Grid Computing and the Globus Toolkit

Jennifer M. Schopf

Argonne National Lab

http://www.mcs.anl.gov/~jms/Talks/

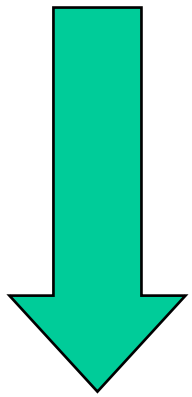
Talk This morning

- Overview of Grids and Globus
- Walk through some use cases
- A bit about Toolkit and contributing

- Please ask questions as they occur to you
  - Extra slides not shown
- Slides will be available online when I get net (or by USB at a break)
- Afternoon is hands on – a little about that just before lunch

Where do these slides come from?

- All these slides are open use
  - Covered under same license as the rest of Globus (Apache 2)
- Many slides donated from other folks
- I've tried to make this known



Donater Name , http://web page when I can

Slides Like This

- Transitions to a new part of the talk

What is a Grid?

- Resource sharing
  - Computers, storage, sensors, networks, …
  - Sharing always conditional: issues of trust, policy, negotiation, payment, …
- Coordinated problem solving
  - Beyond client-server: distributed data analysis, computation, collaboration, …
- Dynamic, multi-institutional virtual orgs
  - Community overlays on classic org structures
  - Large or small, static or dynamic

An Old Idea …

- "The time-sharing computer system can unite a group of investigators …. one can conceive of such a facility as an … intellectual public utility."
  - Fernando Corbato and Robert Fano, 1966
- "We will perhaps see the spread of 'computer utilities', which, like present electric and telephone utilities, will service individual homes and offices across the country."
  - Len Kleinrock, 1967

Why Is this Hard or Different?

- Lack of central control
  - Where things run
  - When they run
- Shared resources
  - Contention, variability
- Communication and coordination
  - Different sites implies different sys admins, users, institutional goals, and often socio-political constraints

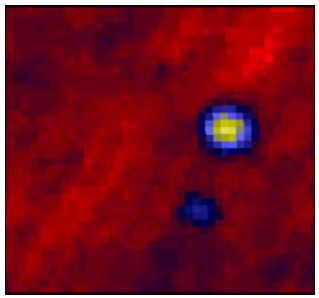
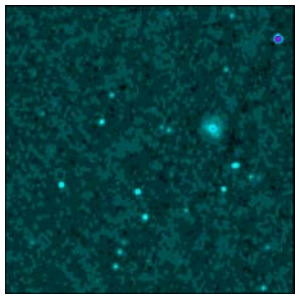
So Why Do It?

- Computations that need to be done with a time limit
- Data that can't fit on one site
- Data owned by multiple sites

- Applications that need to be run bigger, faster, more

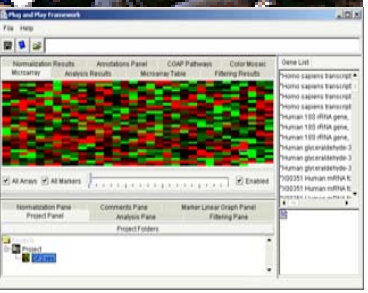
For Example: Digital Astronomy

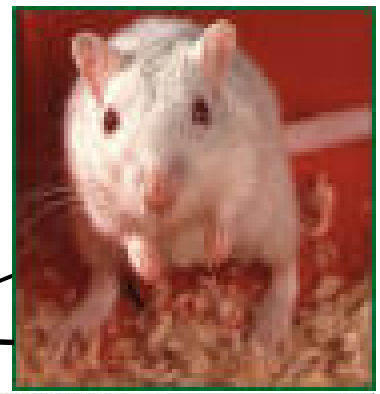
- Digital observatories provide online archives of data at different wavelengths

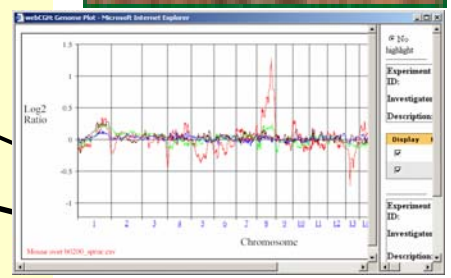



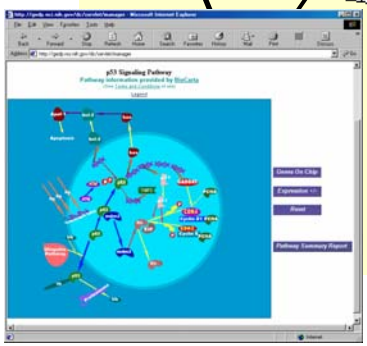
- Ask questions such as: what objects are visible in infrared but not visible spectrum?

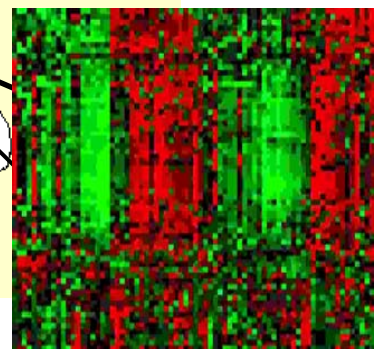
For Example: Cancer Biology





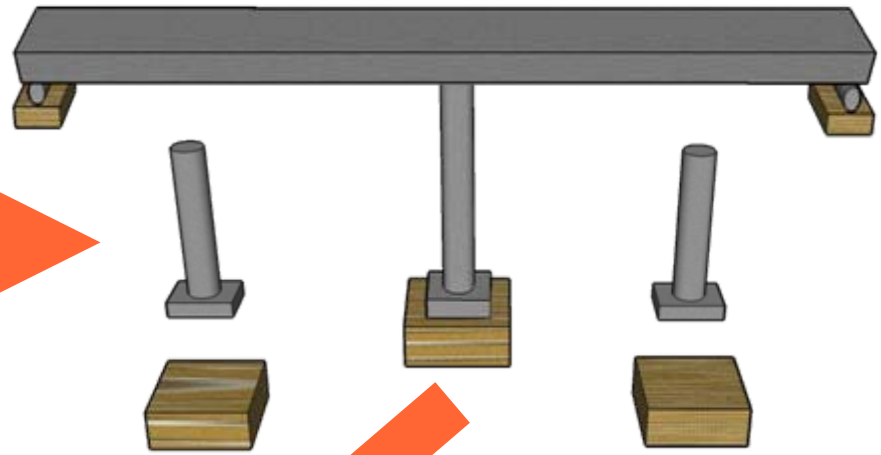






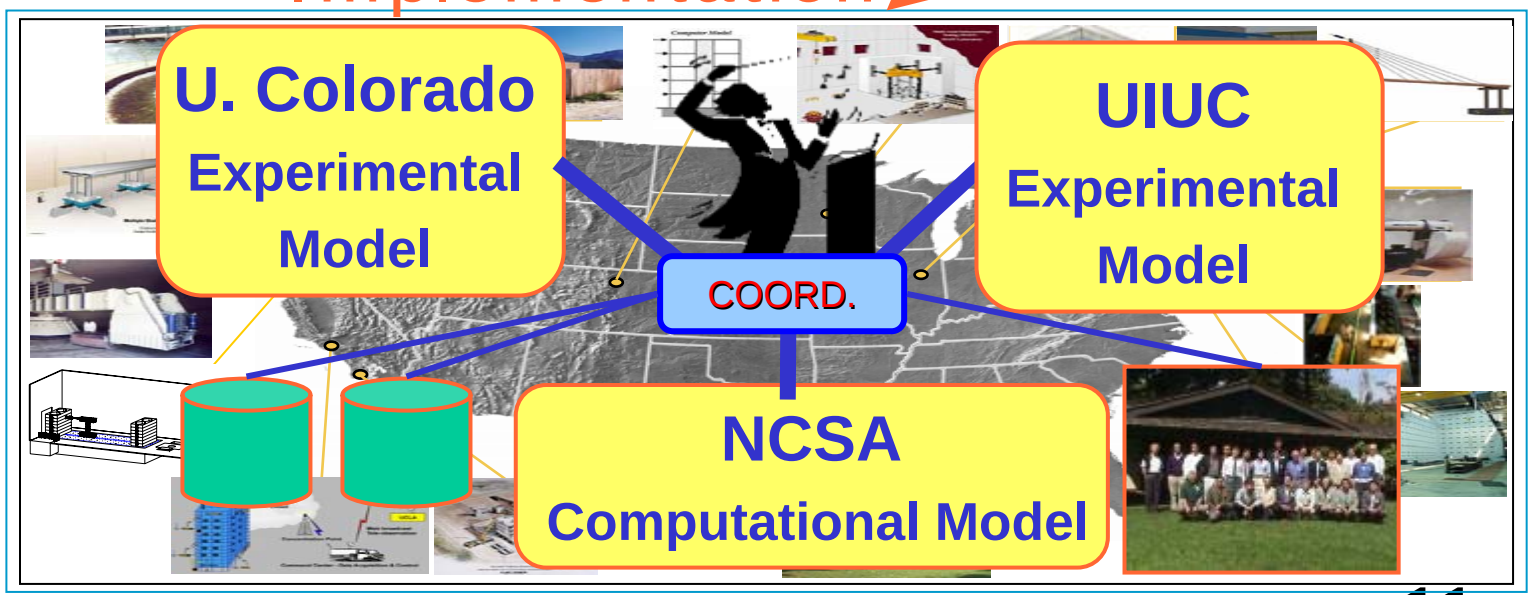
System-Level Problem

Decomposition



Implementation

Facilities
Computers
Storage
Networks
Services
Software
People

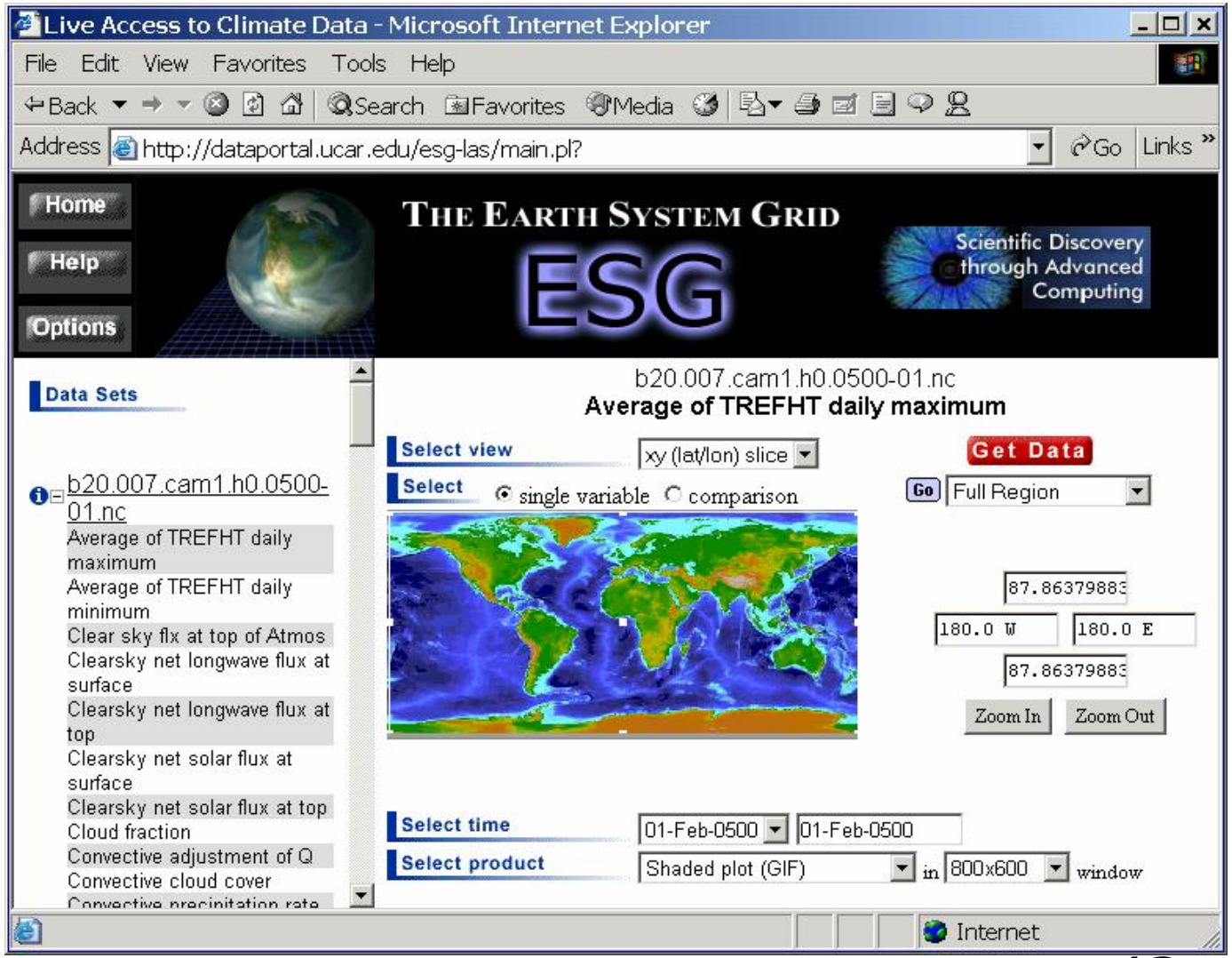


the globus alliance
www.globus.org

Earth System Grid

DOE Earth System Grid

Goal: Enable sharing & analysis of high-volume data from advanced earth system models



What Kinds of Applications?

- Computation intensive
  - Interactive simulation (climate modeling)
  - Large-scale simulation and analysis (galaxy formation, gravity waves, event simulation)
  - Engineering (parameter studies, linked models)
- Data intensive
  - Experimental data analysis (e.g., physics)
  - Image & sensor analysis (astronomy, climate)
- Distributed collaboration
  - Online instrumentation (microscopes, x-ray) Remote visualization (climate studies, biology)
  - Engineering (large-scale structural testing)

Key Common Features

- The size and/or complexity of the problem
- Collaboration between people in several organizations
- Sharing computing resources, data, instruments

Grid Infrastructure

- Distributed management
  - Of physical resources
  - Of software services
  - Of communities and their policies
- Unified treatment
  - Build on Web services framework
  - Use WS-RF, WS-Notification (or WS-Transfer/Man) to represent/access state
  - Common management abstractions & interfaces

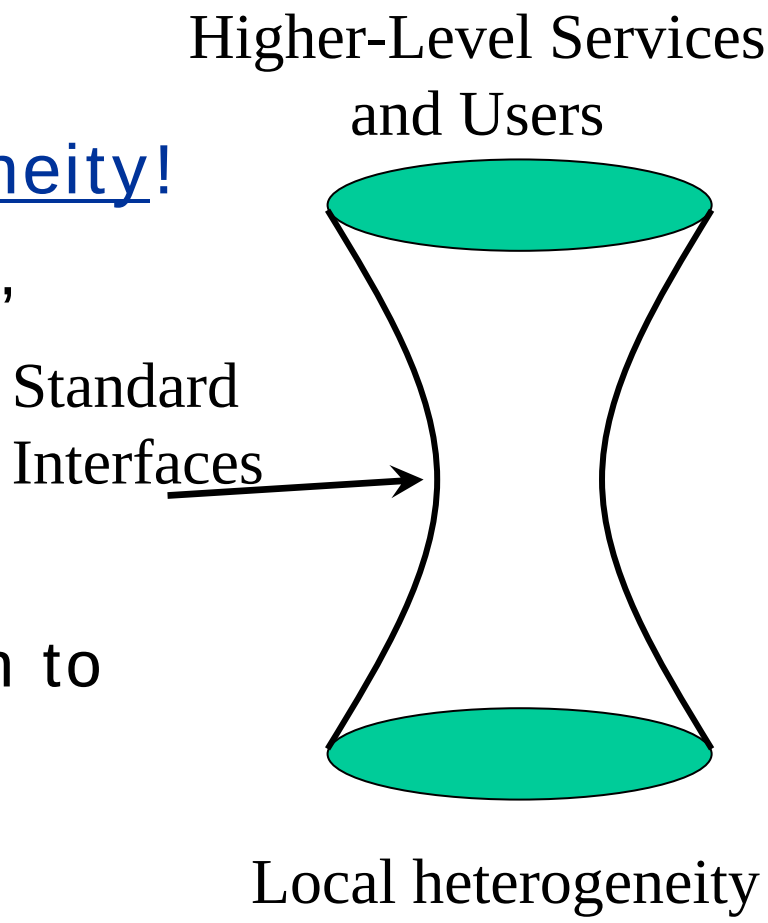
The Globus Approach: Philosophy and Technology

Globus is…

- A collection of solutions to problems that come up frequently when building collaborative distributed applications
- Software for Grid infrastructure
  - Service enable new & existing resources
  - 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

Globus is an Hour Glass

- Local sites have their own policies, installs – heterogeneity!
  - Queuing systems, monitors, network protocols, etc
- Globus unifies – standards!
  - Build on Web services
  - Use WS-RF, WS-Notification to represent/access state
  - Common management abstractions & interfaces



Globus is a Building Block

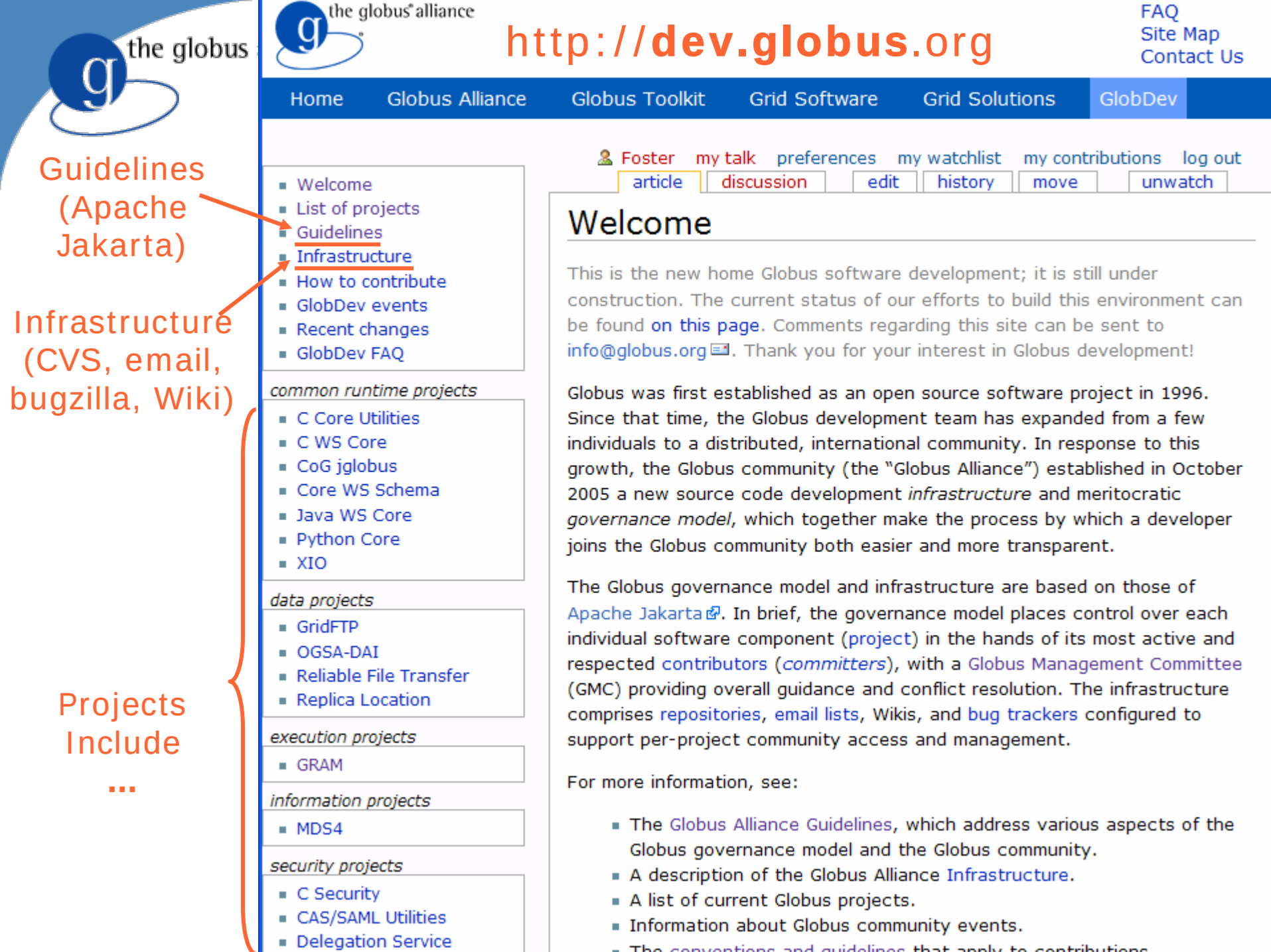
- Basic components for Grid functionality
  - Not turnkey solutions, but building blocks & tools for application developers & system integrators
- Highest-level services are often application specific, we let aps concentrate there
- Easier to reuse than to reinvent
  - Compatibility with other Grid systems comes for free
- We provide basic infrastructure to get you one step closer

Globus Philosophy

- 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

dev.globus

- Governance model based on Apache Jakarta
  - Consensus based decision making
- Globus software is organized as several dozen “Globus Projects”
  - Each project has its own “Committers” responsible for their products
  - Cross-project coordination through shared interactions and committers meetings
- A “Globus Management Committee”
  - Overall guidance and conflict resolution


the globus
Guidelines (Apache Jakarta)
Infrastructure (CVS, email, bugzilla, Wiki)
Projects Include …
the globus alliance
http://dev.globus.org
FAQ
Site Map
Contact Us
Home
Globus Alliance
Globus Toolkit
Grid Software
Grid Solutions
GlobDev
Welcome
List of projects
Guidelines
Infrastructure
How to contribute
GlobDev events
Recent changes
GlobDev FAQ
common runtime projects
C Core Utilities
C WS Core
CoG jglobus
Core WS Schema
Java WS Core
Python Core
XIO
data projects
GridFTP
OGSA-DAI
Reliable File Transfer
Replica Location
execution projects
GRAM
information projects
MDS4
security projects
C Security
CAS/SAML Utilities
Delegation Service
Foster
my talk
preferences
my watchlist
my contributions
log out
article
discussion
edit
history
move
unwatch
Welcome
This is the new home Globus software development; it is still under construction. The current status of our efforts to build this environment can be found on this page. Comments regarding this site can be sent to info@globus.org. Thank you for your interest in Globus development!
Globus was first established as an open source software project in 1996. Since that time, the Globus development team has expanded from a few individuals to a distributed, international community. In response to this growth, the Globus community (the "Globus Alliance") established in October 2005 a new source code development infrastructure and meritocratic governance model, which together make the process by which a developer joins the Globus community both easier and more transparent.
The Globus governance model and infrastructure are based on those of Apache Jakarta. In brief, the governance model places control over each individual software component (project) in the hands of its most active and respected contributors (committers), with a Globus Management Committee (GMC) providing overall guidance and conflict resolution. The infrastructure comprises repositories, email lists, Wikis, and bug trackers configured to support per-project community access and management.
For more information, see:
The Globus Alliance Guidelines, which address various aspects of the Globus governance model and the Globus community.
A description of the Globus Alliance Infrastructure.
A list of current Globus projects.
Information about Globus community events.

