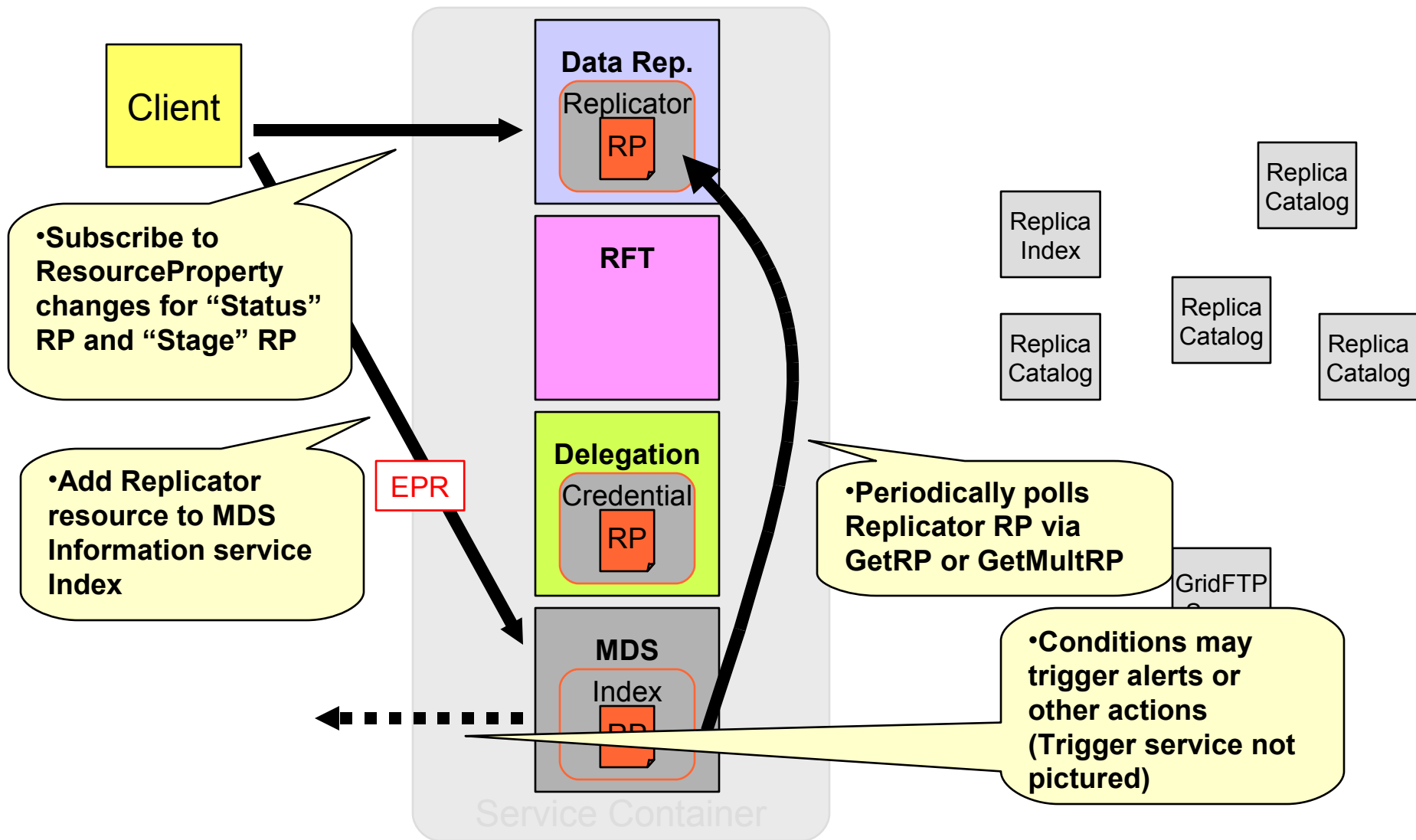
Create Delegated Credential



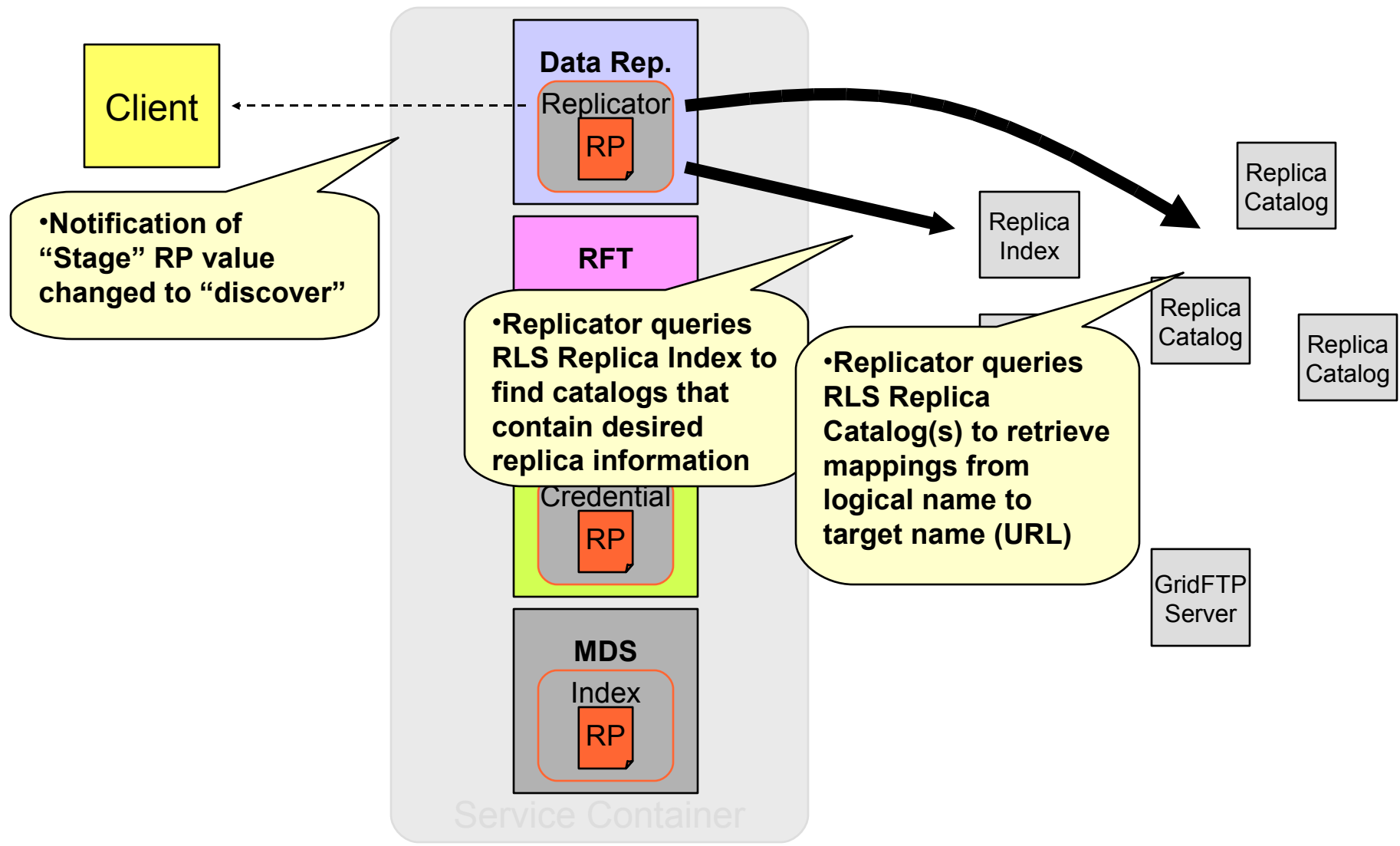
Create Delegated Credential



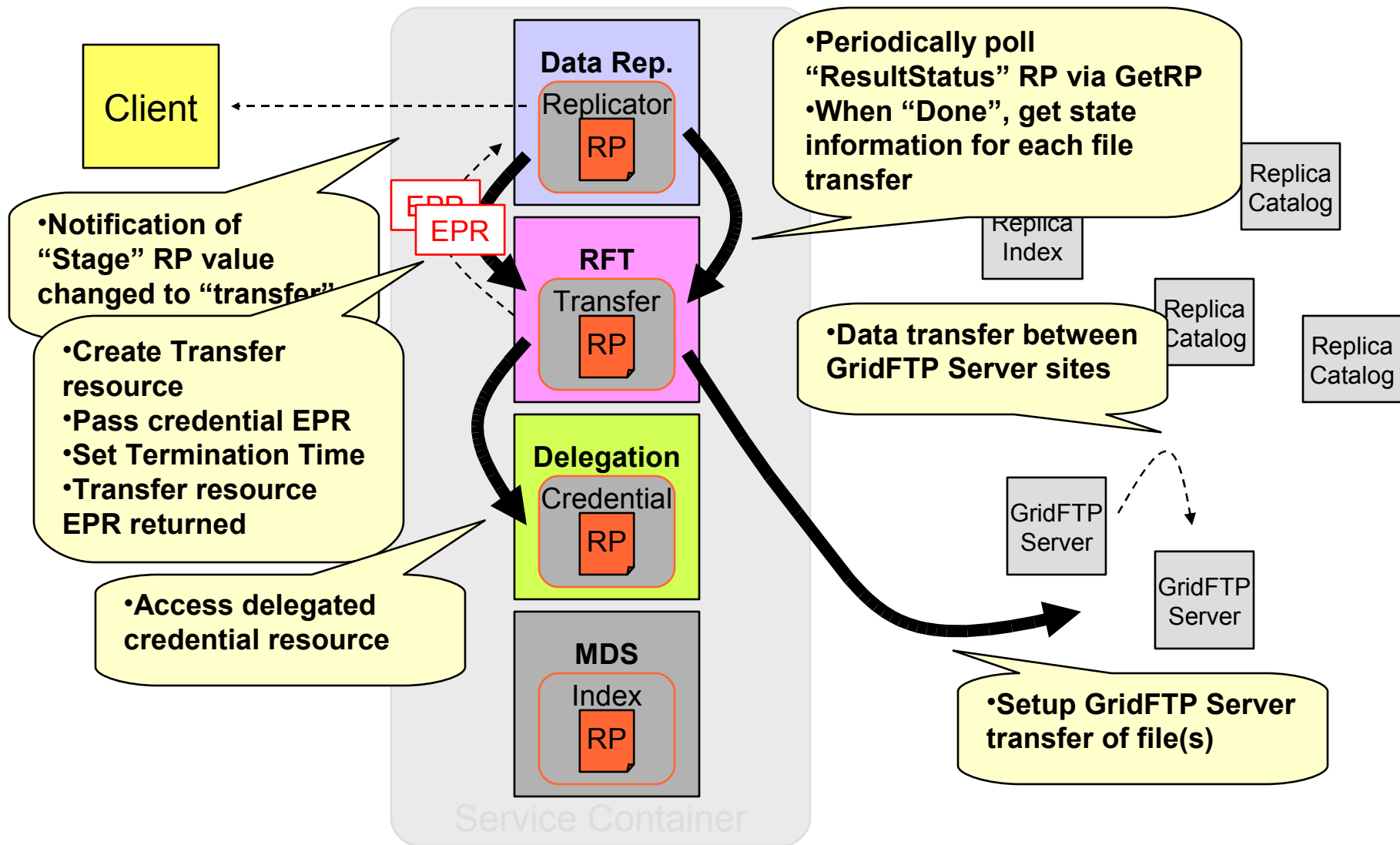
Monitor Replicator Resource



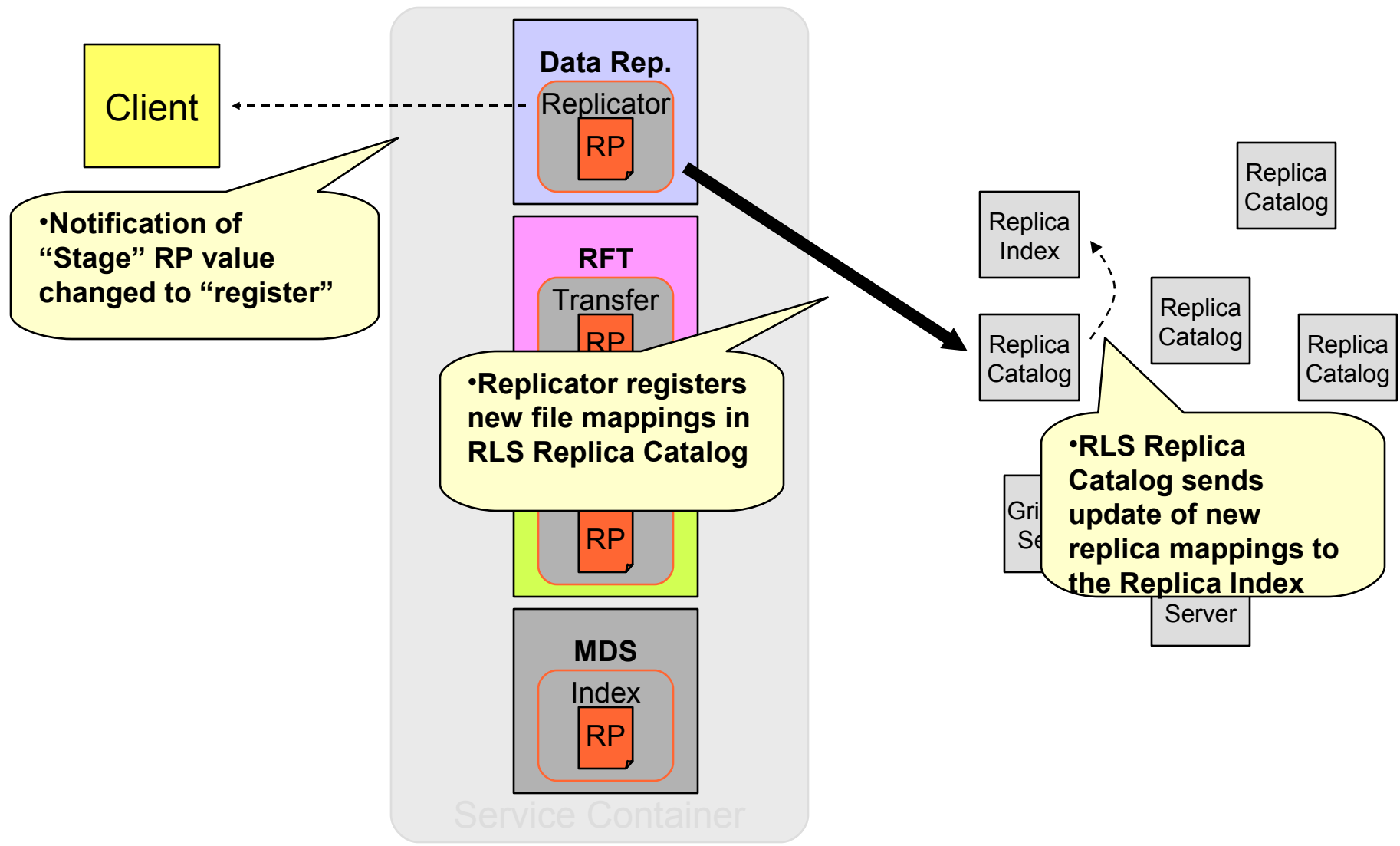
Query Replica Information



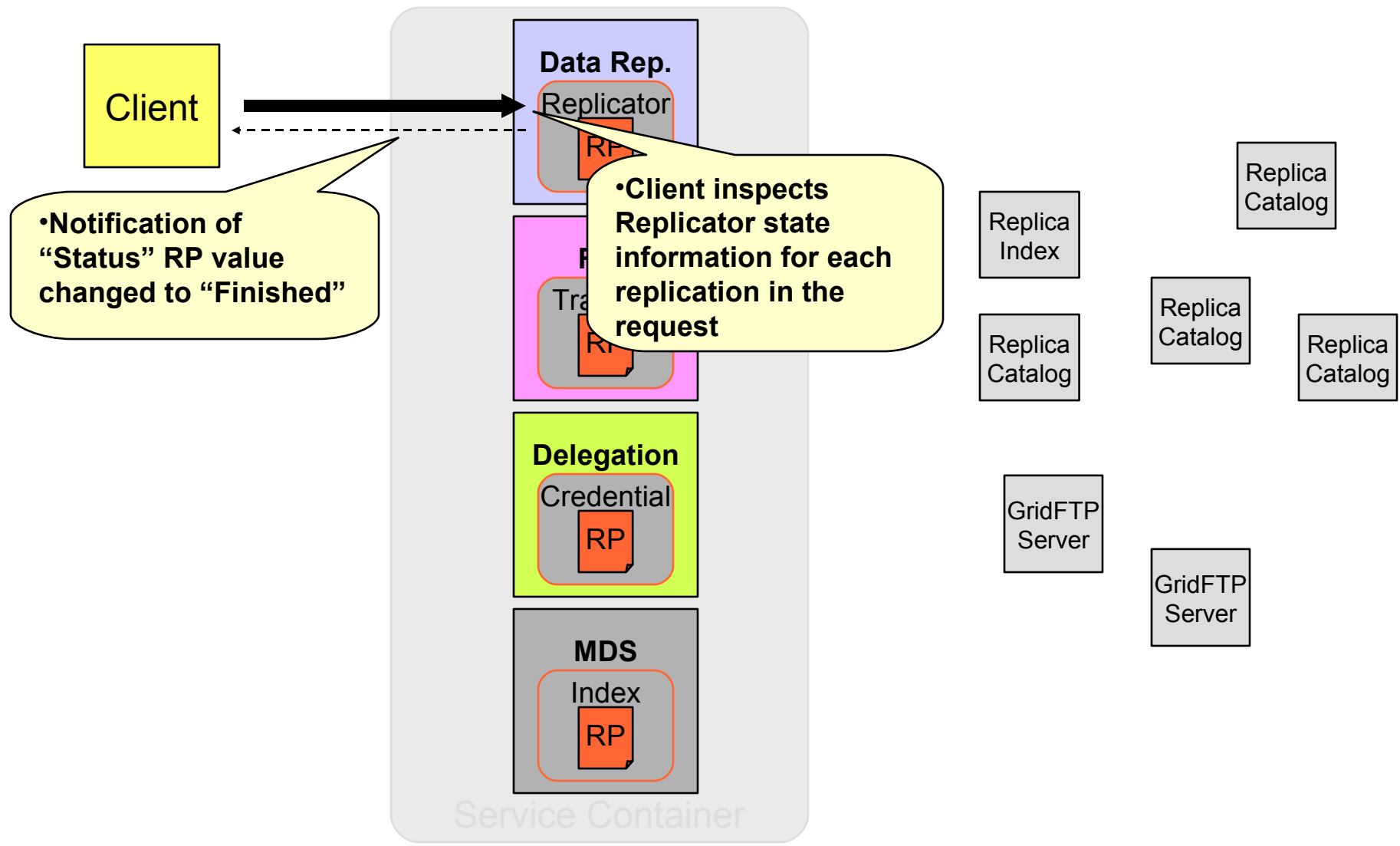
Transfer Data



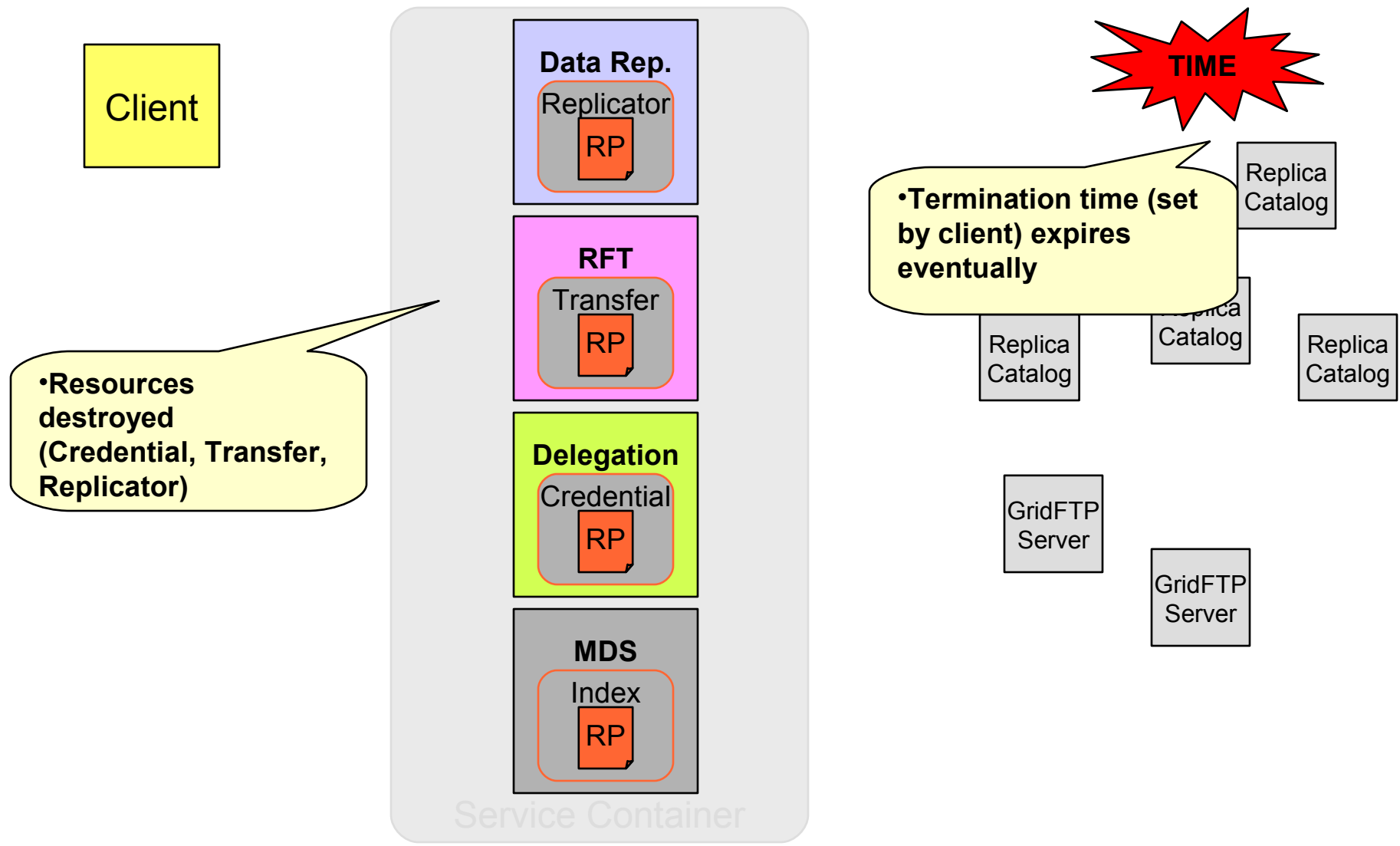
Register Replica Information



Client Inspection of State