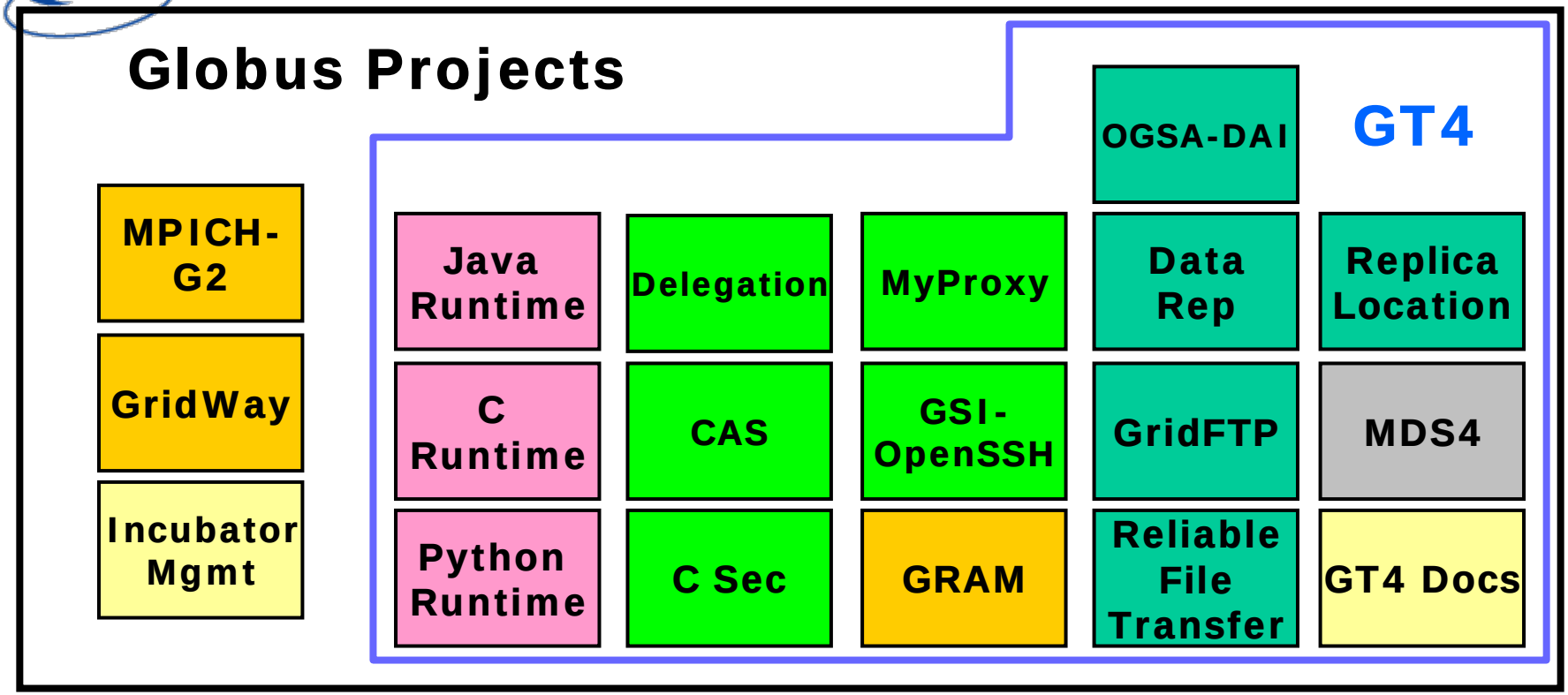
Globus Technology Areas

- 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

Non-Technology Projects

- Distribution Projects
  - Globus Toolkit Distribution
  - Process in use since April ’07
- Documentation Projects
  - GT Release Manuals
- Incubation Projects
  - Incubation management project
  - And any new projects wanting to join

Globus Software: dev.globus.org



Incubator Projects

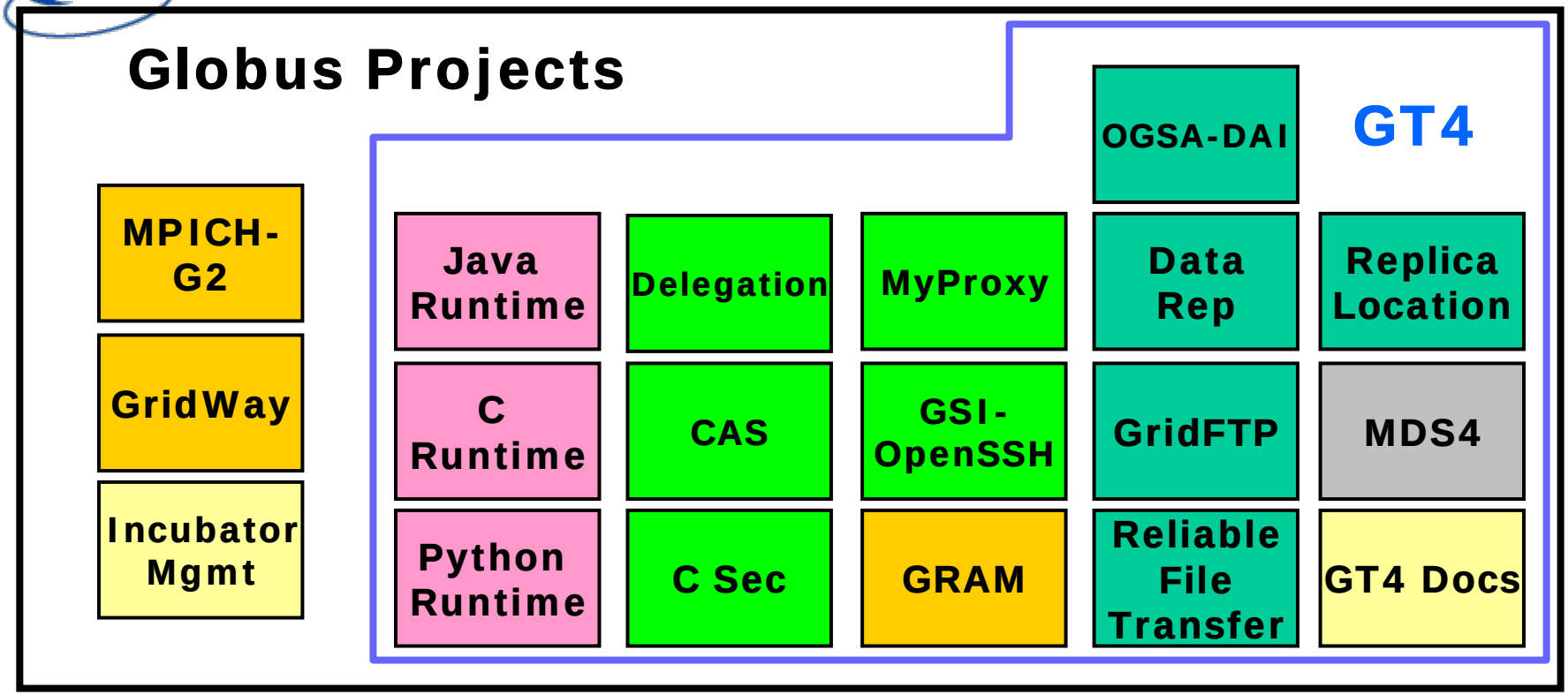
Common Runtime | Security | Execution Mgmt | Data Mgmt | Info Services | Other

Incubator Process in dev.globus

- Entry point for new Globus projects
- Incubator Management Project (IMP)
  - Oversees incubator process form first contact to becoming a Globus project
  - Quarterly reviews of current projects

http://dev.globus.org/wiki/Incubator/Incubator_Process

Globus Software: dev.globus.org

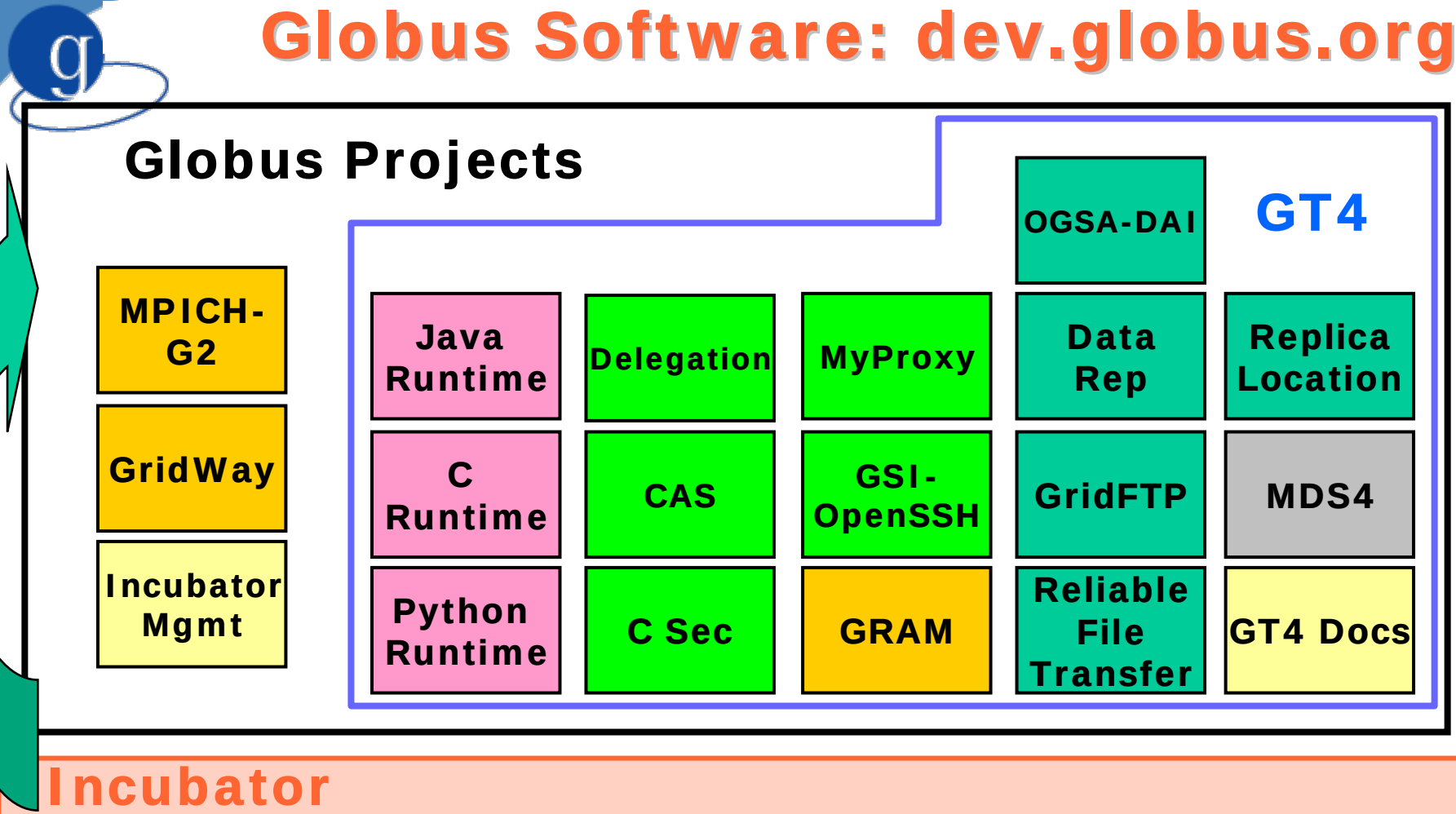


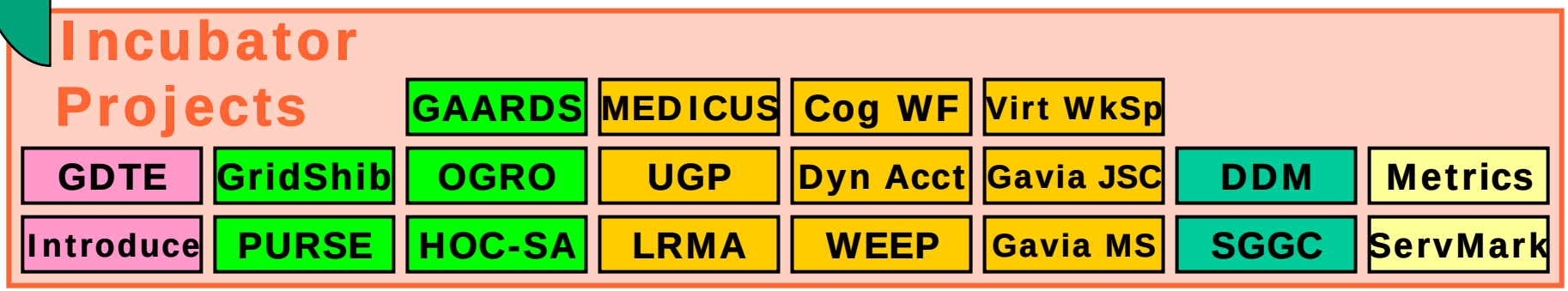
Incubator Projects

		GAARDS	MEDICUS	Cog WF	Virt WkSp		
GDTE	GridShib	OGRO	UGP	Dyn Acct	Gavia JSC	DDM	Metrics
Introduce	PURSE	HOC-SA	LRMA	WEEP	Gavia MS	SGGC	ServMark

Common Runtime | Security | Execution Mgmt | Data Mgmt | Info Services | Other

Globus Software: dev.globus.org





Common Runtime | Security | Execution Mgmt | Data Mgmt | Info Services | Other

Globus Software ... globus.org

Globus Projects

MPICH-G2

GridWay

Incubator Mgmt

GT4

			OGSA-DAI	
Java Runtime	Delegation	MyProxy	Data Rep	Replica Location
C Runtime	CAS	GSI-OpenSSH	GridFTP	MDS4
Python Runtime	C Sec	GRAM	Reliable File Transfer	GT4 Docs

Incubator Projects

		GAARDS	MEDICUS	Cog WF	Virt WkSp		
GDTE	GridShib	OGRO	UGP	Dyn Acct	Gavia JSC	DDM	Metrics
Introduce	PURSE	HOC-SA	LRMA	WEEP	Gavia MS	SGGC	ServMark

Common Runtime | Security | Execution Mgmt | Data Mgmt | Info Services | Other

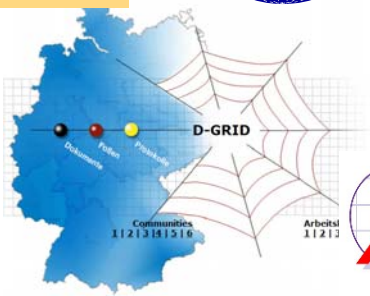
Globus User Community

- Large & diverse
  - 10s of national Grids, 100s of applications, 1000s of users; probably much more
  - Every continent except Antarctica
  - Applications ranging across many sciences
  - Dozens (at least) of commercial deployments
- Successful
  - Many production systems doing real work
  - Many applications producing real results
- Smart, energetic, demanding
  - Constant stream of new use cases & tools

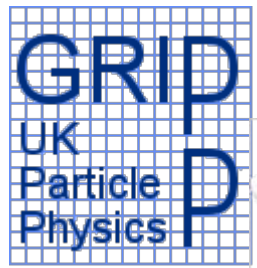
Global Community

grid.it project

K*GRID



National Grid
NG
SINGAPORE



CERN
openlab for DataGrid applications
Developing Solutions for the Data-Intensive Science of the Large Hadron Collider

Grid Consortium Japan

Examples of Production Scientific Grids

- APAC (Australia)
- China Grid
- China National Grid
- DGrid (Germany)
- EGEE
- NAREGI (Japan)
- Open Science Grid
- Taiwan Grid
- TeraGrid
- ThaiGrid
- UK Nat'l Grid Service

Open Science Grid

More Specifically, I May Want To …

- Create a service for use by my colleagues
- Manage who is allowed to access my service (or my experimental data or …)
- Ensure reliable & secure distribution of data from my lab to my partners
- Run 10,000 jobs on whatever computers I can get hold of
- Monitor the status of the different resources to which I have access

Summary So Far

- General Grid definition
- Why we need Globus
- Some basic use cases

More Specifically, I May Want To …

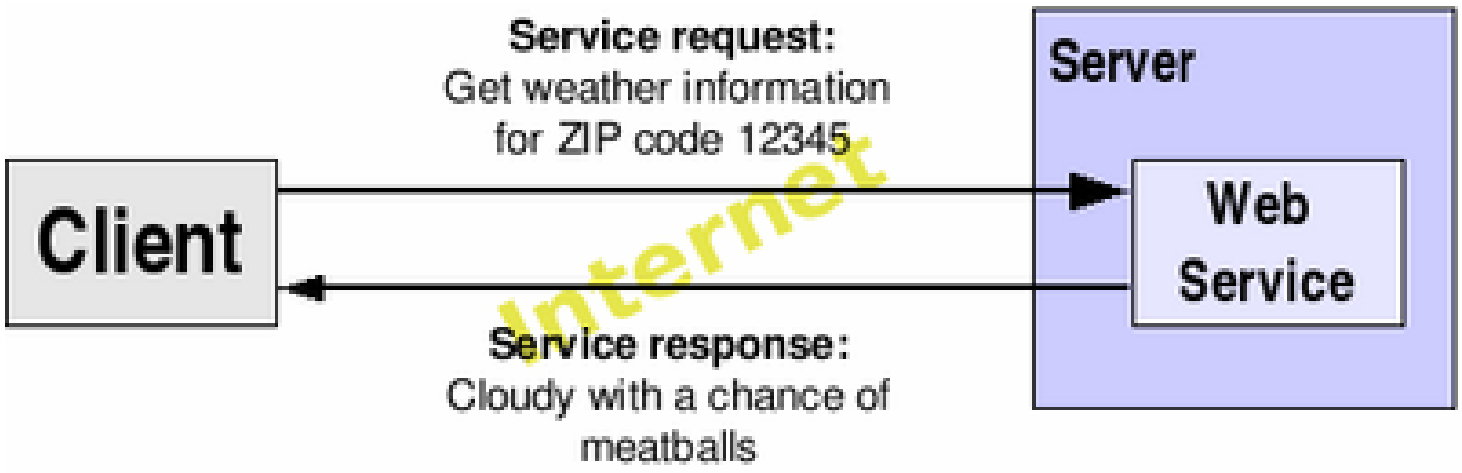
- Create a service for use by my colleagues
- Manage who is allowed to access my service (or my experimental data or …)
- Ensure reliable & secure distribution of data from my lab to my partners
- Run 10,000 jobs on whatever computers I can get hold of
- Monitor the status of the different resources to which I have access

More Specifically, I May Want To …

- Create a service for use by my colleagues
  - What is a Web service?
  - What tools does Globus have to support this?

Web Service Basics

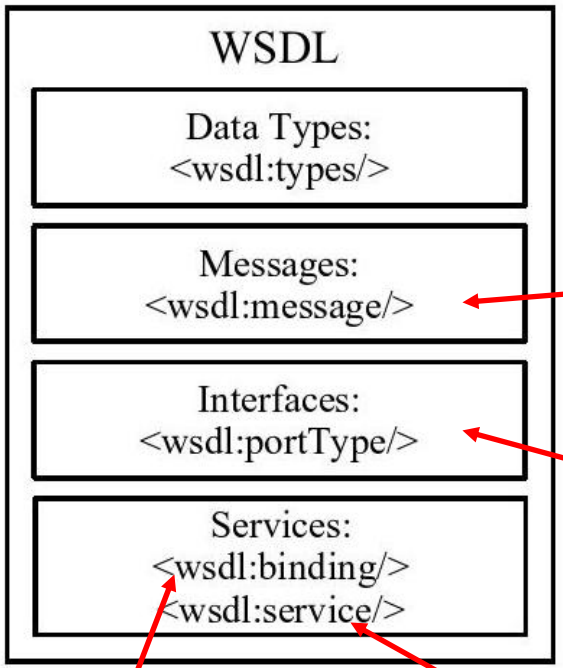
- Web Services are basic distributed computing technology that let us construct client-server interactions



Web Service Basics 2

- Web services are platform independent and language independent
  - Client and server program can be written in diff langs, run in diff envt's and still interact
- Web services describe themselves
  - Once located you can ask it how to use it
- Web service is *not* a website
  - Web service is accessed by sw, not humans
- Web services are ideal for loosely coupled systems
  - Unlike CORBA, EJB, etc.

WSDL: Web Services Description Language



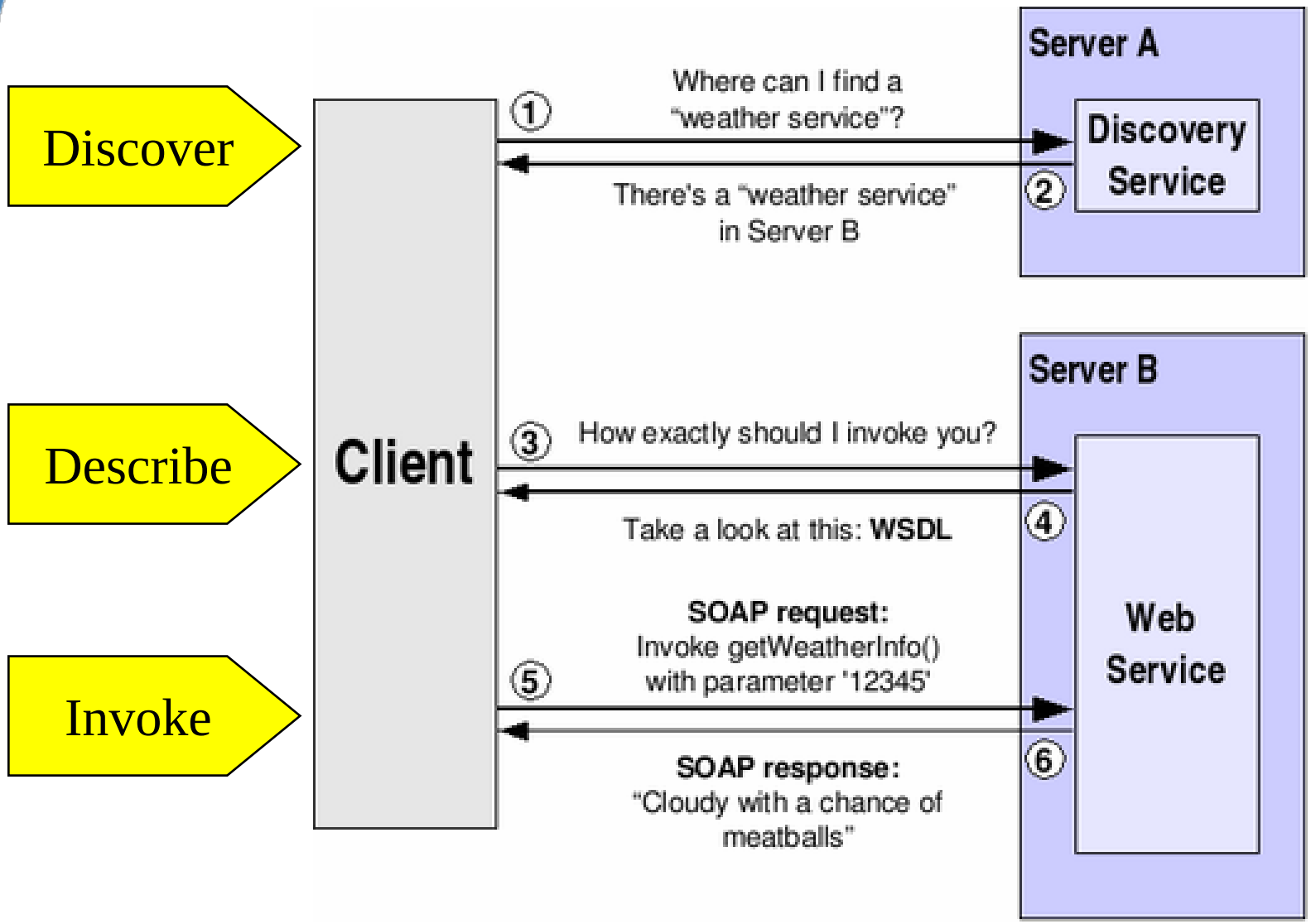
Define expected messages for a service, and their (input or output parameters)

An interface groups together a number of messages (operations)

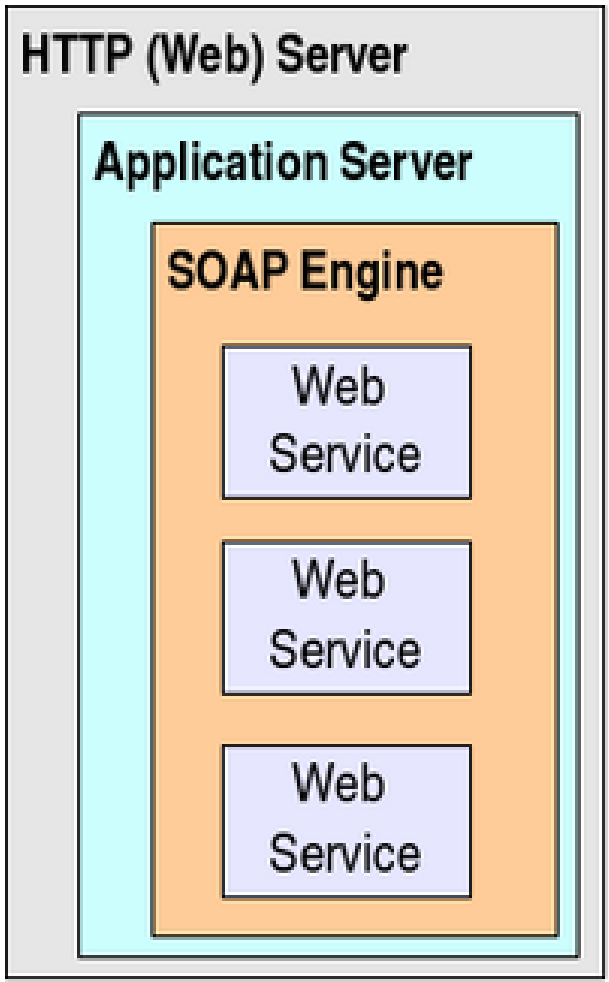
Bind an Interface via a definition to a specific transport (e.g. HTTP) and messaging (e.g. SOAP) protocol

The network location where the service is implemented , e.g. http://localhost:8080

Real Web Service Invocation



Web Services Server Applications

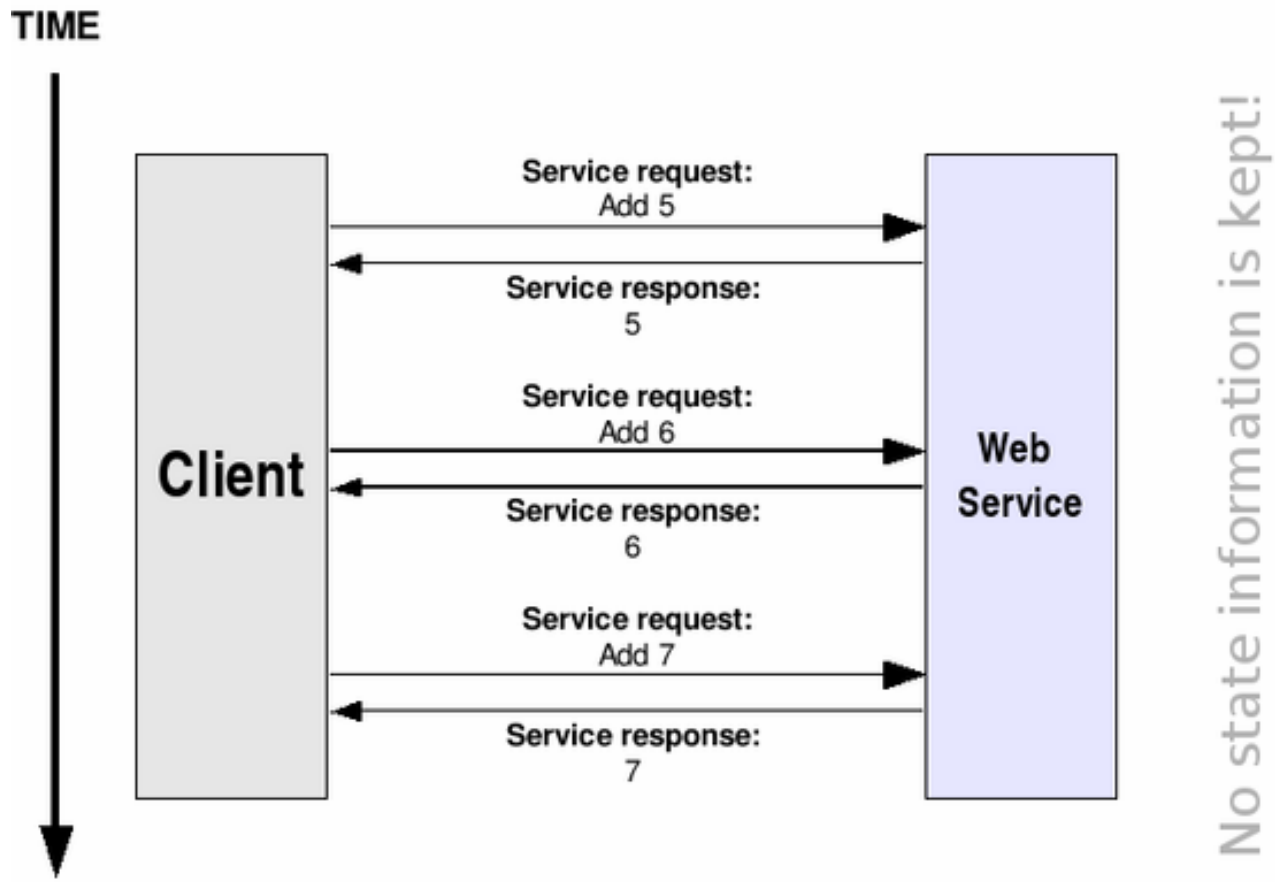


- **Web service** – software that exposes a set of operations
- **SOAP Engine** – handle SOAP requests and responses (Apache Axis)
- **Application Server** – "living space" for app that must be accessed by different clients (Tomcat)
- **HTTP server-** also called a Web server, handles http messages

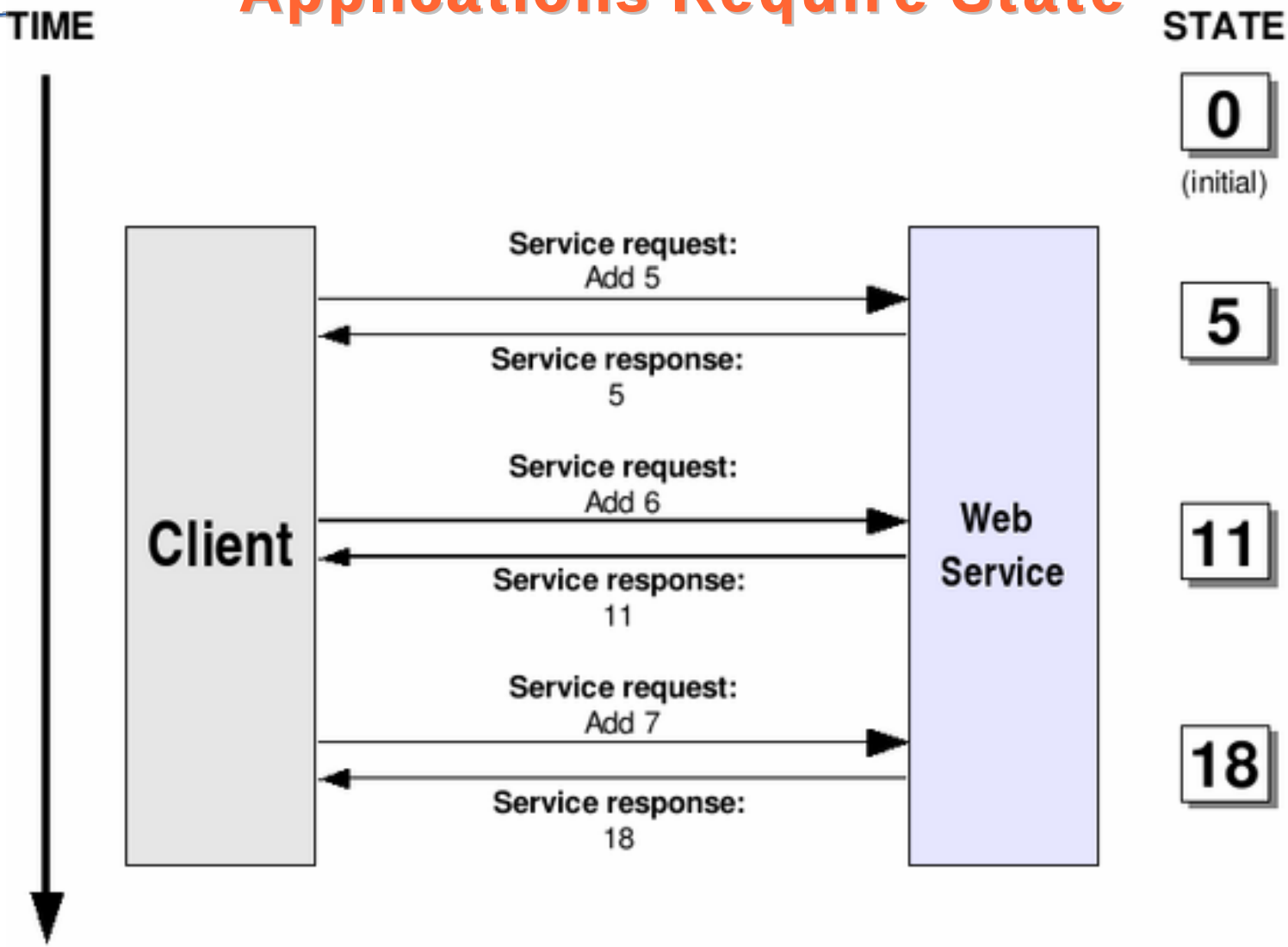
Container

Let's talk about state

- Plain Web services are stateless

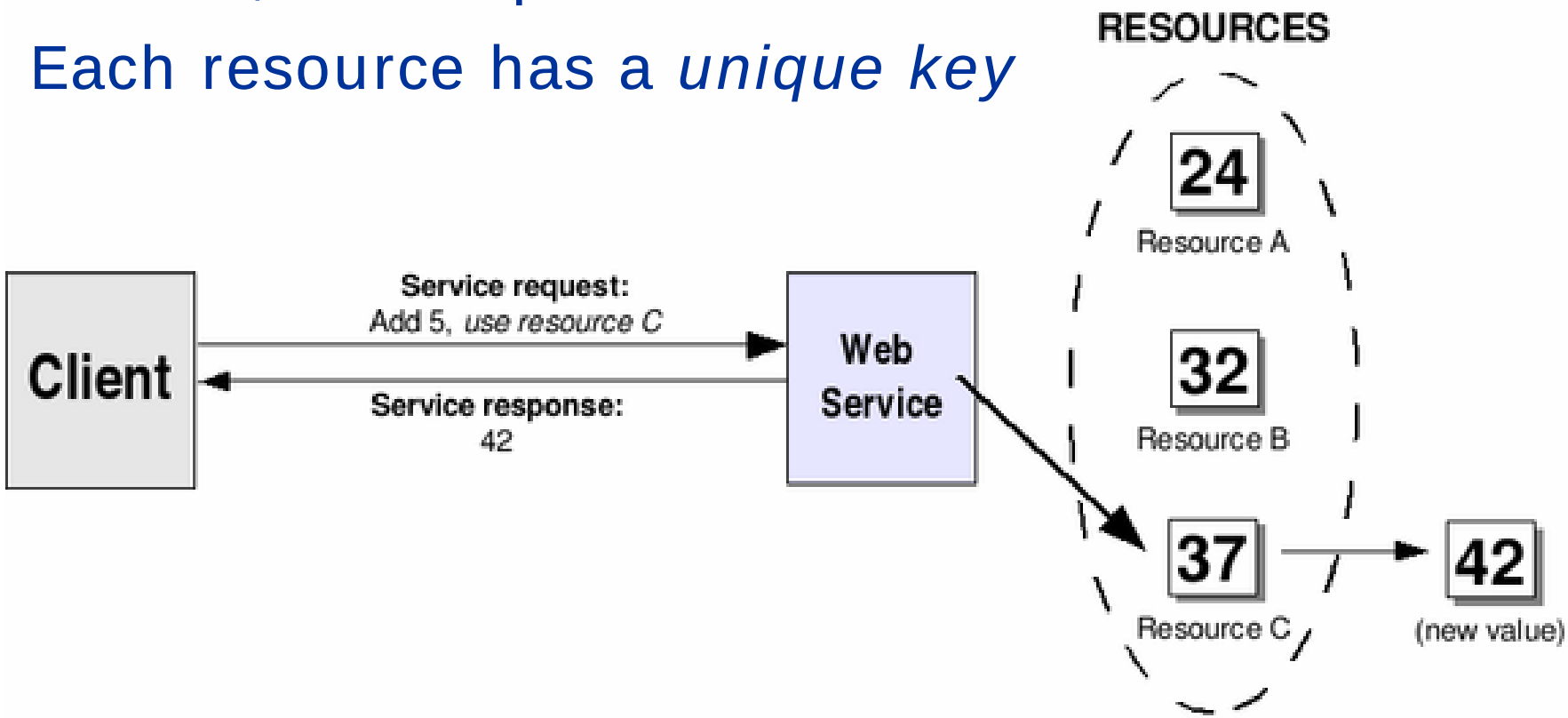


However, Many Grid Applications Require State

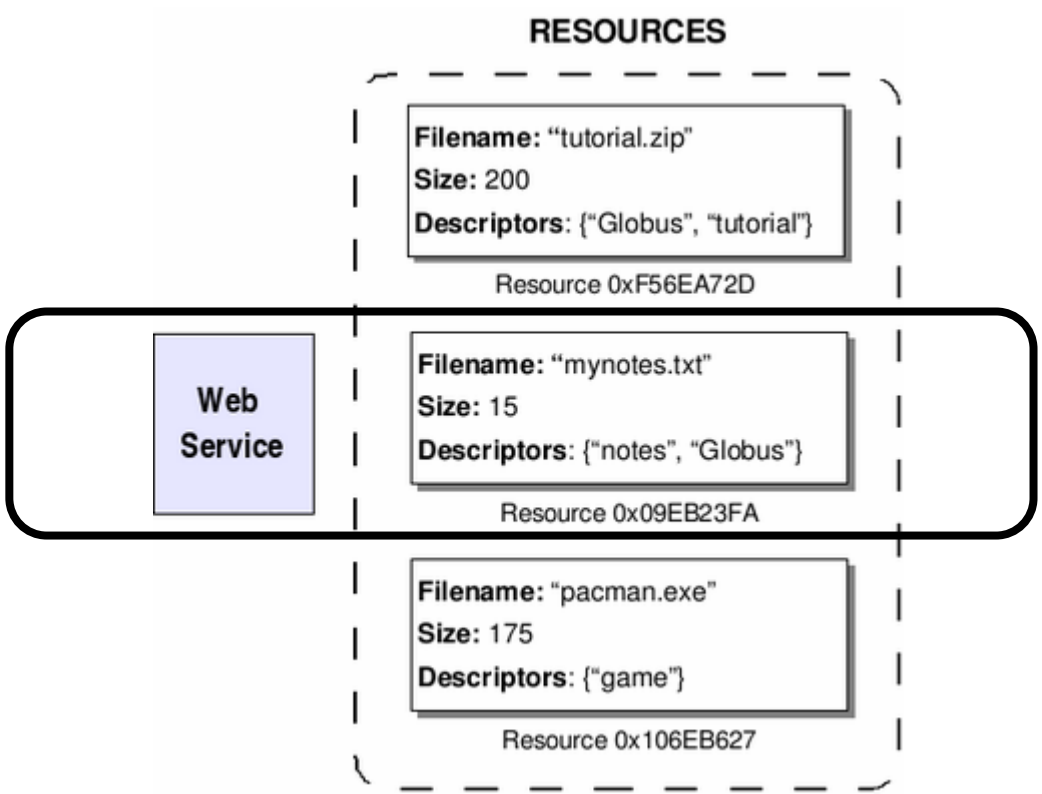


Keep the Web Service and the State Separate

- Instead of putting state *in* a Web service, we keep it in a *resource*
- Each resource has a *unique key*



Resources Can Be Anything Stored



Web Service
+
Resource
=
WS-Resource

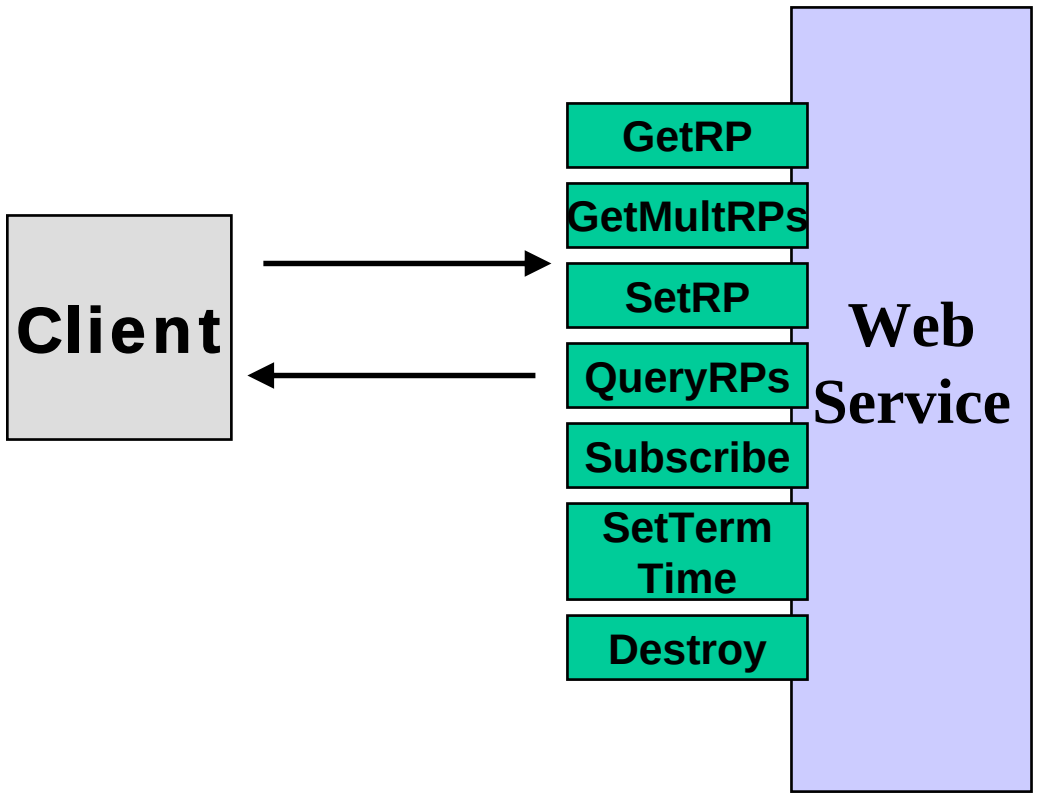
Address of a WS-resource is called an *end-point reference*

Web Services So Far

- Basic client-server interactions
- Stateless, but with associated resources
- Self describing using WSDL

- But we'd really like is a common way to
  - Name and do bindings
  - Start and end services
  - Query, subscription, and notification
  - Share error messages

Standard Interfaces



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

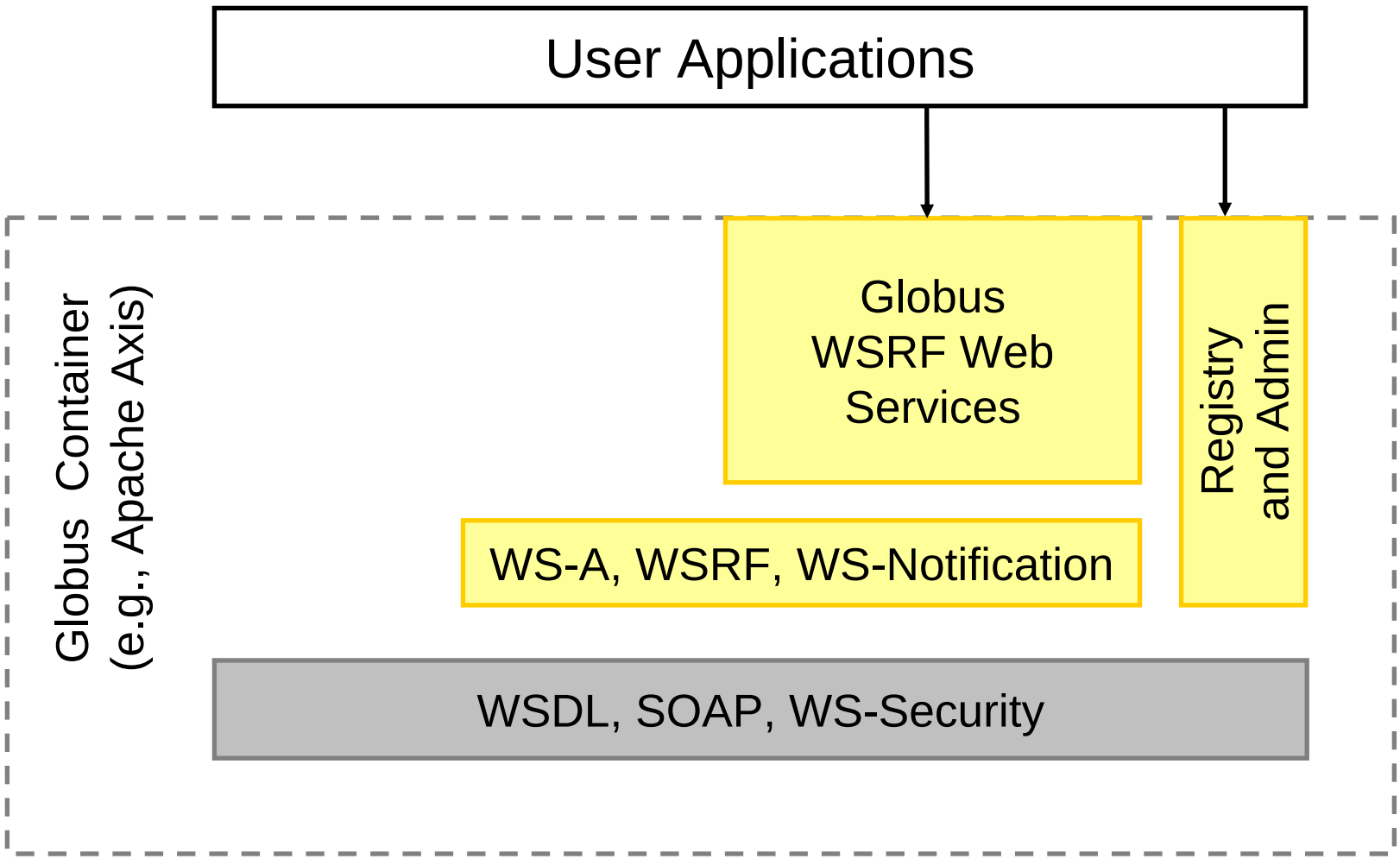
WSRF & WS-Notification

- Naming and bindings (basis for virtualization)
  - Every resource can be uniquely referenced, and has one or more associated services for interacting with it
- Lifecycle (basis for fault resilient state management)
  - Resources created by services following factory pattern
  - Resources destroyed immediately or scheduled
- Information model (basis for monitoring & discovery)
  - Resource properties associated with resources
  - Operations for querying and setting this info
  - Asynchronous notification of changes to properties
- Service Groups (basis for registries & collective svcs)
  - Group membership rules & membership management
- Base Fault type

WSRF vs XML/SOAP

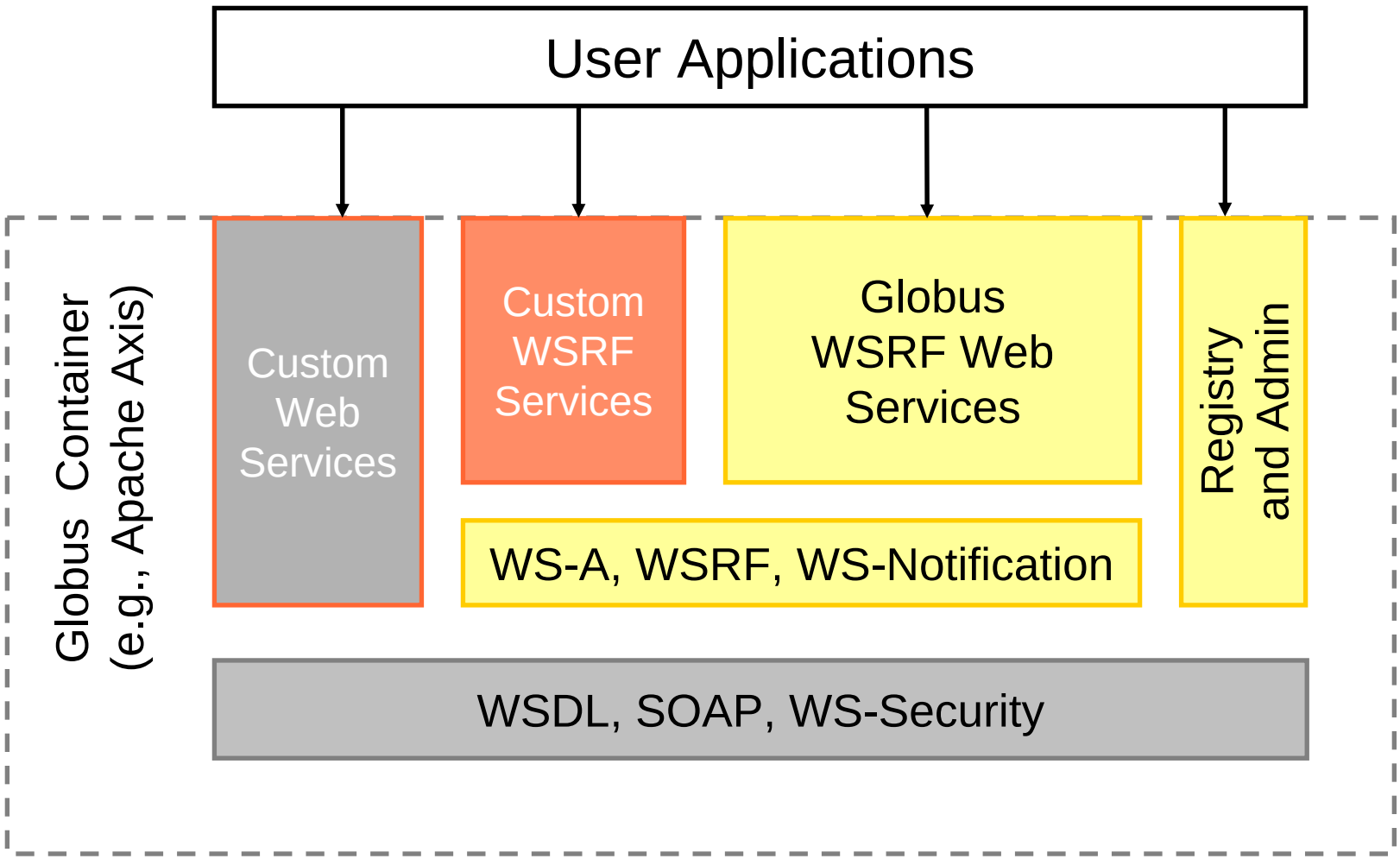
- The definition of WSRF means that the Grid and Web services communities can move forward on a common base
- Why Not Just Use XML/SOAP?
  - WSRF and WS-N *are* just XML and SOAP
  - WSRF and WS-N are just Web services
- Benefits of following the specs:
  - These patterns represent best practices that have been learned in many Grid applications
  - There is a community behind them
  - Why reinvent the wheel?
  - Standards facilitate interoperability