Resource Termination



DRS: WSDL (PortType)

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions name="Replication" ...>

...

  <wsdl:portType name="ReplicatorPortType"
    wsrp:ResourceProperties="ReplicatorResourceProperties">

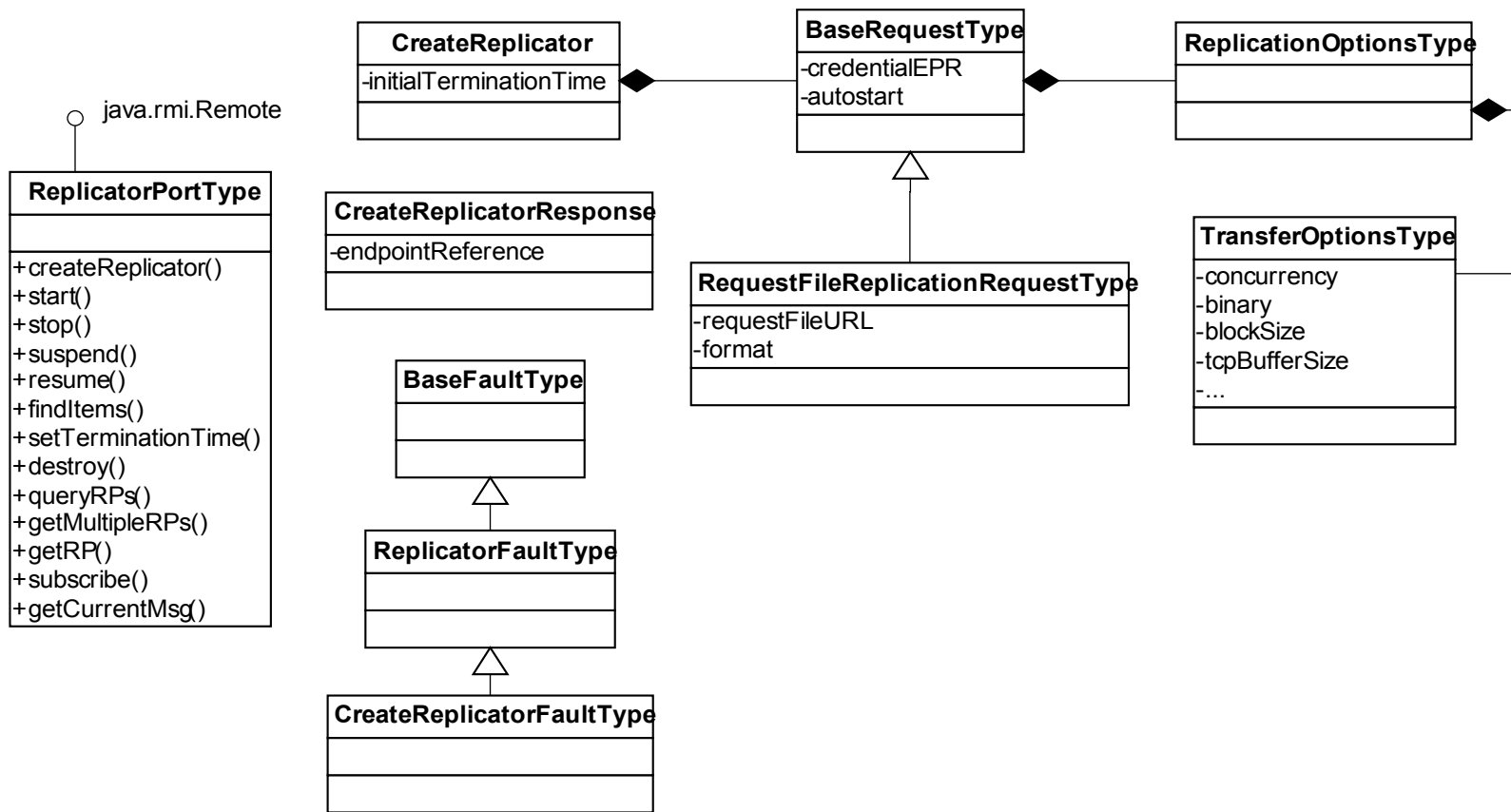
    <wsdl:operation name="createReplicator"> ...
    <wsdl:operation name="start" ...
    <wsdl:operation name="stop"> ...
    <wsdl:operation name="suspend"> ...