Globus and Web Services



Globus Core: Java , C (fast, small footprint), Python

Globus and Web Services

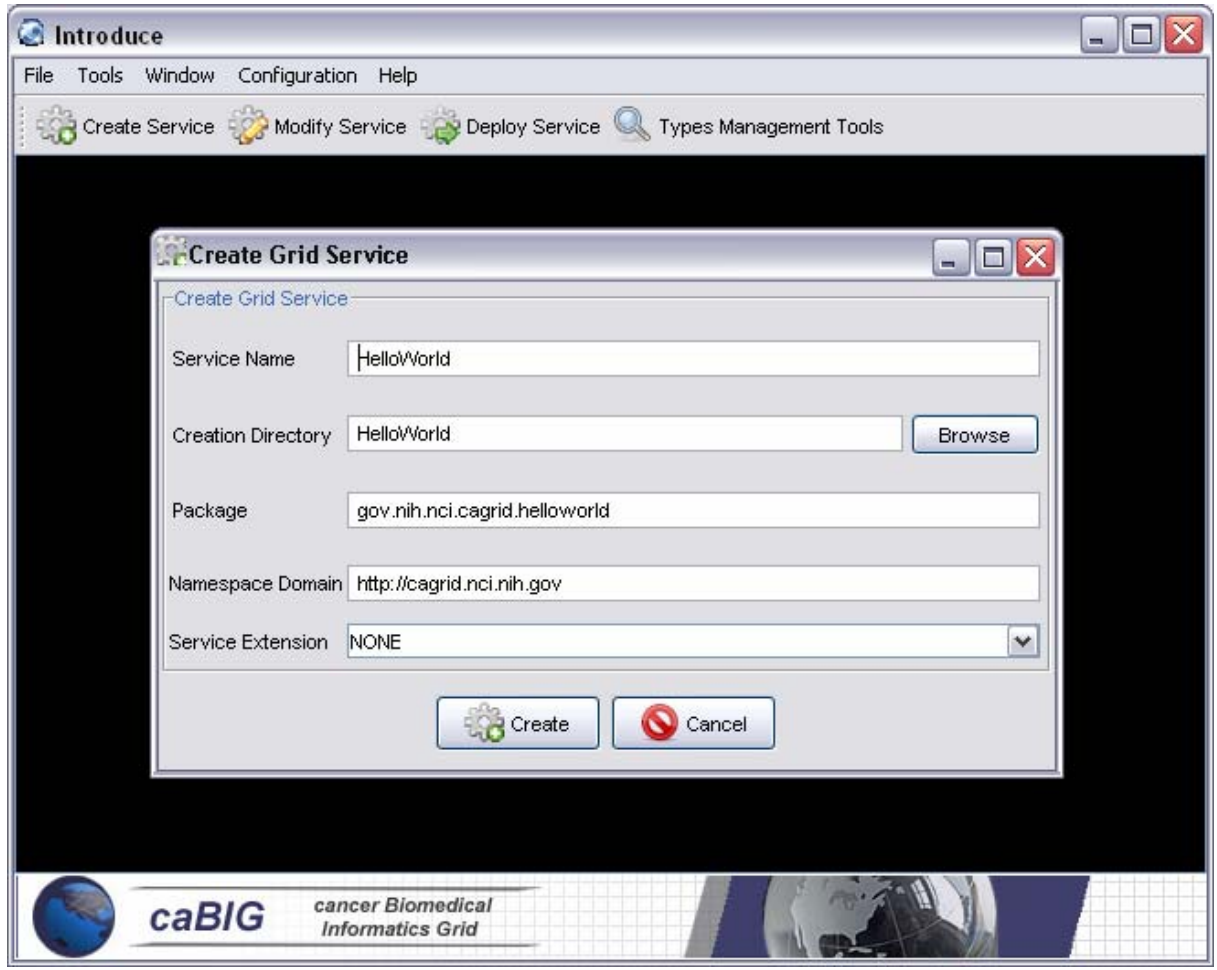


Globus Core: Java , C (fast, small footprint), Python

the globus alliance
www.globus.org

The Introduce Authoring Tool

- Define service
- Create skeleton
- Discover types
- Add operations
- Configure security
- Modify service



See also: SOAPLab, OPAL, pyGlobus, Gannon, etc.

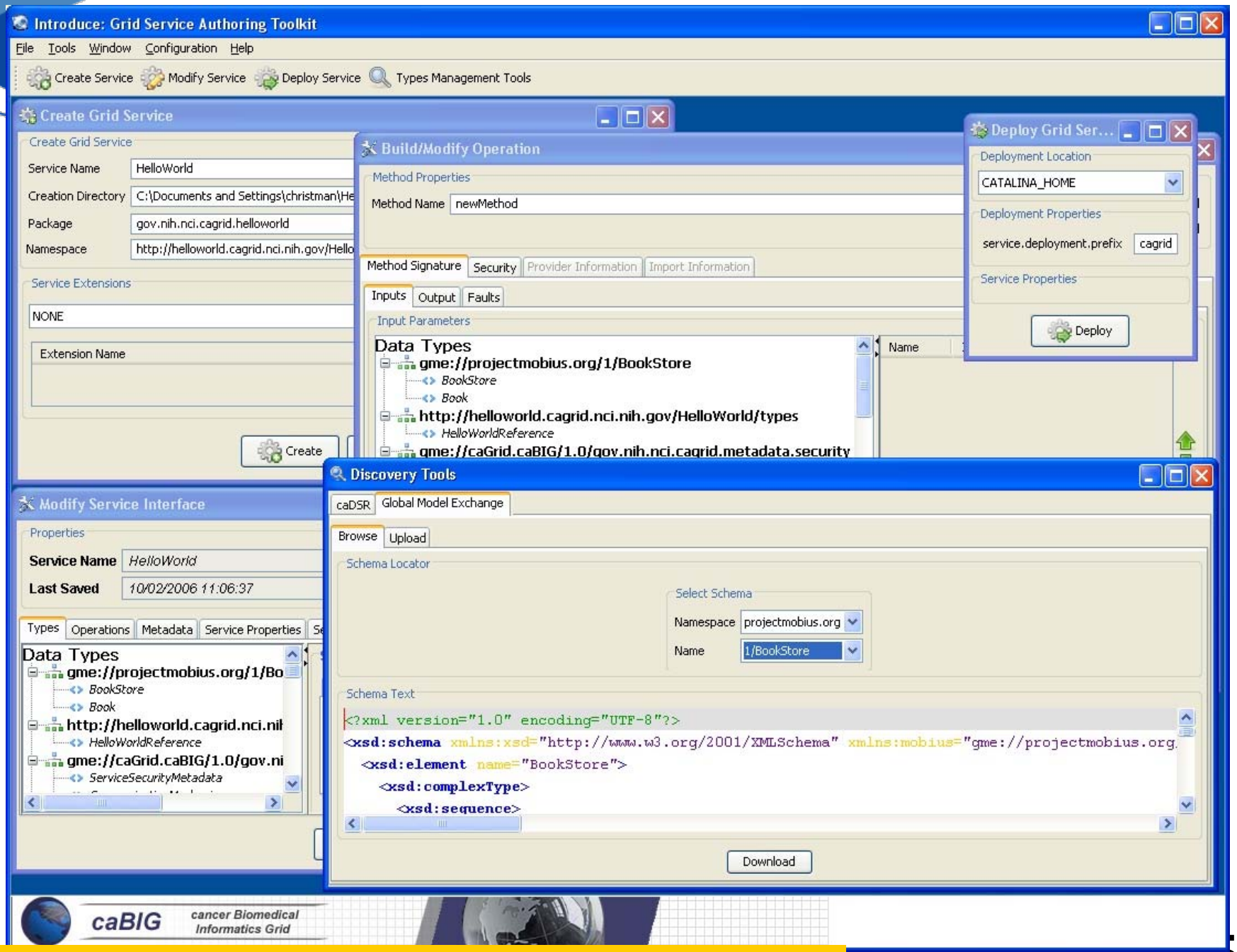
Introduce: Hastings, Saltz, et al., Ohio State University

Generated Service Features

- Dynamic discovery and use of published data types
- Creates WSDL2.0 / WSRF compliant services
- Supports creating multiple resource/services using the Web Service Resource Framework (WSRF)
- Globus GSI Security Configuration
- Grid Map and GridGrouper Authorization Support
- Resource Property configuration and Index service registration
- Rich extension/plug-in framework for creating custom services or adding custom functionality to Introduce

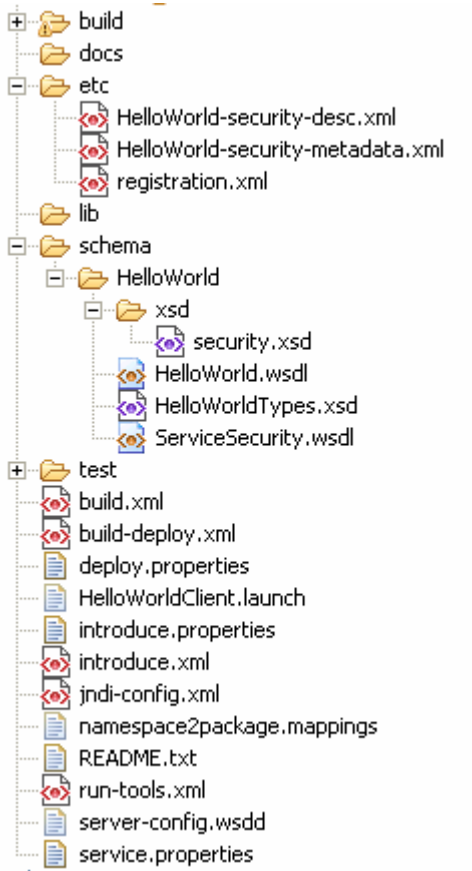
Introduce Toolkit Architecture

- Service Model maintains the complete representation of the services, methods, resources, resource properties, data types, security configuration. Index service registration, etc.
- Graphical Development Environment modifies service model and passes the modified model to the engine
- Introduce Engine re-syncs the service with changes with the modified model
- Use of Java Emitter Templates (JET) for creation and modification of source code and configuration files required in the service
- Auto generation of unboxed complete client and stubbed service implementation

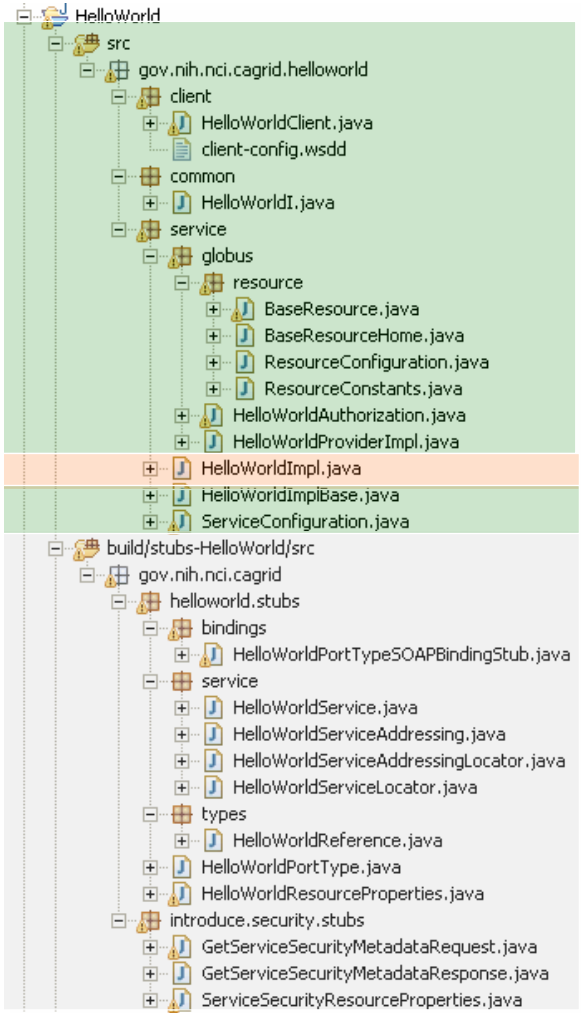


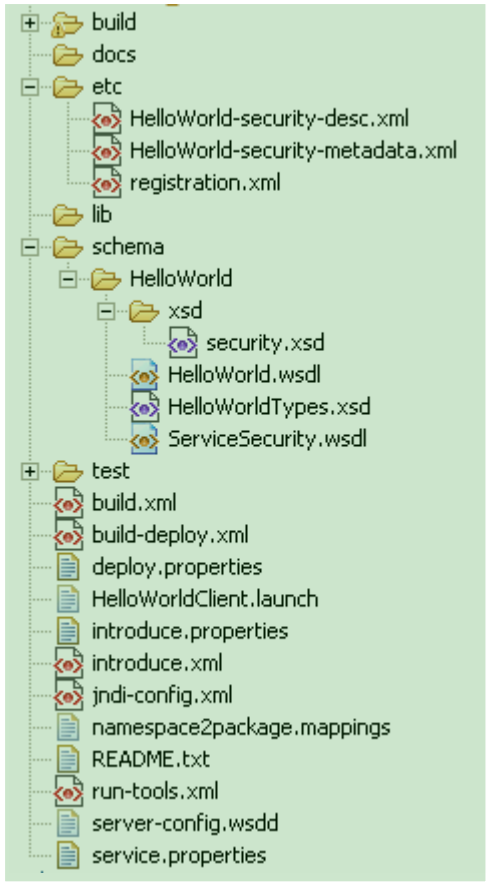
Shannon Hastings, http://dev.globus.org/wiki/Incubator/Introduce

Generated Service Skeleton



Generated Service Skeleton





= introduce generated

= globus/axis generated

= developers contribution

Summary so far

- Introduction to Web services
  - Underlying need for standards
- What Globus Core gives you in C, Java and Python
- Introduce for easy service development

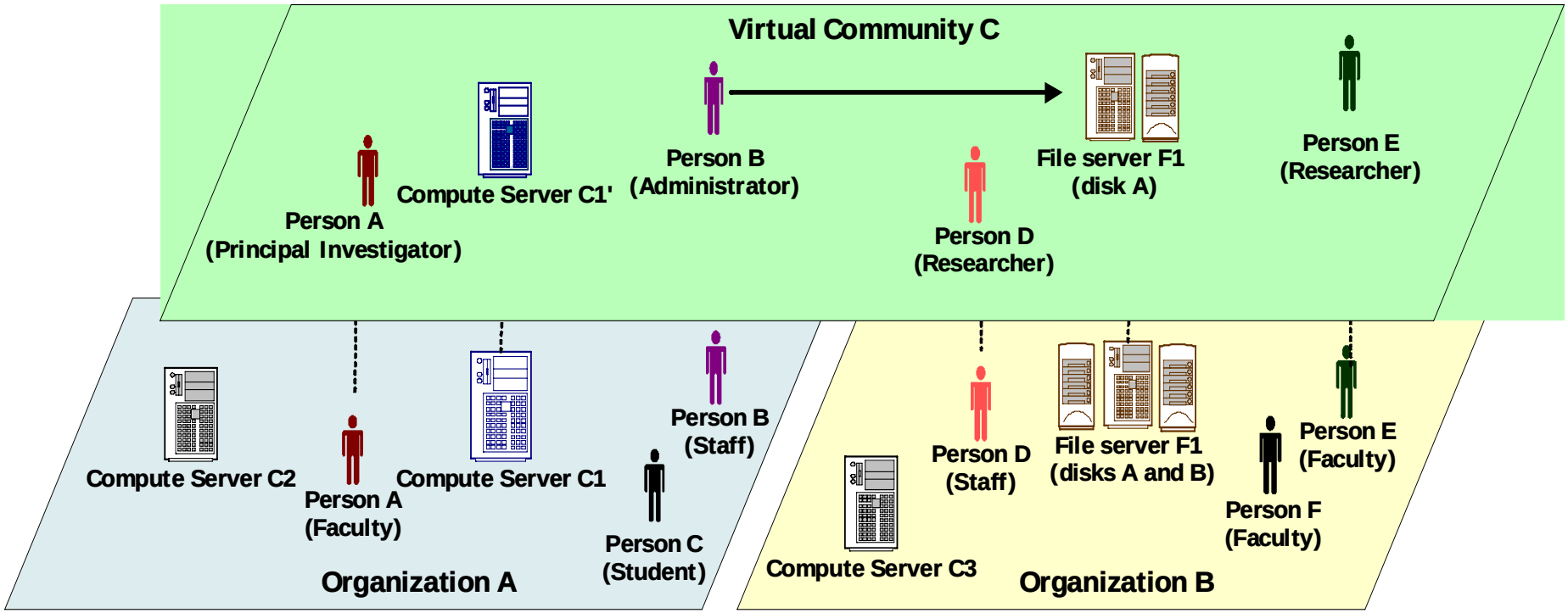
More Specifically, I May Want To …

- Create a service for use by my colleagues
- Manage who is allowed to access my service (or my experimental data or …)
- Ensure reliable & secure distribution of data from my lab to my partners
- Run 10,000 jobs on whatever computers I can get hold of
- Monitor the status of the different resources to which I have access

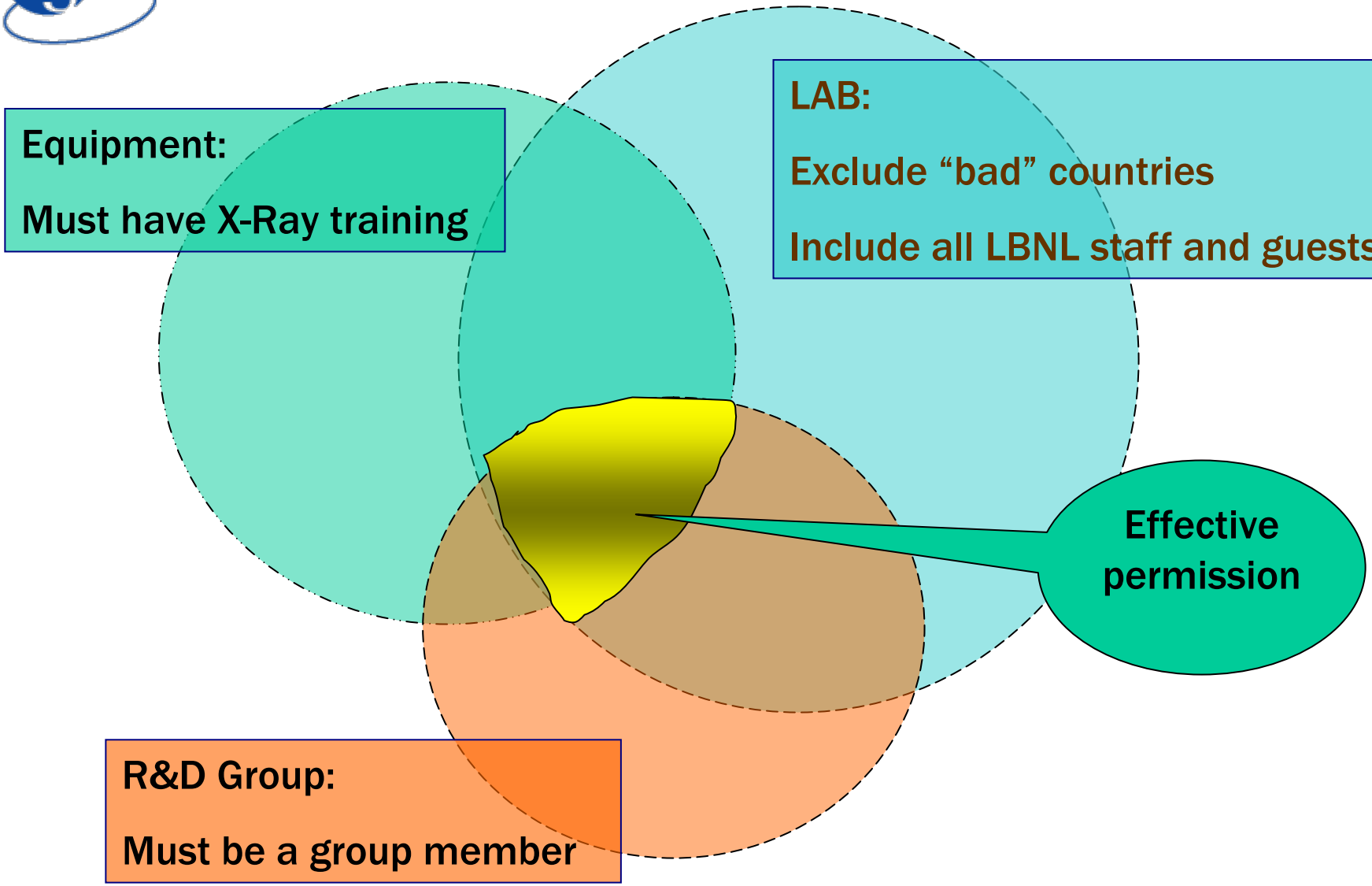
Grid Security Concerns

- 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

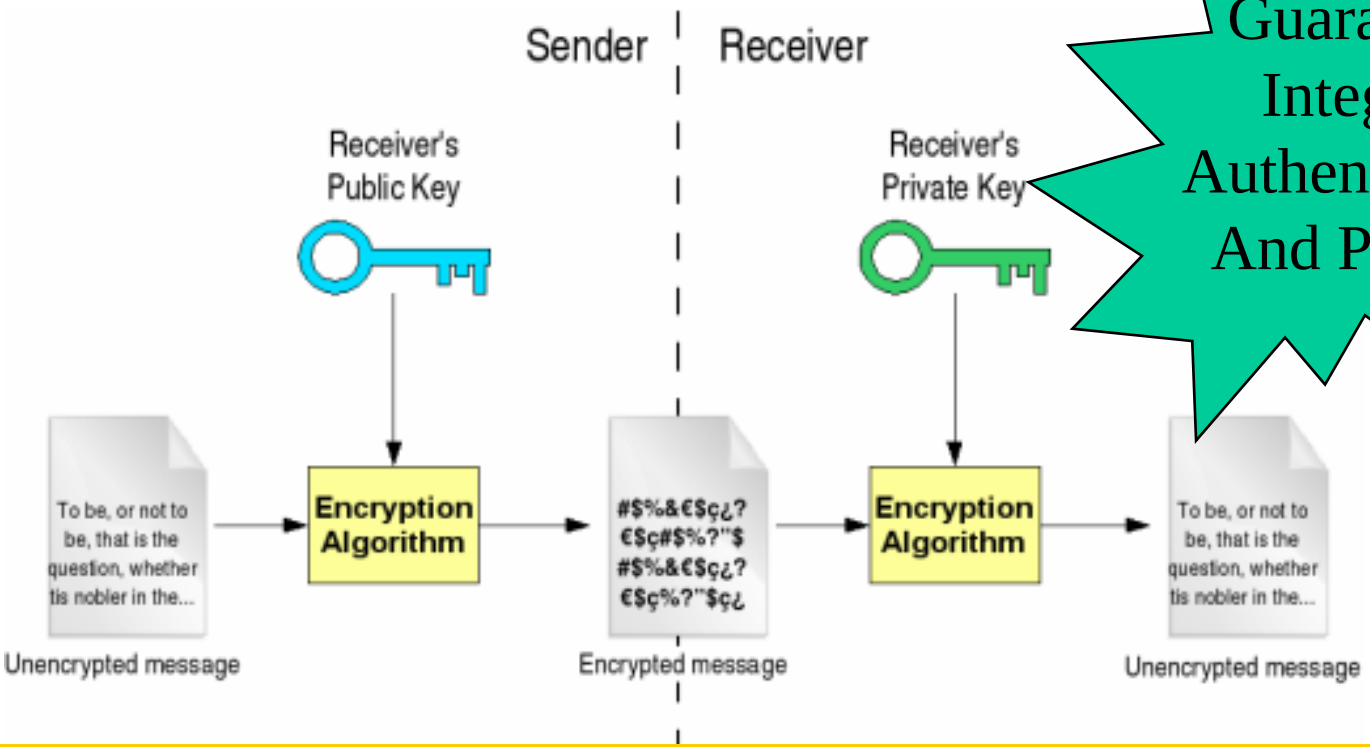

Equipment:
Must have X-Ray training
LAB:
Exclude “bad” countries
Include all LBNL staff and guests
Effective permission
R&D Group:
Must be a group member

Security Basics

- Privacy
  - Only the sender and receiver should be able to understand the conversation
- Integrity
  - Receiving end must know that the received message was the one from the sender
- Authentication
  - Users are who they say they are (authentic)
- Authorization
  - Is user allowed to perform the action

Authentication

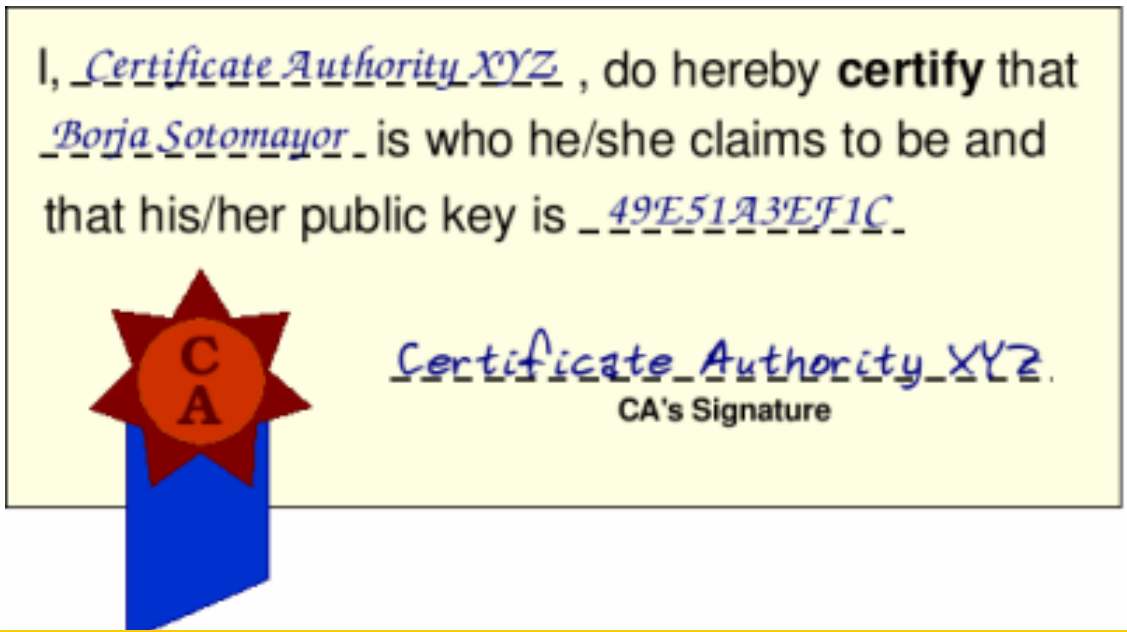
- Private Key - known only by owner
- Public Key- known to everyone
- What one key encrypts, the other de[illegible]ts



Guarantees
Integrity
Authentication
And Privacy

Authentication using Digital Certificates

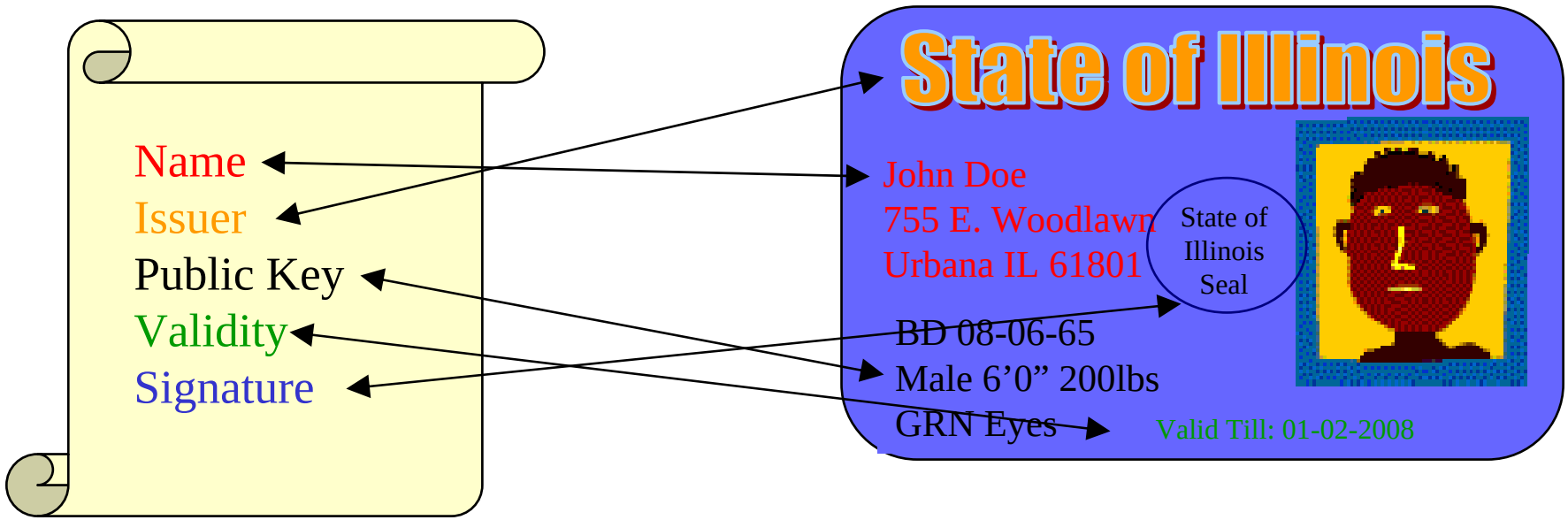
- Digital document that certifies a public key is owned by a particular user
- Signed by 3rd party – the Certificate Authority (CA)



To know if you should trust the certificate, you just have to trust the CA

Certificates

- Similar to passport or driver's license

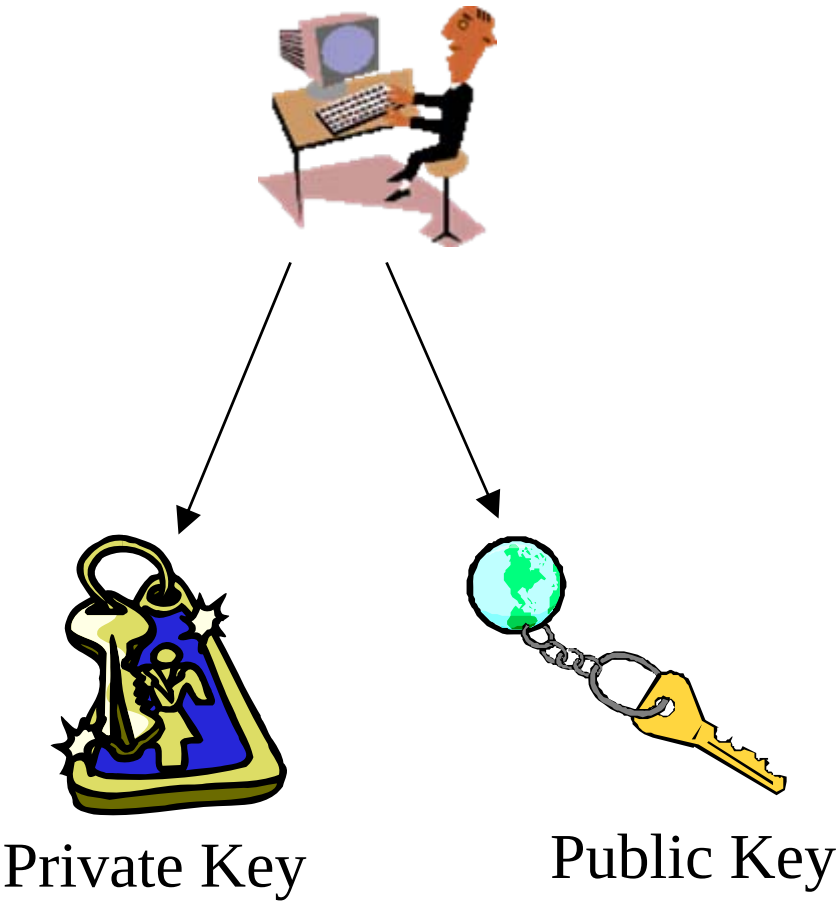


Globus Security

- Globus security is based on the Grid Security Infrastructure (GSI)
  - Set of IETF standards for security interaction
- Public-key-based authentication using X509 certificates

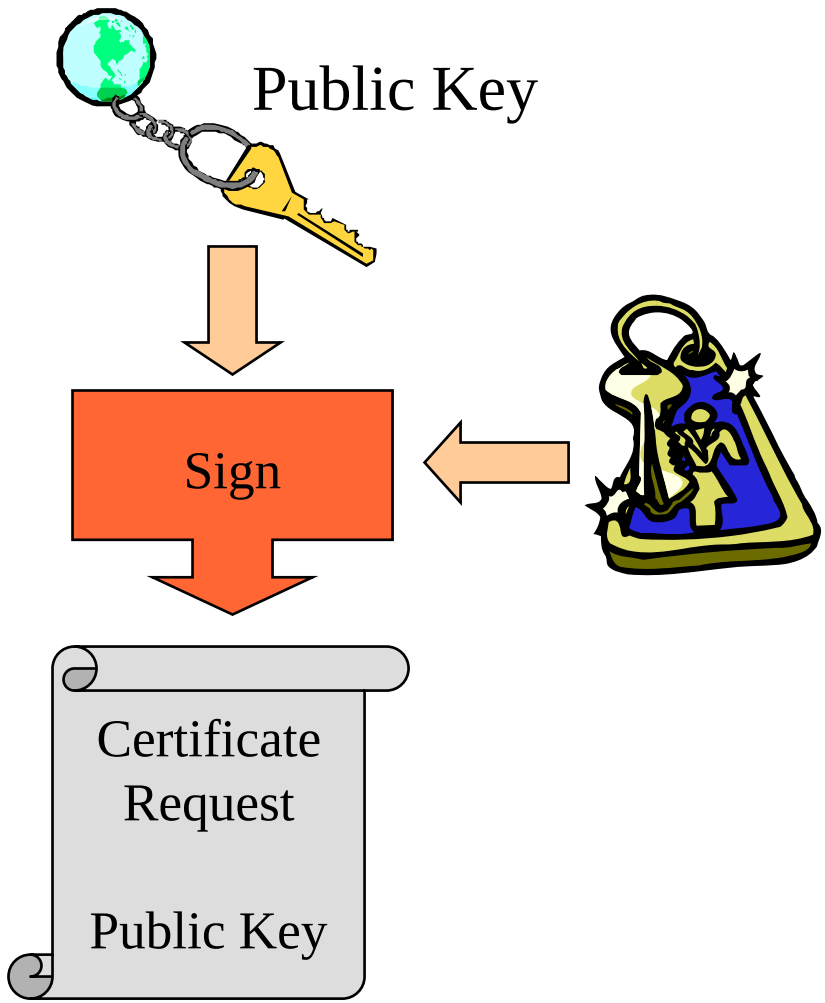
Requesting a Certificate

- To request a certificate a user starts by generating a key pair



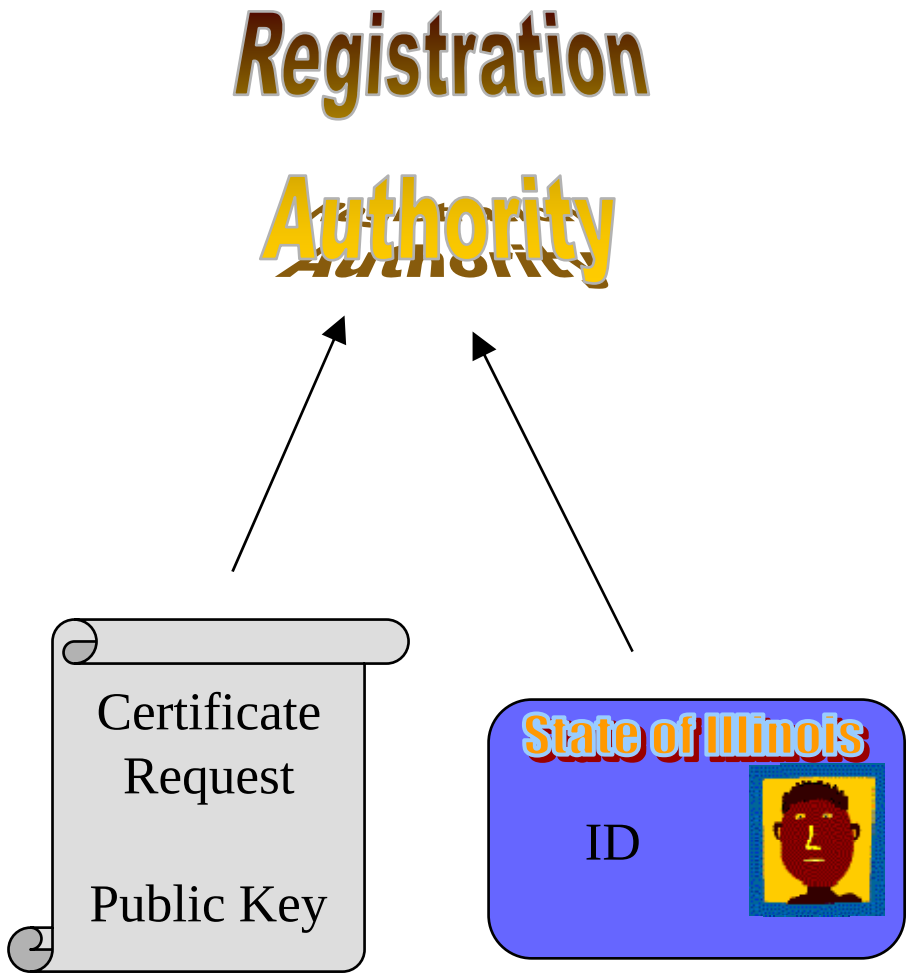
Certificate Request

- The user signs their own public key to form what is called a Certificate Request
- Email/Web upload
- Note private key is never sent anywhere



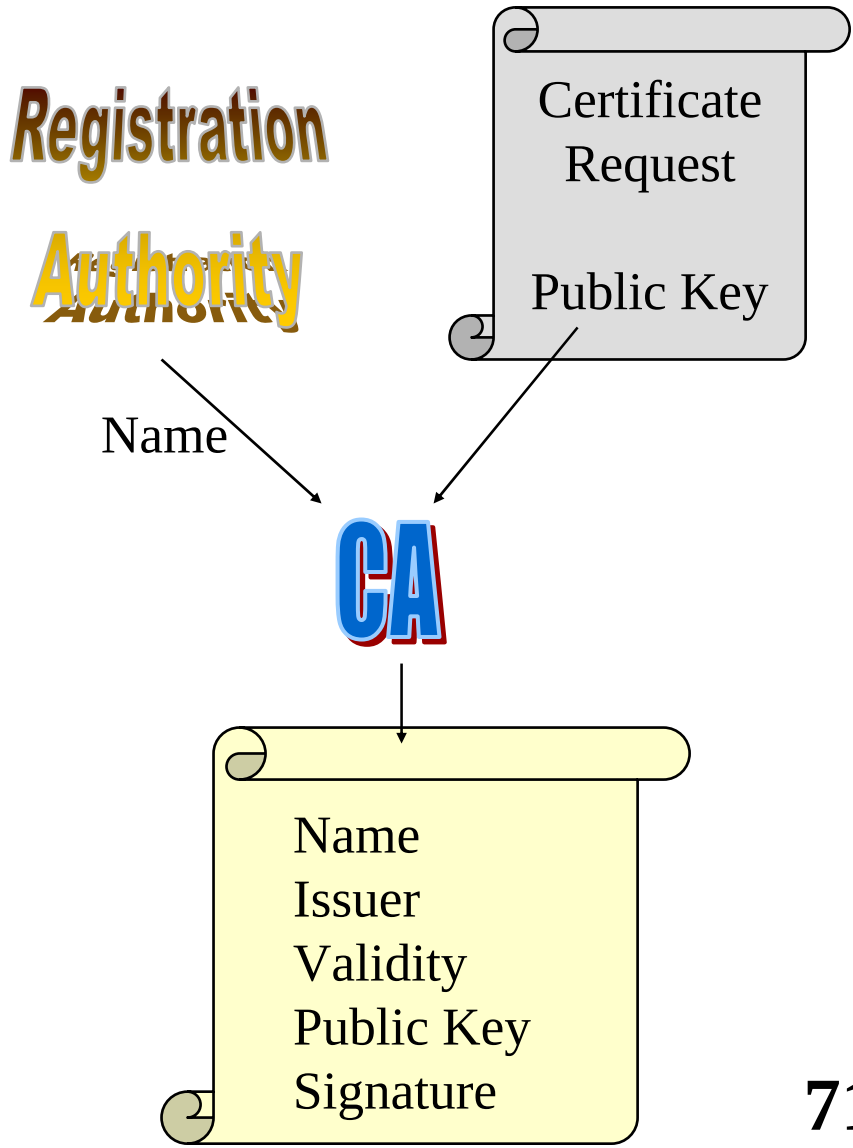
Registration Authority (RA)

- The user then takes the certificate to a Registration Authority (RA)
- Vetting of user's identity
- Often the RA coexists with the CA and is not apparent to the user



Certificate Issuance

- The CA then takes the identity from the RA and the public key from the certificate request
- It then creates, signs and issues a certificate for the user



Globus Security Tools

- Basic Grid Security Mechanisms
- Certificate Generation Tools
- Certificate Management Tools
  - Getting users “registered” to use a Grid
  - Getting Grid credentials to wherever they’re needed in the system
- Authorization/Access Control Tools
  - Storing and providing access to system-wide authorization information

Simple CA

- An online service that issues low-quality GSI certificates
  - Intended for people who want to experiment with Grid components that require certificates but do not have any other means of acquiring certificates
  - *Not* to be used on production systems
- Not a true Certificate Authority (CA)
  - No revoking or reissuing certificates
  - No verification of identities
  - The service itself is not especially secure

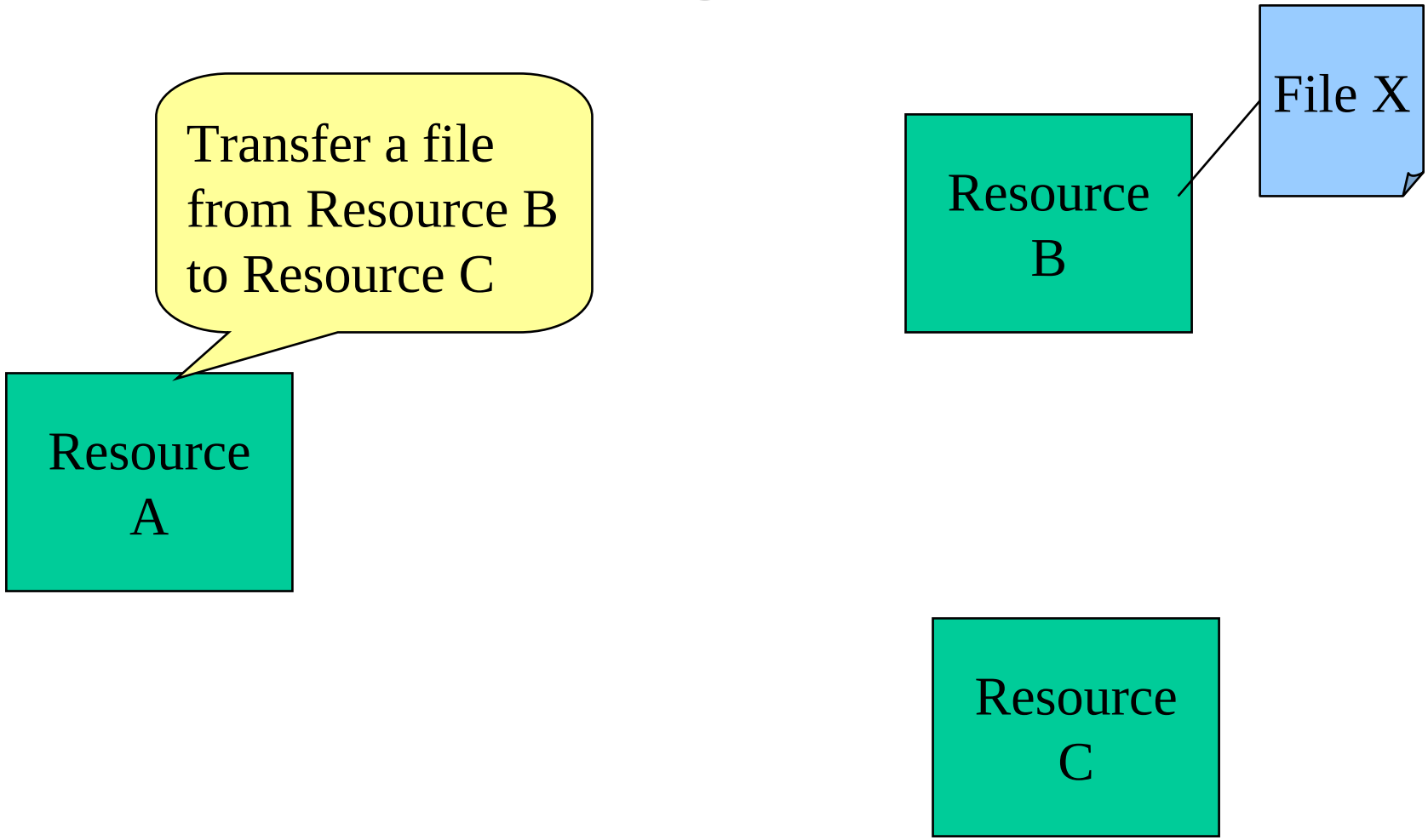
Simple CA (2)

- Most production Grids will not accept certificates that are not signed by a well-known CA
- Certificates generated by Simple CA will usually not be sufficient to gain access to production services

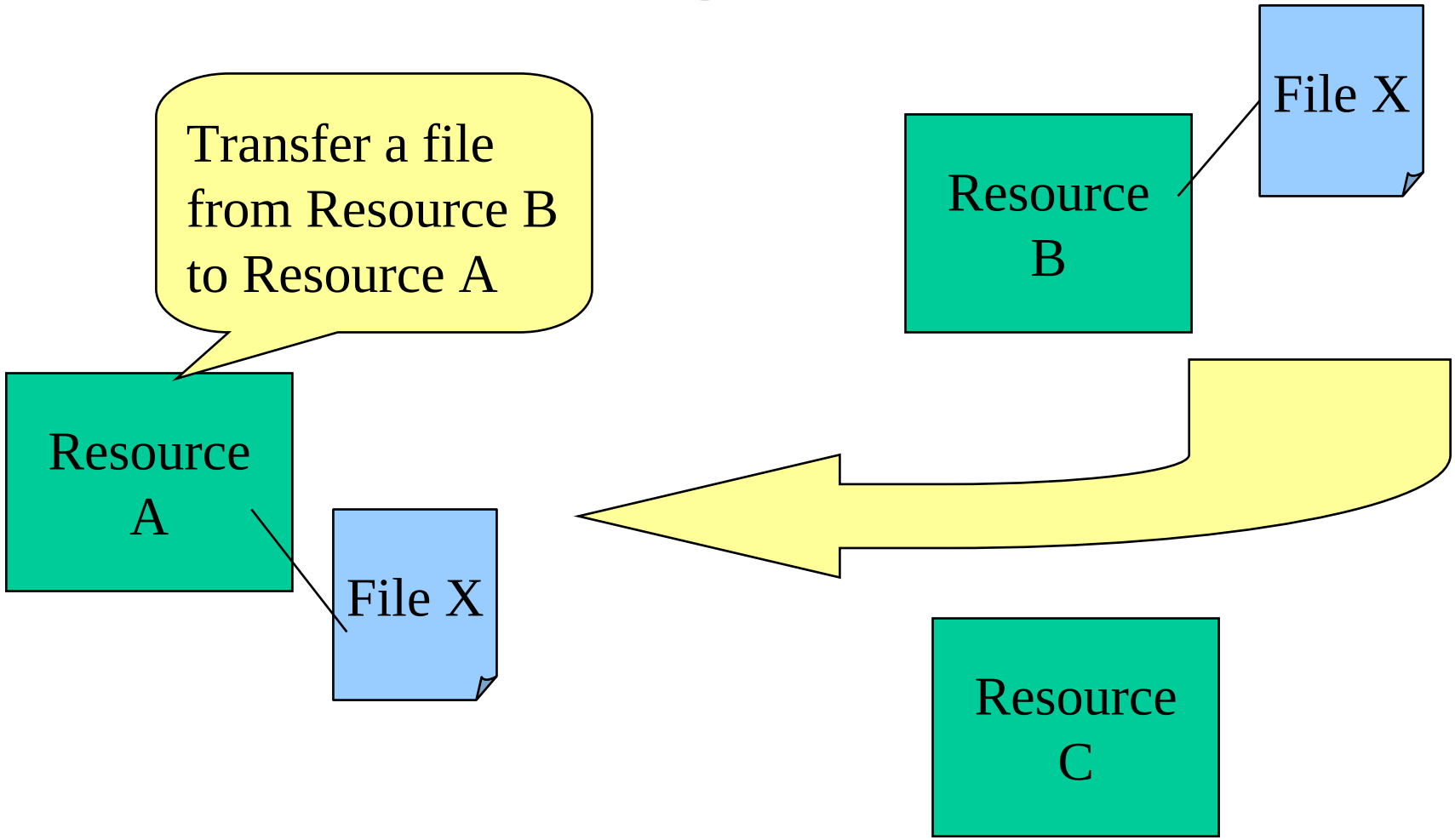
GridMap File

- Maps distinguished names (found in certificates) to local names (such as login accounts)
  - schopf@mcs.anl.gov
  - jms@nesc.ed.ac.uk
  - u11270@sdsc.edu
- Can also serve as a access control list for GSI enabled services