    <wsdl:operation name="resume"> ...
    <wsdl:operation name="findItems"> ...
    <wsdl:operation name="SetTerminationTime">
    <wsdl:operation name="Destroy"> ...
    <wsdl:operation name="QueryResourceProperties"> ...
    <wsdl:operation name="GetMultipleResourceProperties"> ...
    <wsdl:operation name="GetResourceProperty"> ...
    <wsdl:operation name="Subscribe"> ...
    <wsdl:operation name="GetCurrentMessage"> ...
  </wsdl:portType>
</wsdl:definitions>
```

DRS: WSDL (RPs)

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions name="Replication" ...>
...
  <wsdl:portType name="ReplicatorPortType"
    wsrp:ResourceProperties="ReplicatorResourceProperties">
    <wsdl:operation name="createReplicator" ...>
    <wsdl:operation name="start" ...>
    <wsdl:operation name="stop"> ...
    <wsdl:operation name="suspend"> ...
    <wsdl:operation name="resume"> ...
    <wsdl:operation name="findItems"> ...
    <wsdl:operation name="SetTerminationTime"> ...
    <wsdl:operation name="Destroy"> ...
    <wsdl:operation name="QueryResource"> ...
    <wsdl:operation name="GetMultipleResources"> ...
    <wsdl:operation name="GetResource"> ...
    <wsdl:operation name="Subscribe"> ...
    <wsdl:operation name="GetCurrentMetadata"> ...
  </wsdl:portType>
</wsdl:definitions>
```

```
<xsd:element name="ReplicatorResourceProperties">
...
  <xsd:element name="status" .../>
  <xsd:element name="stage" .../>
  <xsd:element name="result" .../>
  <xsd:element name="errorMessage" .../>
  <xsd:element name="count" .../>
  <xsd:element name="Topic" .../>
  <xsd:element name="TopicExprDialect" .../>
  <xsd:element name="TerminationTime" .../>
  <xsd:element name="CurrentTime" .../>
  <xsd:element name="FixedTopicSet" .../>
...
</xsd:element>
```