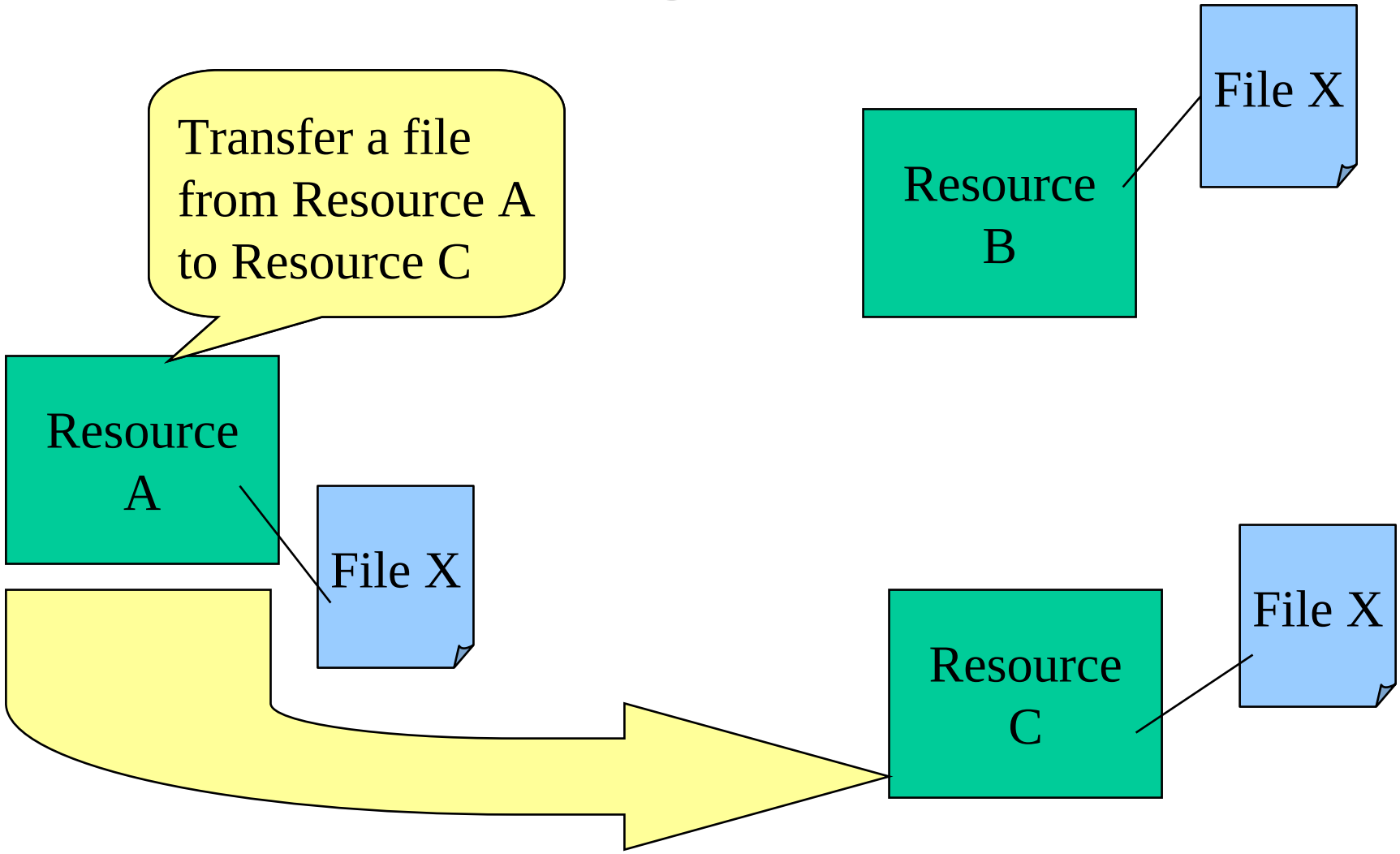
Delegation


Transfer a file from Resource B to Resource C
Resource A
Resource B
File X
Resource C

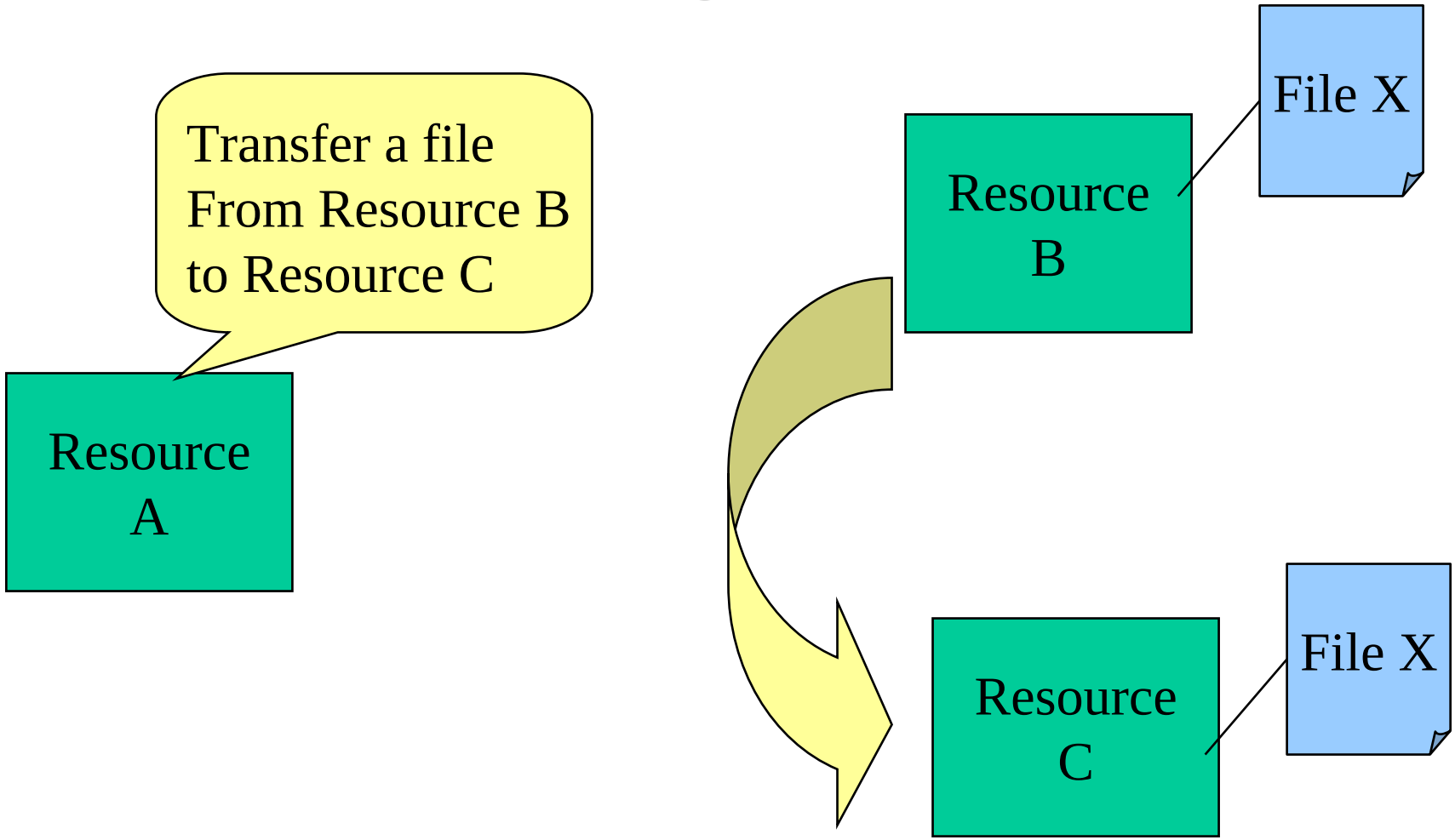
Delegation


Transfer a file from Resource B to Resource A
Resource A
File X
Resource B
File X
Resource C

Delegation


Transfer a file from Resource A to Resource C
Resource B
File X
Resource A
File X
Resource C
File X

Delegation


Transfer a file
From Resource B
to Resource C
Resource
A
Resource
B
File X
Resource
C
File X

Proxy Certificate

- Proxy Certificate allows another user to act upon their behalf
  - Credential delegation

Proxy Certificate

- Proxy empowers 3rd party to act upon your behalf
- Proxy certificate is signed by the end user, not a CA
- Proxy cert's public key is a new one from the private-public key pair generated specifically for the proxy certificate
- Proxy also allows you to do single sign-on
  - Setup a proxy for a time period and you don't need to sign in again

Benefits of Single Sign-on

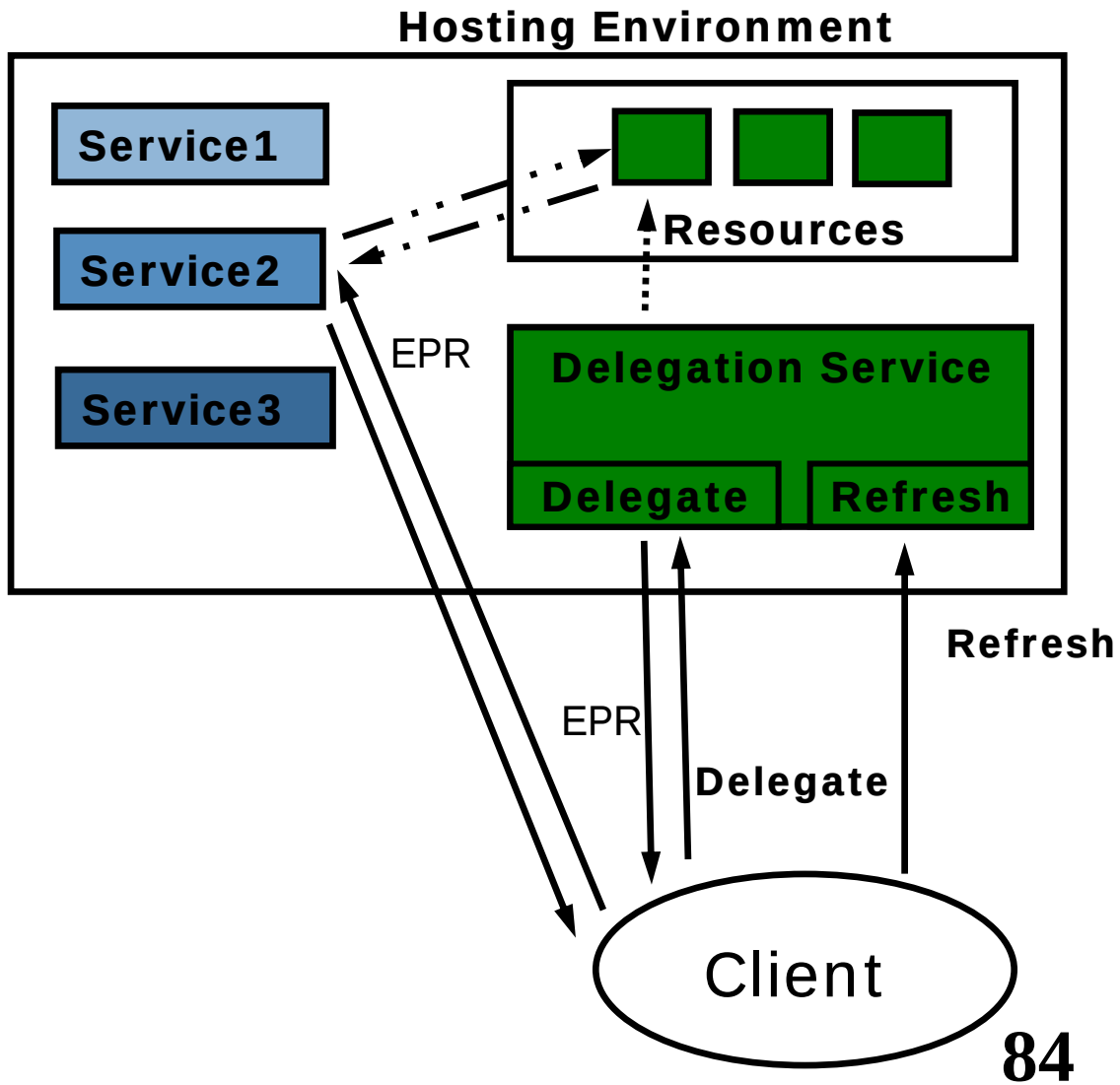
- Don't need to remember (or even know) ID/passwords for each resource.
- Automatically get a Grid proxy certificate for use with other Grid tools
- More secure
  - No ID/password is sent over the wire: not even in encrypted form
  - Proxy certificate expires in a few hours and then is useless to anyone else
  - Don't need to write down 10 passwords
- *It's fast and it's easy!*

the globus alliance
www.globus.org

Proxy Certificate Chain

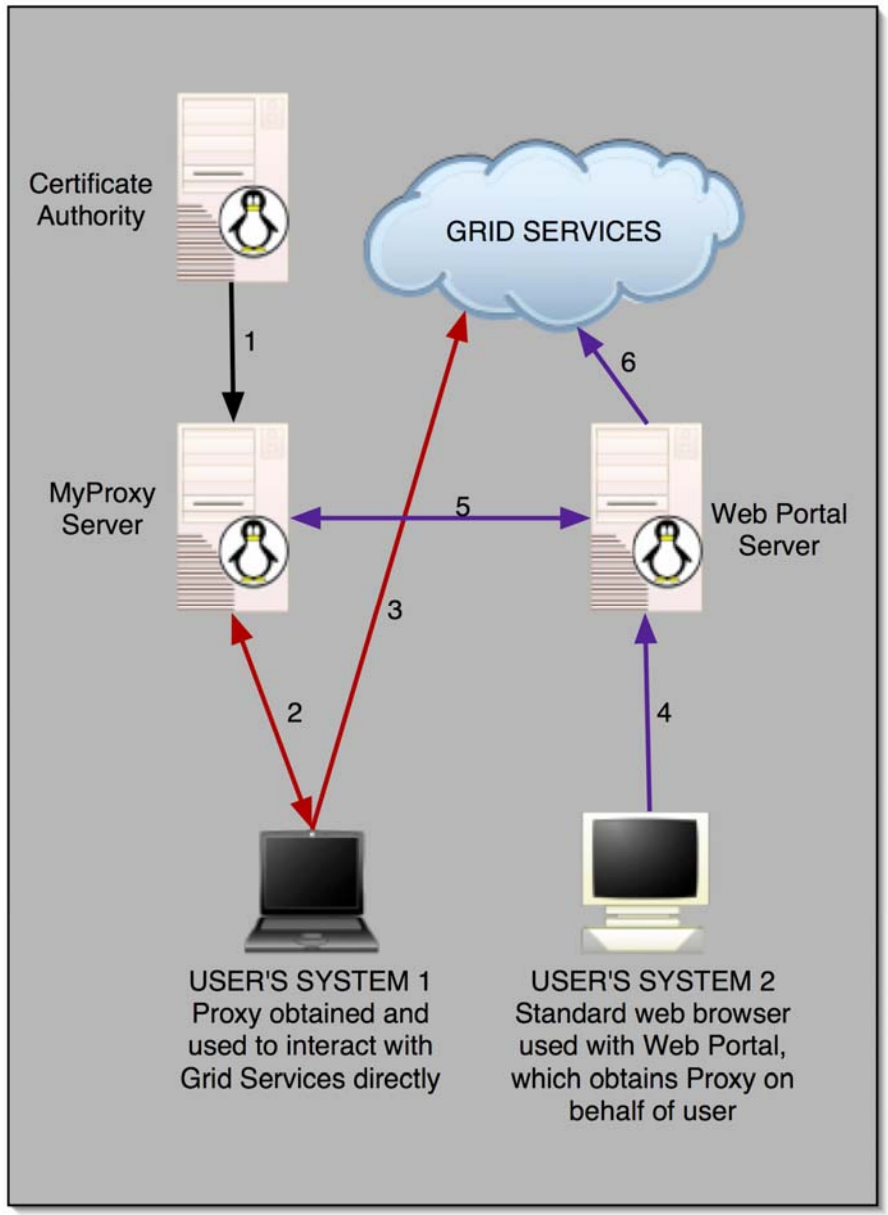
Delegation Service

- Higher level service
- Authentication protocol independent
- Refresh interface
- Delegate once, share across services and invocation

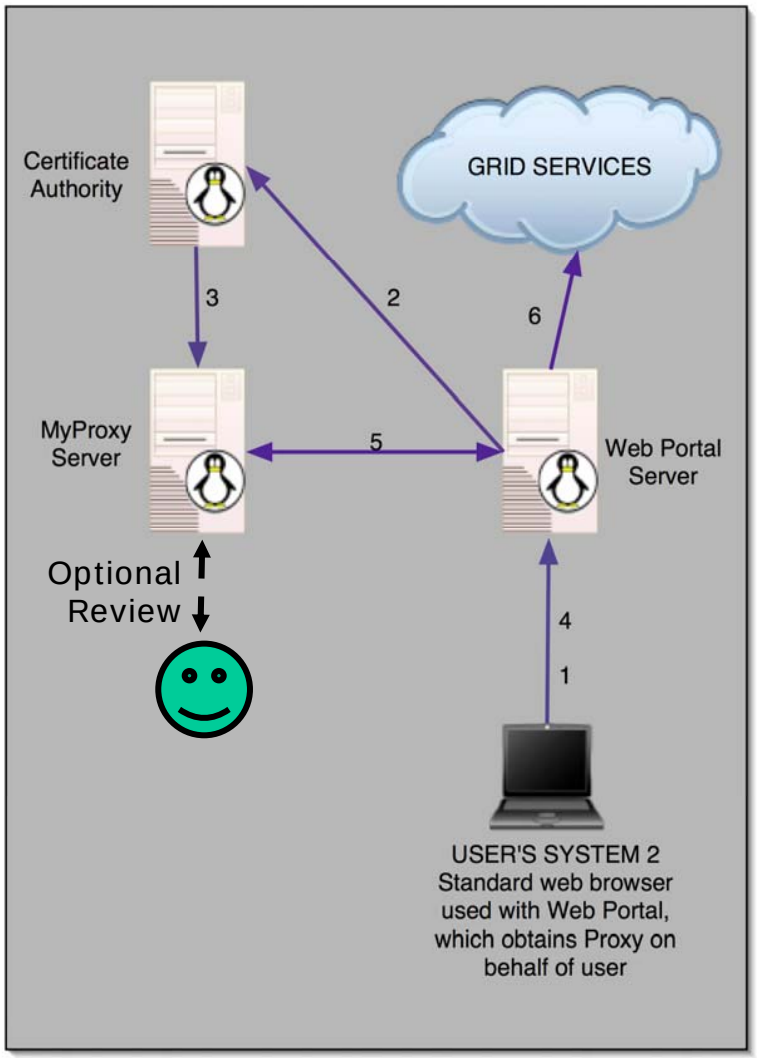


Globus and Delegation: MyProxy

- Remote service that stores user credentials
  - Users request proxies for local use
  - Web Portals request user proxies for use with back-end Grid services
- Grid administrators can pre-load credentials in the server for users to retrieve when needed



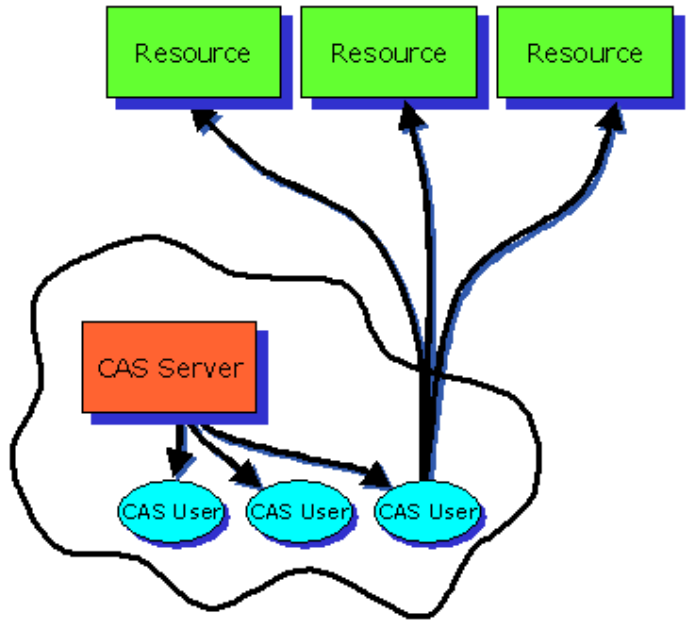
Portal-Based User Registration Service (PURSE)



- Portal extensions (CGI scripts) that automate user registration requests
  - Solicits basic data from user
  - Generates cert request from CA
  - Admin interface allows CA admin to accept/reject request
  - Generates a certificate and stores in MyProxy service
  - Gives user ID/password for MyProxy
- Benefits
  - Users never have to deal with certs
  - Portal can get user cert from MyProxy when needed
  - Database is populated with user data
- Originally written for ESG, now generalized

CAS: Community Authorization Service

- Allows resource providers to specify
  - Course-grained access control policies in terms of communities as a whole
  - Fine-grained access control is delegated to the community
- Resource providers maintain authority over their resources use



TeraGrid's Remote Login Capability

- All TG resources that support the CTSS Remote Login capability support
  - Grid authentication via GSI-SSH
  - All TG users have a TG Grid certificate accessible via MyProxy
- Benefits
  - Use one ID/password (the one you use with the TG User Portal)
  - Enter ID/password once per session, regardless of how many TG resources used
  - Sets up a Grid certificate when you login, so Grid tools work automatically
  - Integrated with other remote interfaces (e.g., GRAM)

Obtaining a TG Certificate

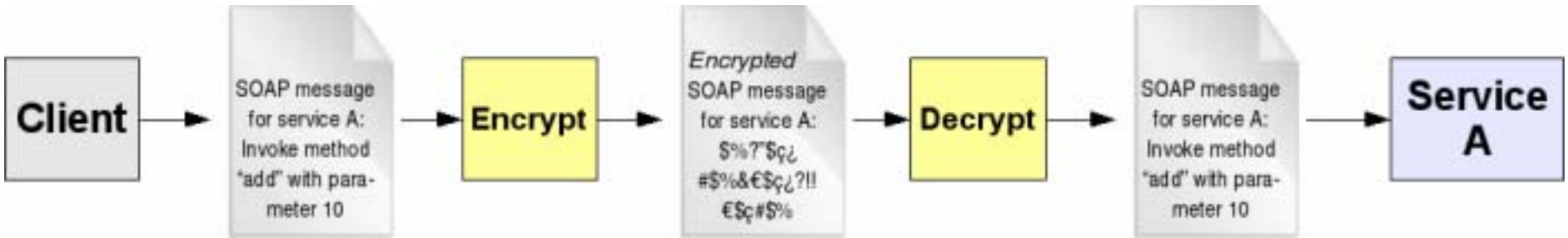
- TG provides a Grid certificate for each user
- Access your certificate via MyProxy.
  - You can install MyProxy on your home system and use it before logging in to TG.
    - > Instructions: http://www.teragrid.org/userinfo/access/
    - > Server: myproxy.teragrid.org
    - > Port: 7514
    - > Use your TG User Portal ID/password
- Or, you can login to a TG resource first (using info provided with your new user packet) and then use MyProxy on the resource

Enabling Private Communication

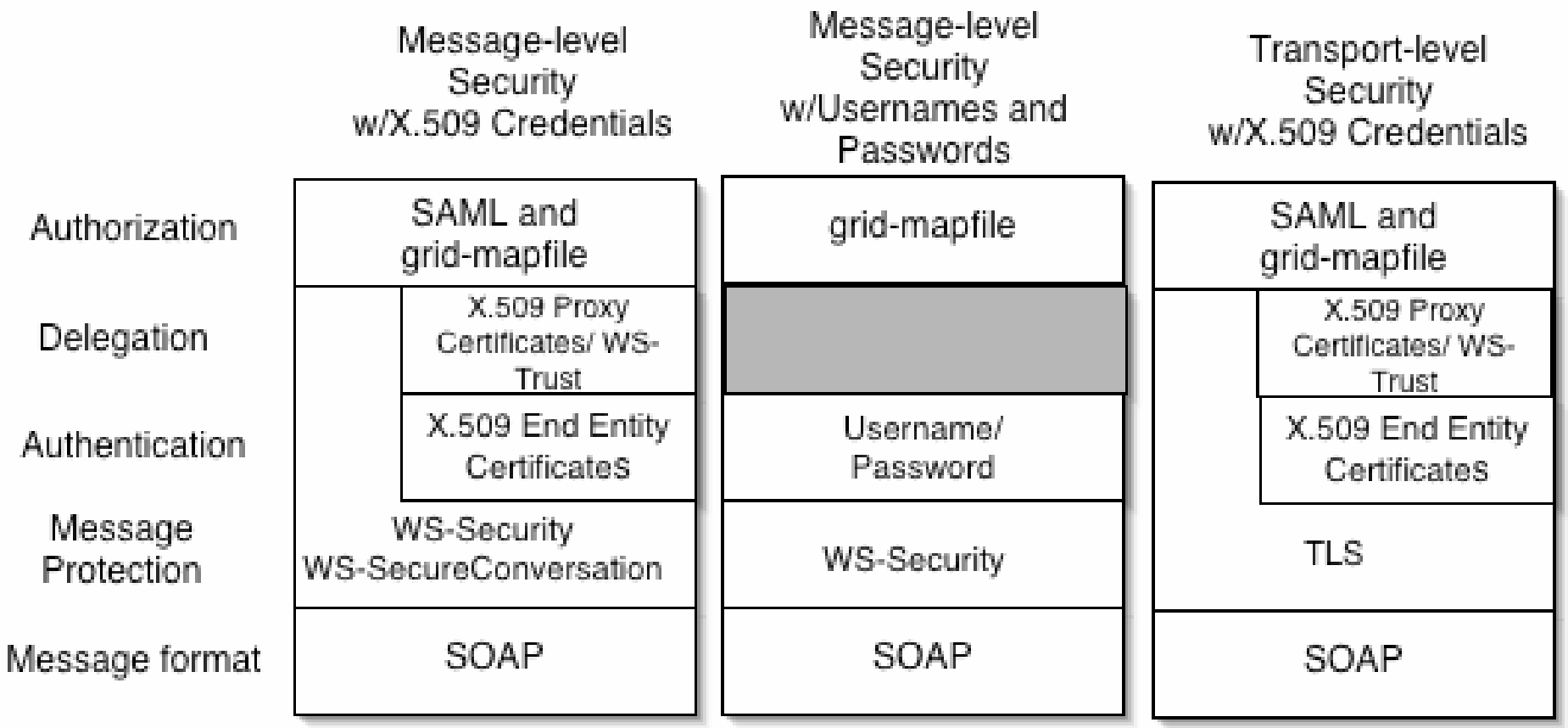
GSI enables security at 2 levels

Transport-level Security (https)