DRS: Stubs

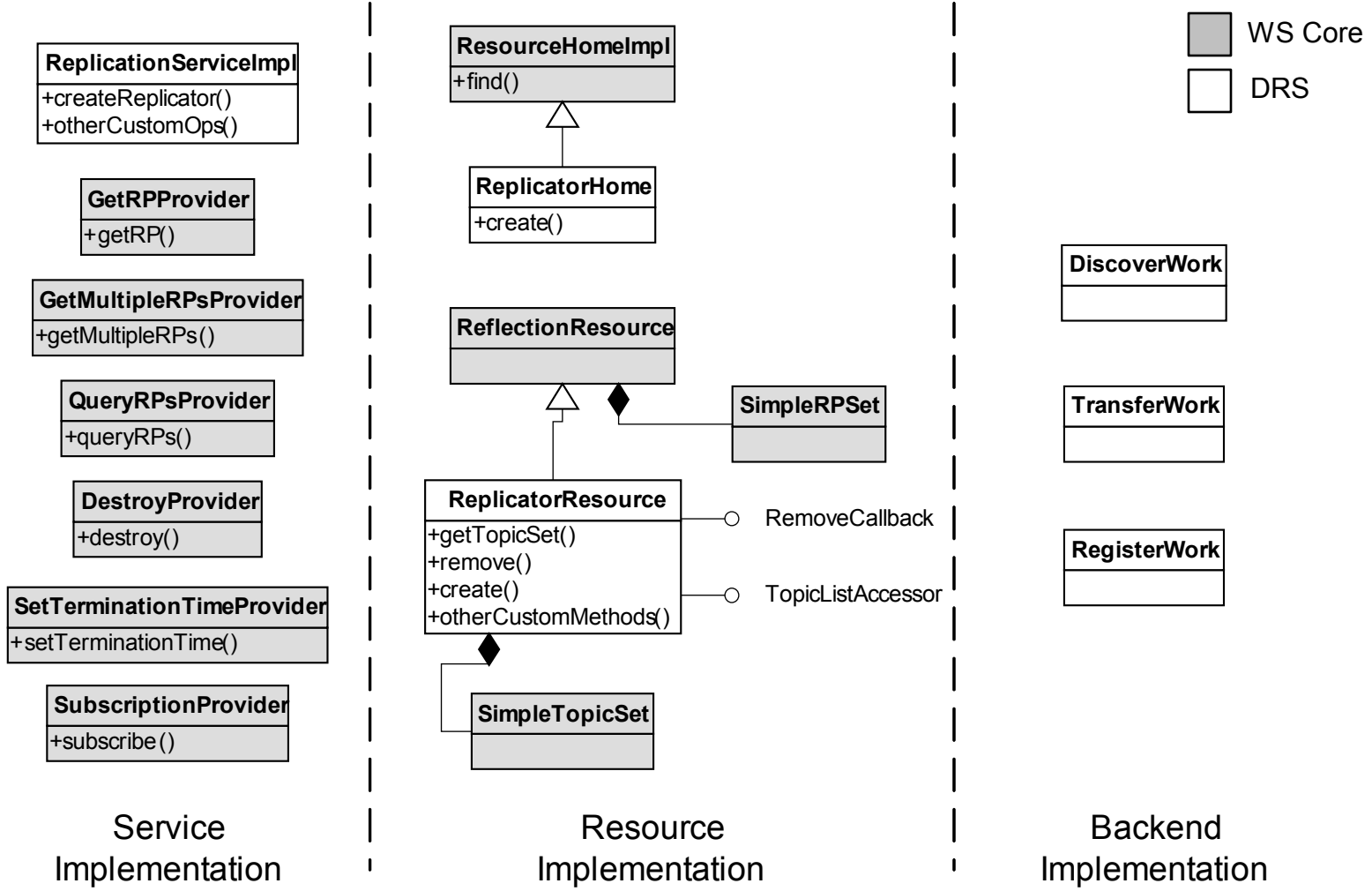


PortType

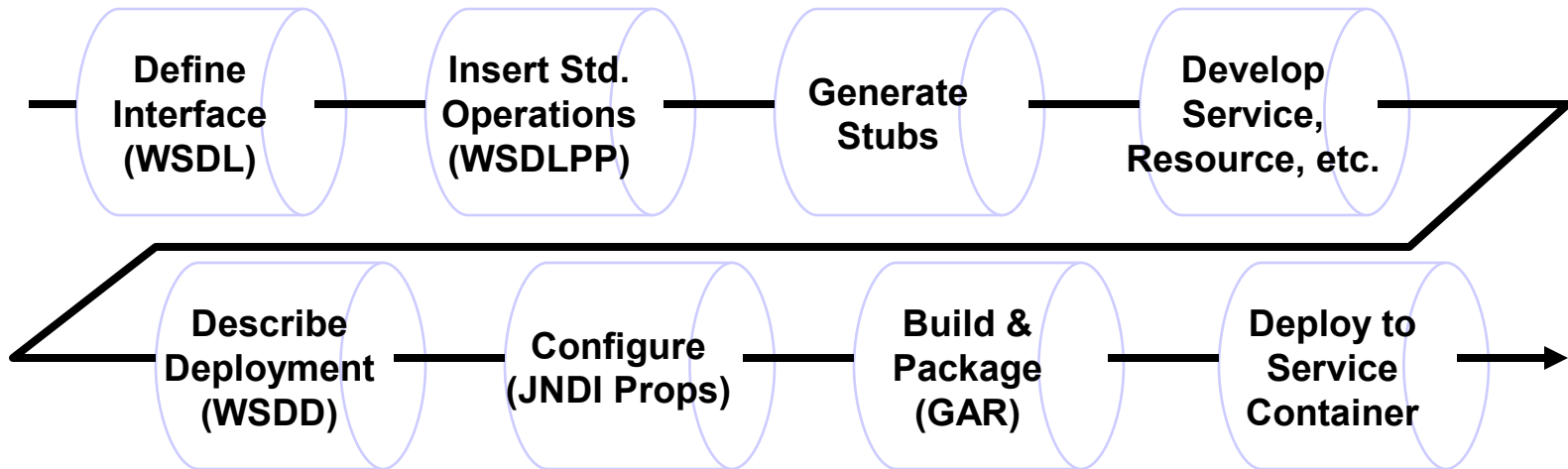
Messages

Types

DRS: Implementation



Development and Deployment Process



1. Define Interface (WSDL, XSD)
2. WSDLPP (adds standard operations)
3. Generate Stubs
4. Develop service, resource, custom logic
5. Describe deployment (WSDD)
6. Configure JNDI properties
7. Build, package as GAR, and deploy to container
8. Deploy to Service Container

The Globus Ecosystem

- Globus **components** address core issues relating to resource access, monitoring, discovery, security, data movement, etc.
 - ◆ GT4 being the latest version
- A larger Globus **ecosystem** of open source and proprietary components provide complementary components
 - ◆ A growing list of components
- These components can be combined to produce **solutions** to Grid problems
 - ◆ We're building a list of such solutions

Many Tools Build on, or Can Contribute to, GT4-Based Grids

- Condor-G, DAGman
- MPICH-G2
- GRMS
- Nimrod-G
- Ninf-G
- Open Grid Computing Env.
- Commodity Grid Toolkit
- GriPhyN Virtual Data System
- Virtual Data Toolkit
- GridXpert Synergy
- Platform Globus Toolkit
- VOMS
- PERMIS
- GT4IDE
- Sun Grid Engine
- PBS scheduler
- LSF scheduler
- GridBus
- TeraGrid CTSS
- NEES
- IBM Grid Toolbox
- ...

Example Solutions

- Portal-based User Reg. System (PURSE)
- VO Management Registration Service
- Service Monitoring Service
- TeraGrid TGCP Tool
- Lightweight Data Replicator
- GriPhyN Virtual Data System

The Globus Commitment to Open Source

- Globus was first established as an open source project in 1996
- The Globus Toolkit is open source to:
 - ◆ allow for inspection
 - for consideration in standardization processes
 - ◆ encourage adoption
 - in pursuit of ubiquity and interoperability
 - ◆ encourage contributions
 - harness the expertise of the community
- The Globus Toolkit is distributed under the (BSD-style) Apache License version 2

The Future: Structure

- NSF Community Driven Improvement of Globus Software (CDIGS) project
 - ◆ 5 years of funding for GT enhancement
 - ◆ Regular Globus roadmaps outlining plans
- GlobDev <http://dev.globus.org>
 - ◆ Apache-like community development site
 - ◆ Community governance of components
 - ◆ “Globus Toolkit” & other related software
 - ◆ Open for business early 2006
 - ◆ “Globus Alliance” = “GlobDev committers”

The Future: Content

- We now have a solid and extremely powerful Web services base
- Next, we will build an expanded open source Grid infrastructure
 - ◆ Virtualization
 - ◆ New services for provisioning, data management, security, VO management
 - ◆ End-user tools for application development
 - ◆ Etc., etc.
- And of course responding to user requests for other short-term needs

What to Expect from the Globus Alliance in the Coming Months

- Support for users of GT4
 - ◆ Working to make sure the toolkit meets user needs
 - ◆ Answering questions on the mailing lists
 - ◆ Further improving documentation
- Normal evolution of performance, scalability and feature enhancements
- Further development of tools and services in support of VOs
- Expanding contributions to Globus

GlobDev

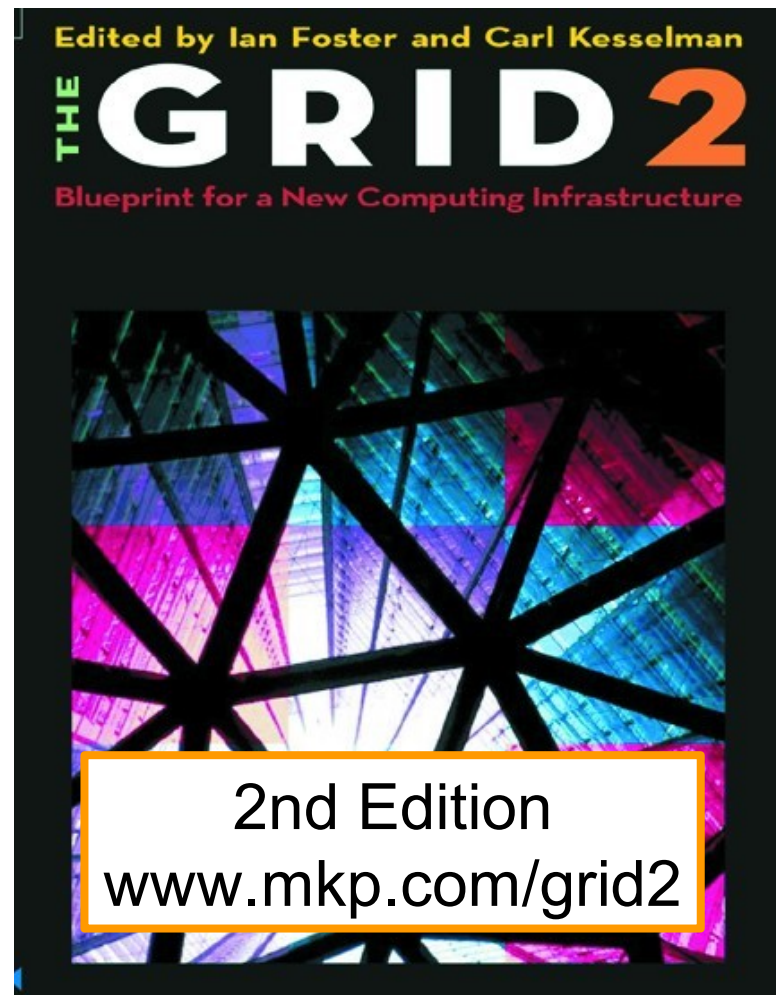
- The current set of Globus components will be organized into several “Globus Projects”
 - ◆ Projects release products
- Each project will have its own group of “Committers”
 - ◆ committers are responsible for governance on matters relating to their products
- The “Globus Management Committee” will
 - ◆ provide overall guidance and conflict resolution
 - ◆ approve the creation of new Globus Projects

Opportunities for Collaboration

- Use of Globus software
 - ◆ Feedback & involvement in design
- Development of new Globus components
 - ◆ E.g., new information providers to enable use of GT to manage an entire Grid
 - ◆ Examples and documentation
- Globalization and localization of software
- New applications and tools
 - ◆ E.g., Grid operations, emergency response, ecogrid, bioinformatics, ...

For More Information

- Globus Alliance
 - ◆ www.globus.org
- NMI and GRIDS Center
 - ◆ www.nsf-middleware.org
 - ◆ www.grids-center.org
- Infrastructure
 - ◆ www.opensciencegrid.org
 - ◆ www.teragrid.org



For More Information

- GT4 Programming