Message-level Security



Globus's Use of Security Standards



Supported, but slow

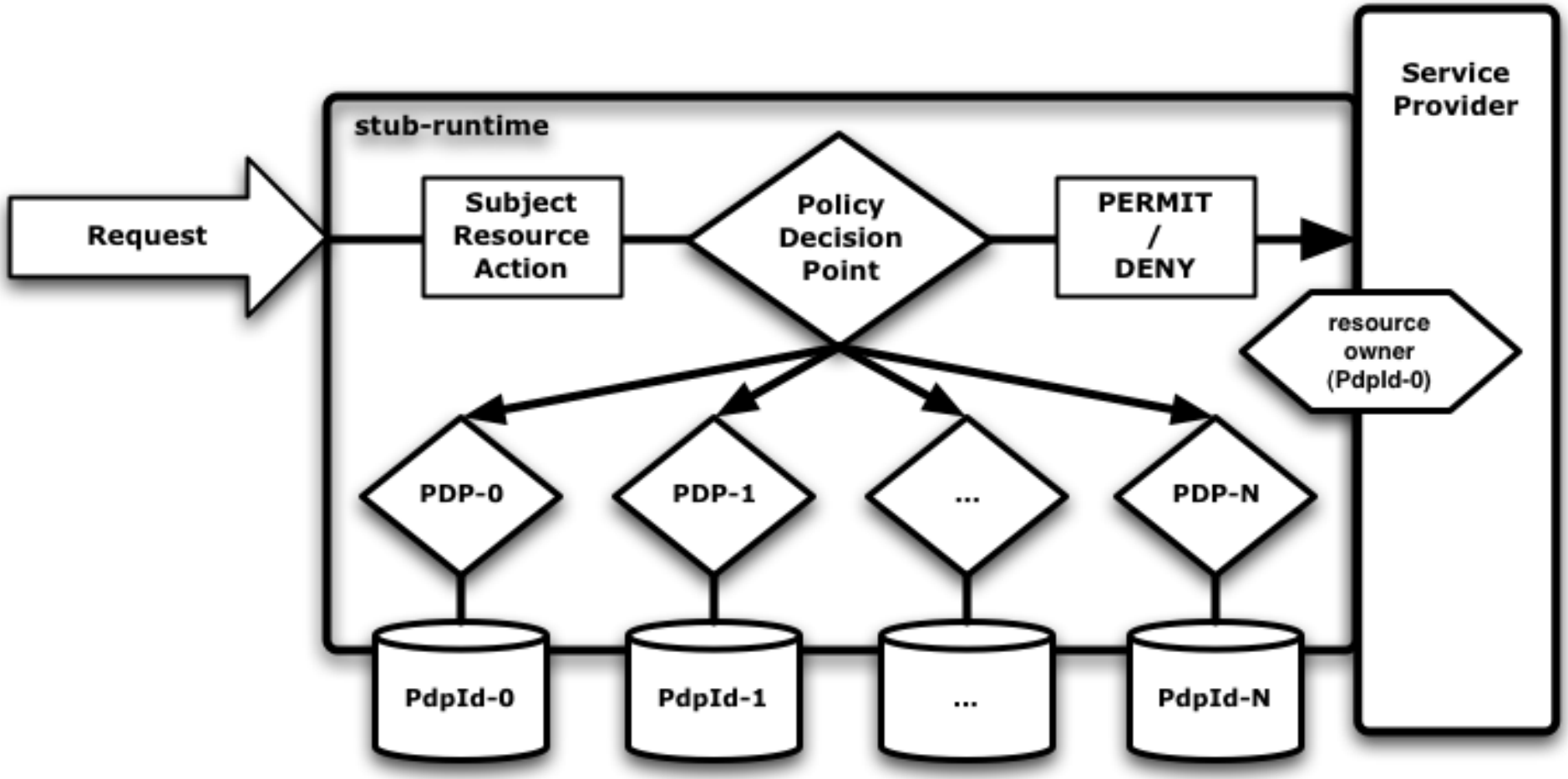
Supported, but insecure

Fastest, so default

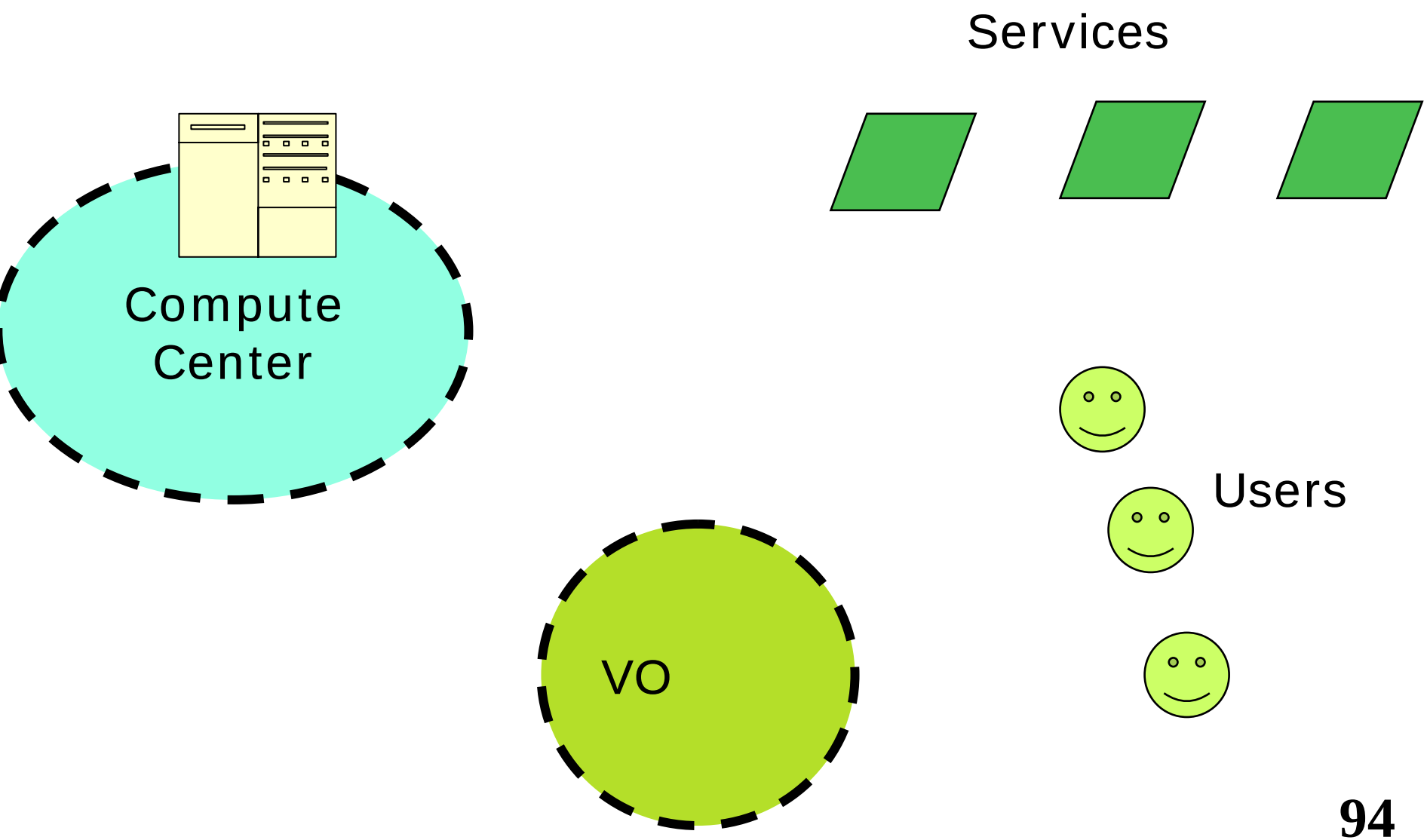
Globus Security

- Extensible authorization framework based on Web services standards
  - SAML-based authorization callout
    - Security Assertion Markup Language, OASIS standard
    - Used for Web Browers authentication often
    - Very short-lived bearer credentials
  - Integrated policy decision engine
    - XACML (eXtensible Access Control Markup Language) policy language, per-operation policies, pluggable

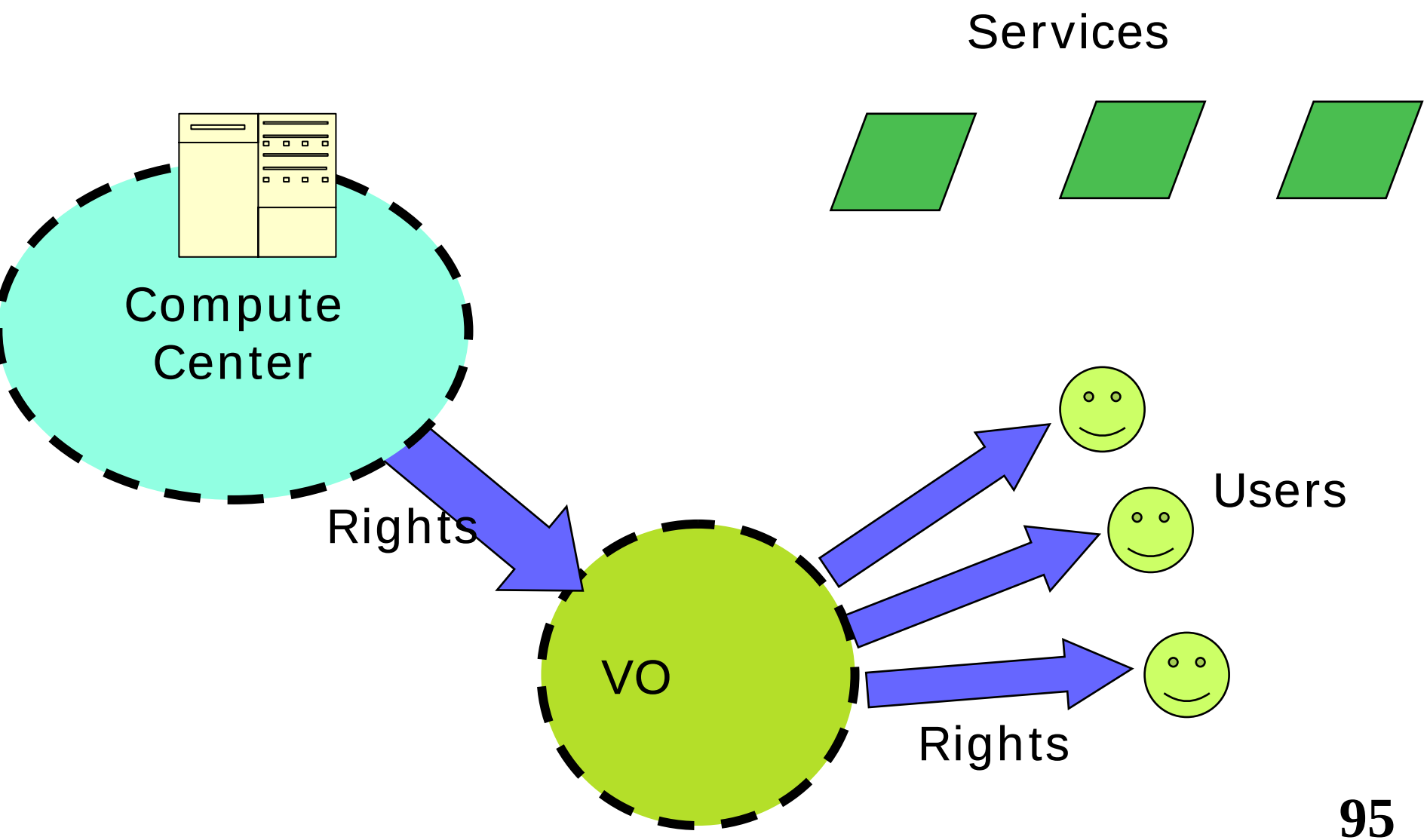
Globus Authorization Framework



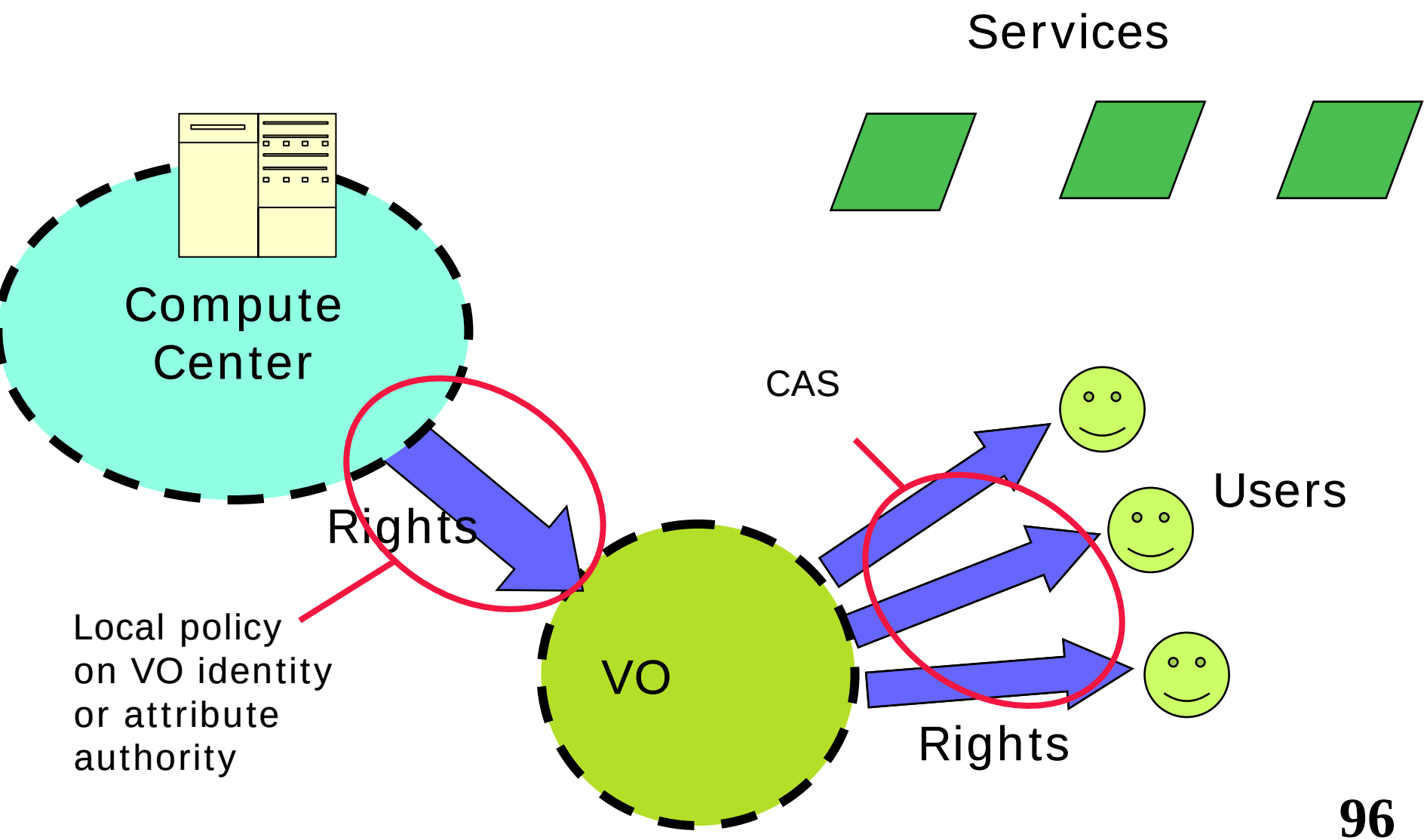
Globus Security: How It Works



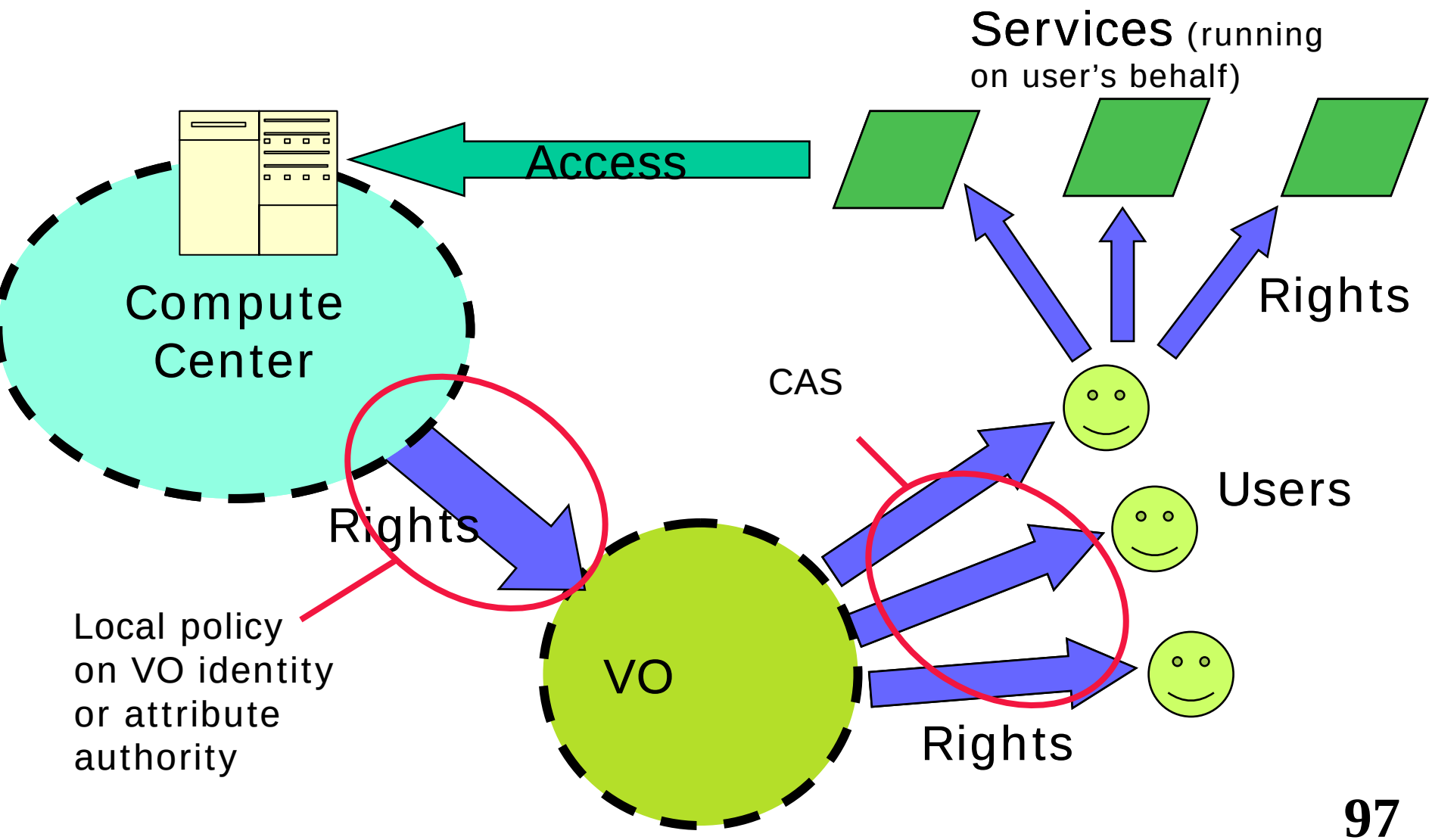
Globus Security: How It Works



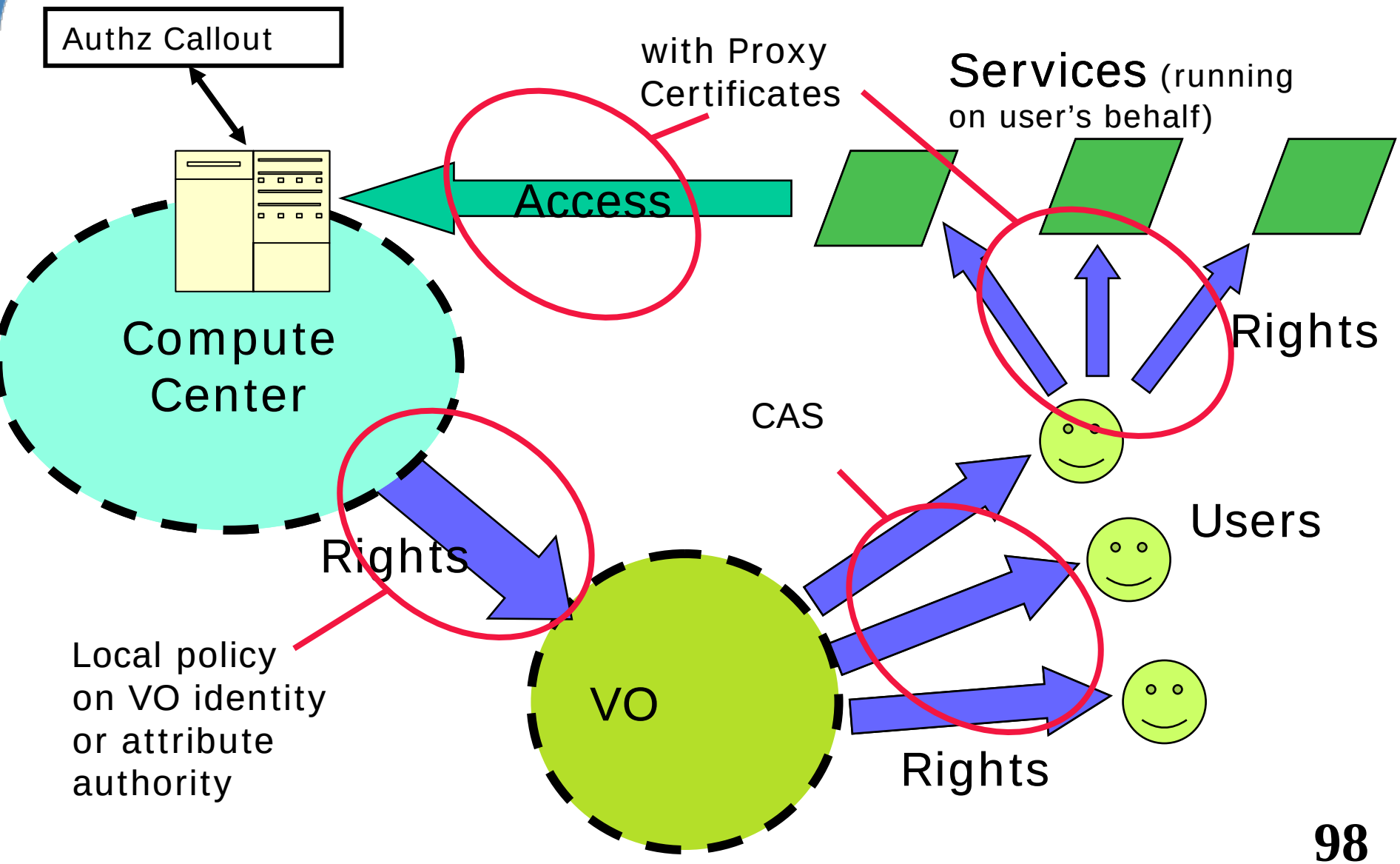
Globus Security: How It Works



Globus Security: How It Works



Globus Security: How It Works



Summary so far

- Basic security concepts
  - Authentication
  - Authorization
- Security tools in Globus

More Specifically, I May Want To …

- Create a service for use by my colleagues
- Manage who is allowed to access my service (or my experimental data or …)
- Ensure reliable & secure distribution of data from my lab to my partners
- Run 10,000 jobs on whatever computers I can get hold of
- Monitor the status of the different resources to which I have access

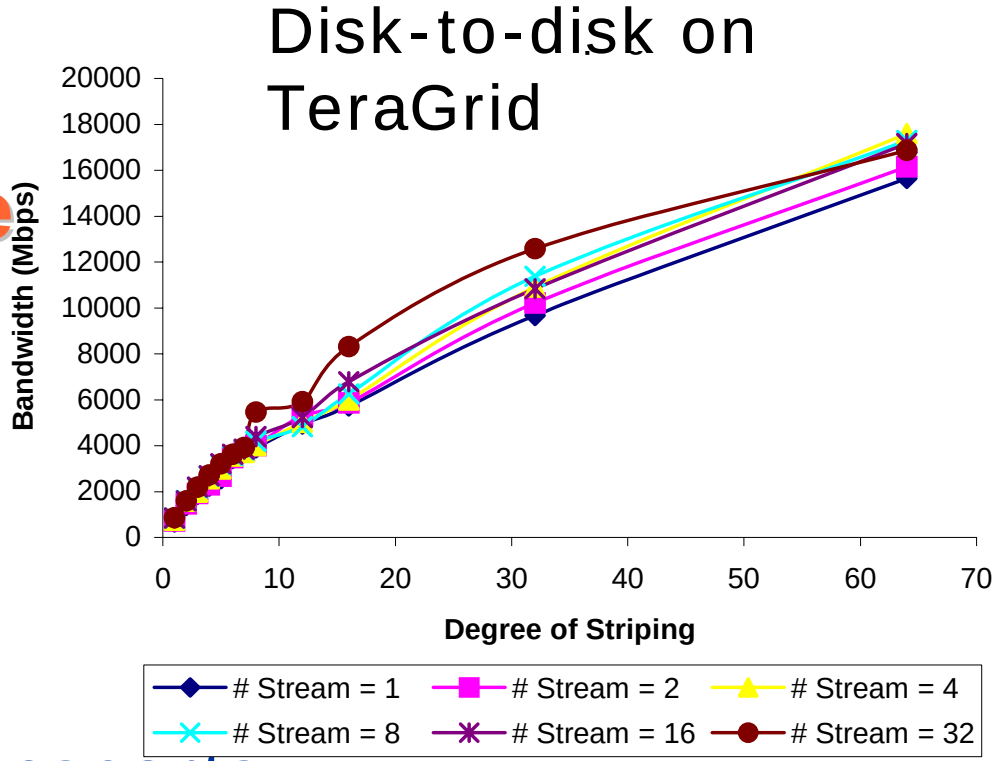
File Replica Management

- Why replicate files?
  - Fault tolerance: avoid single points of failure
  - Reduce latency: use “nearest” copy
- **Stage / move** large data to/from nodes
  - GridFTP for basic file movement
  - Reliable File Transfer (RFT)
- **Locate** data of interest
  - Replica Location Service (RLS)

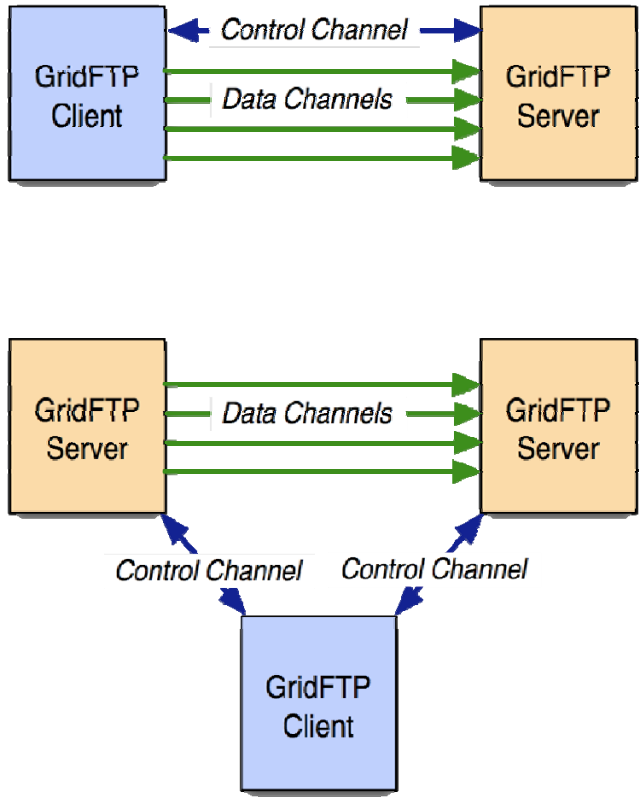
GridFTP: The Protocol

- A high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks
  - FTP with well-defined extensions
  - Uses basic Grid security
  - Multiple data channels for parallel transfers
  - Partial file transfers
  - Third-party transfers
  - Reusable data channels
  - Command pipelining
- GGF recommendation GFD.20

GridFTP the Service

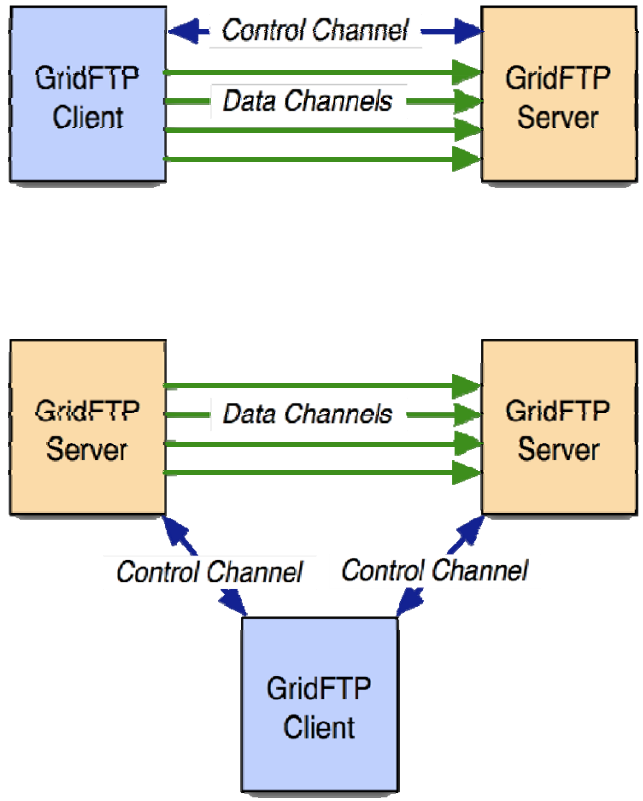


- 100% Globus code
  - No licensing issues
  - Stable, extensible
- IPv6 Support
- XIO for different transports
- Striping → multi-Gb/sec wide area transport
- Pluggable
  - Front-end: e.g., future WS control channel
  - Back-end: e.g., HPSS, cluster file systems
  - Transfer: e.g., UDP, NetBLT transport


Control Channel
GridFTP Client
Data Channels
GridFTP Server
GridFTP Server
Data Channels
GridFTP Server
Control Channel
Control Channel
GridFTP Client

Parallel Data Streams

- Multiple TCP streams between sender and receiver
- Sender pushes multiple blocks in parallel streams
- Blocks reassembled at receiving side and put into correct order
- Protection against dropped packets for each stream

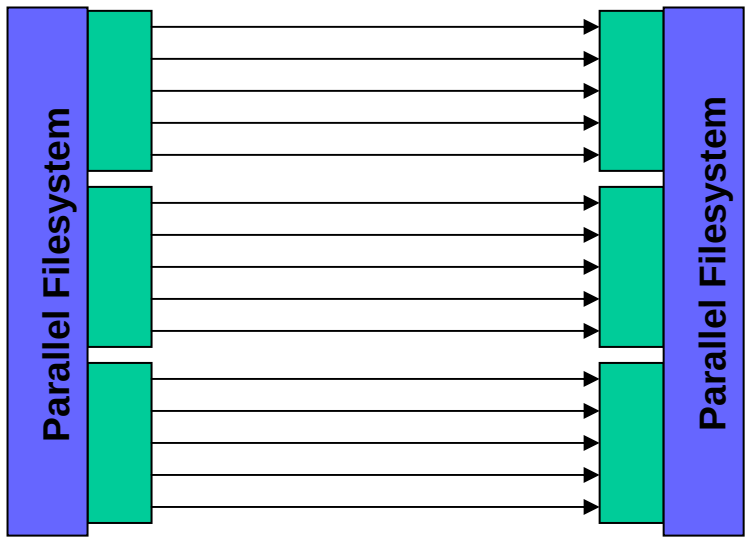


Parallel Transfer

Fully utilizes bandwidth of network interface on single nodes

Striped GridFTP Service

- Multiple nodes work together as a single logical GridFTP server
- Every node of the cluster is used to transfer data into/out of the cluster
  - Each node reads/writes only pieces they're responsible for
  - Head node coordinates transfers
- Multiple levels of parallelism
  - CPU, bus, NIC, disk etc.
  - Maximizes use of Gbit+ WANs



Striped Transfer
Fully utilizes bandwidth of Gb+ WAN using multiple nodes.

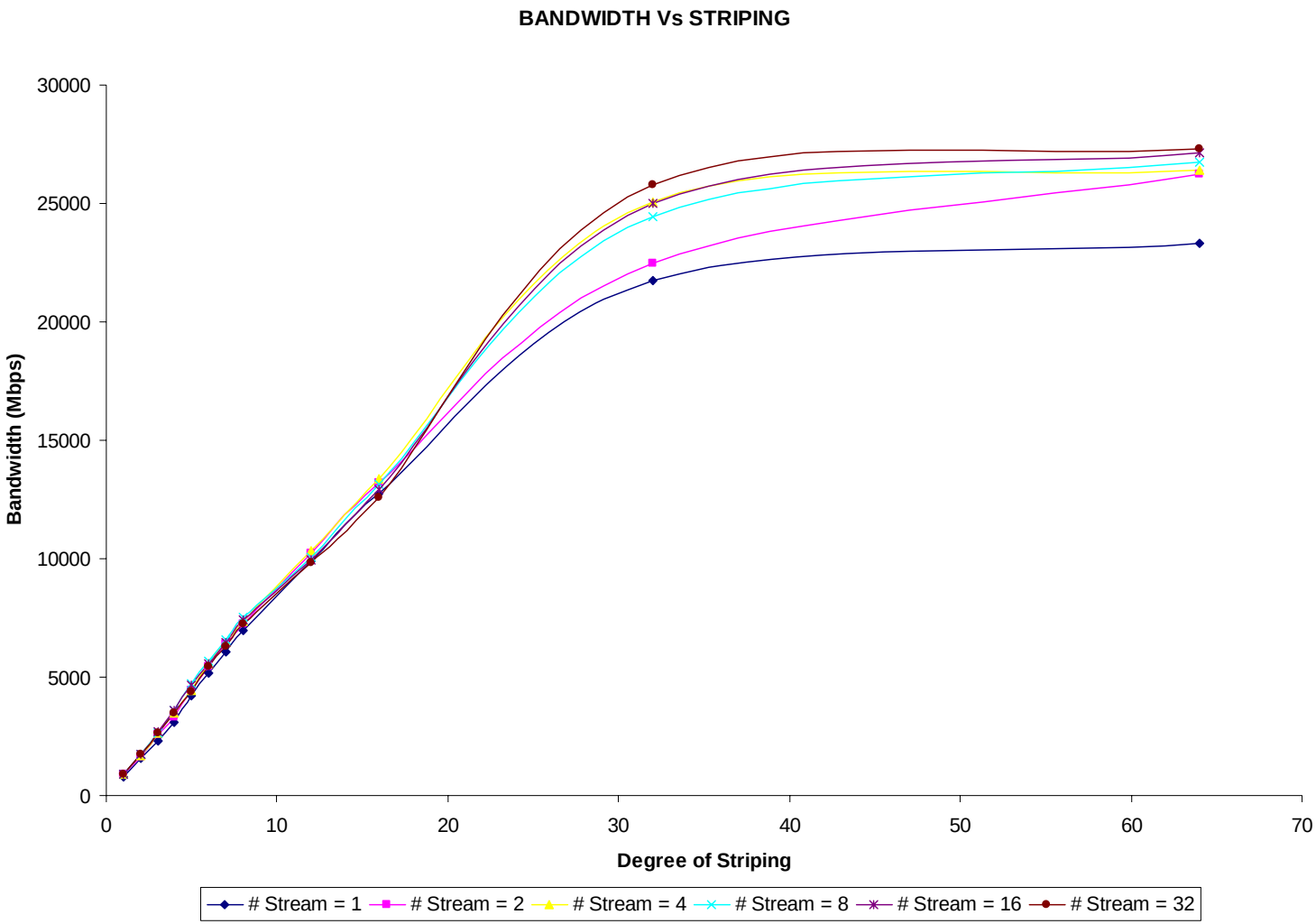
GridFTP Features

- TCP buffer size control
  - Tune buffers to latency of network
  - Regular FTP optimized for low latency networks, not tunable
- Dramatic improvements for high latency WAN transfers
  - 90% of network utilization possible
  - 27 GB/s achieved with commodity hardware

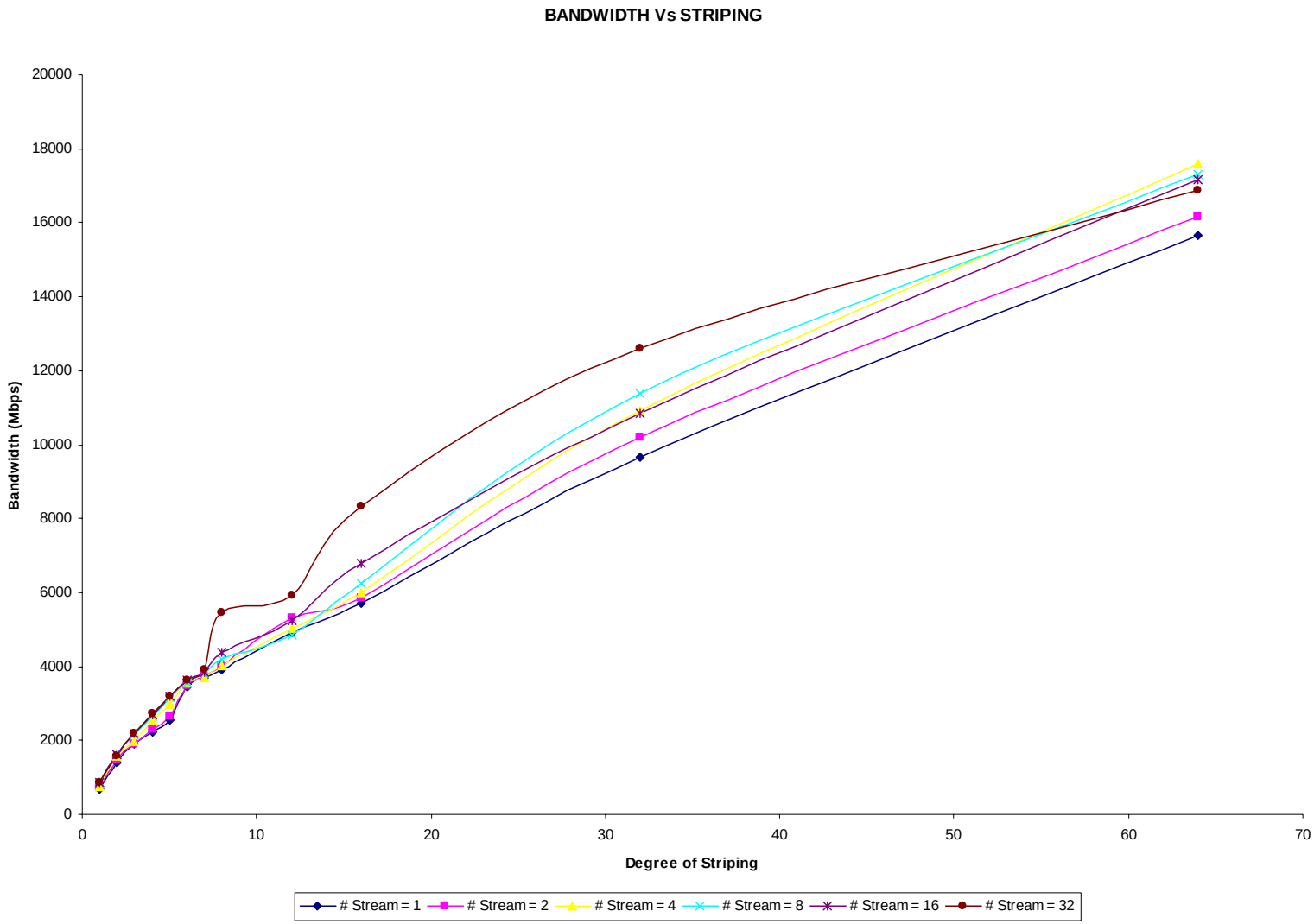
GridFTP

- Integrated instrumentation: Developers can use client API and plug-in mechanism to leverage different instrumentation
  - Performance markers
  - Restart markers
  - Throughput performance
  - Netlogger style performance tracking

Memory to Memory Striping Performance



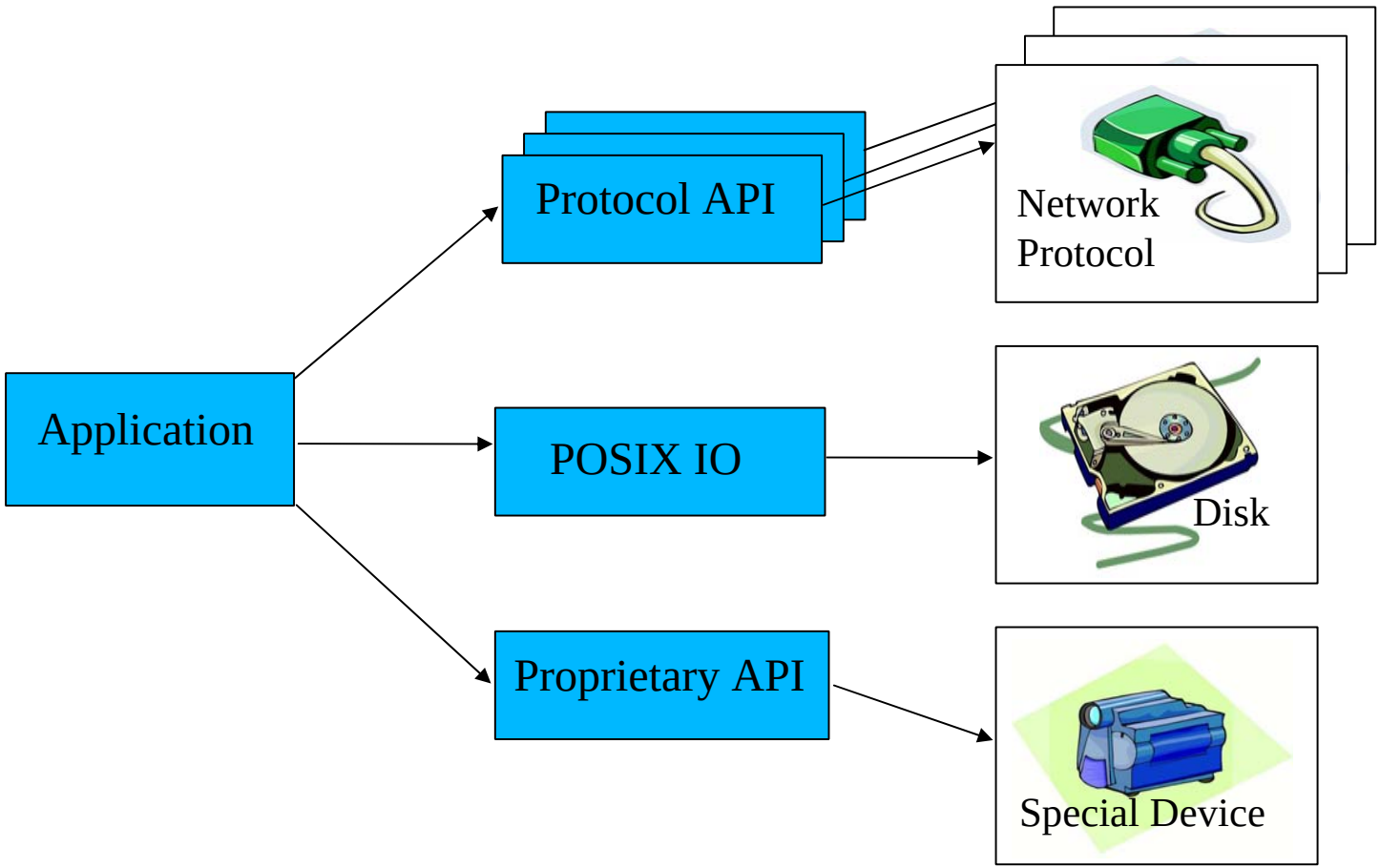
Disk to Disk Striping Performance



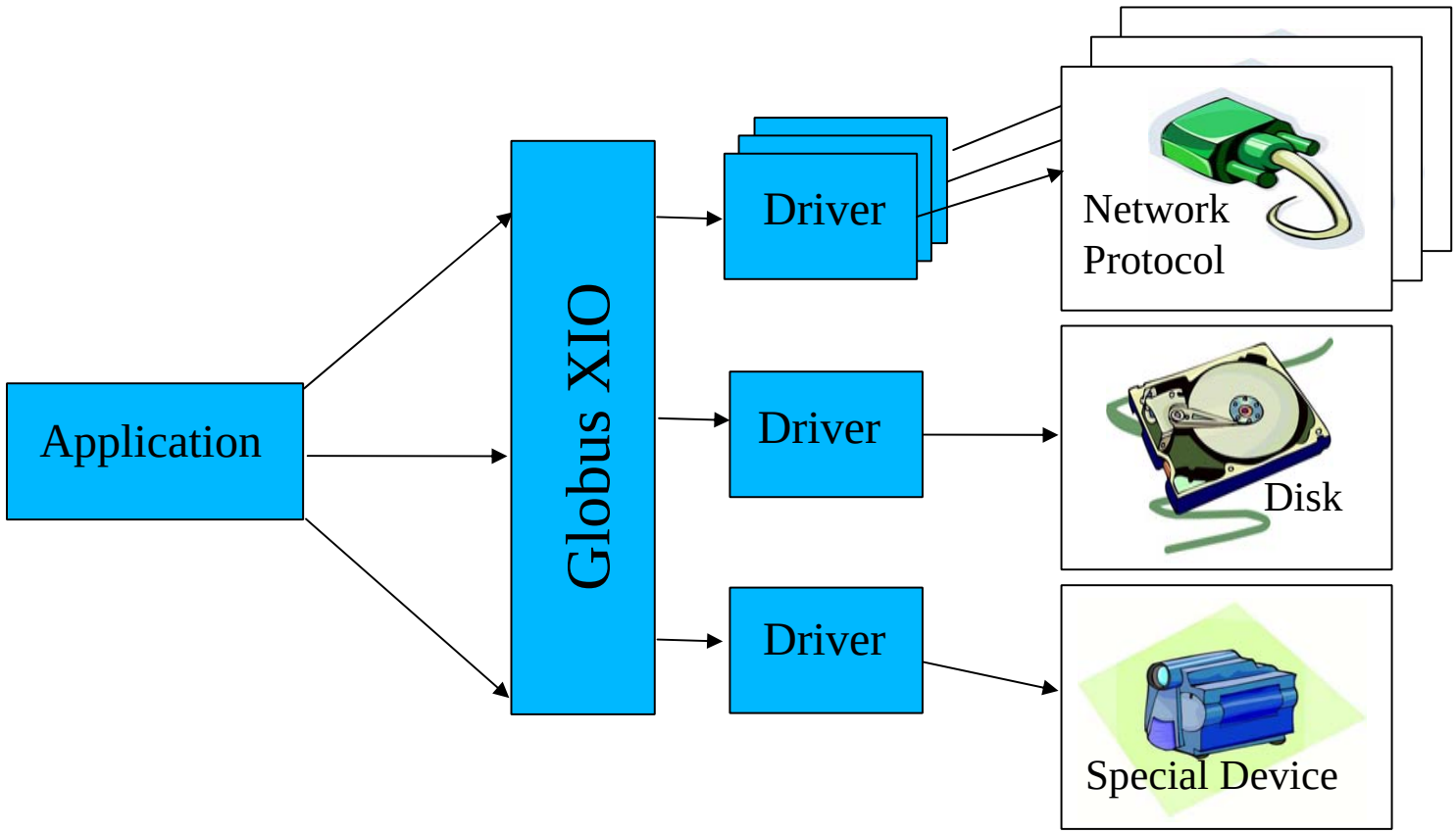
Working with Different Data Transports

- XIO: eXtensible Input/Output
- Library written in C
- Provides a single API that supports multiple wire protocols
- Standard Posix interfaces
  - open/close/read/write
- Protocol implementations encapsulated as drivers

Typical Approach (without XIO)



Globus XIO Approach

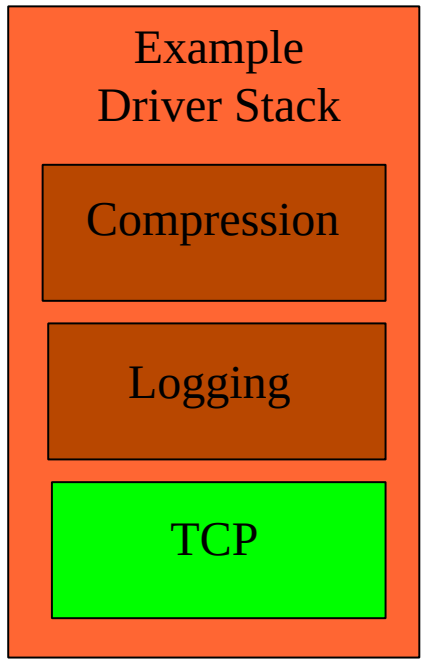


Drivers

- Make 1 API do many types of IO
- Specific drivers for specific protocols/devices
- Transform
  - Manipulate or examine data
  - Do not move data outside of process space
  - Compression, Security, Logging
- Transport
  - Moves data across a wire
  - TCP, UDP, File IO, Device IO
  - Typically move data outside of process space

Stack

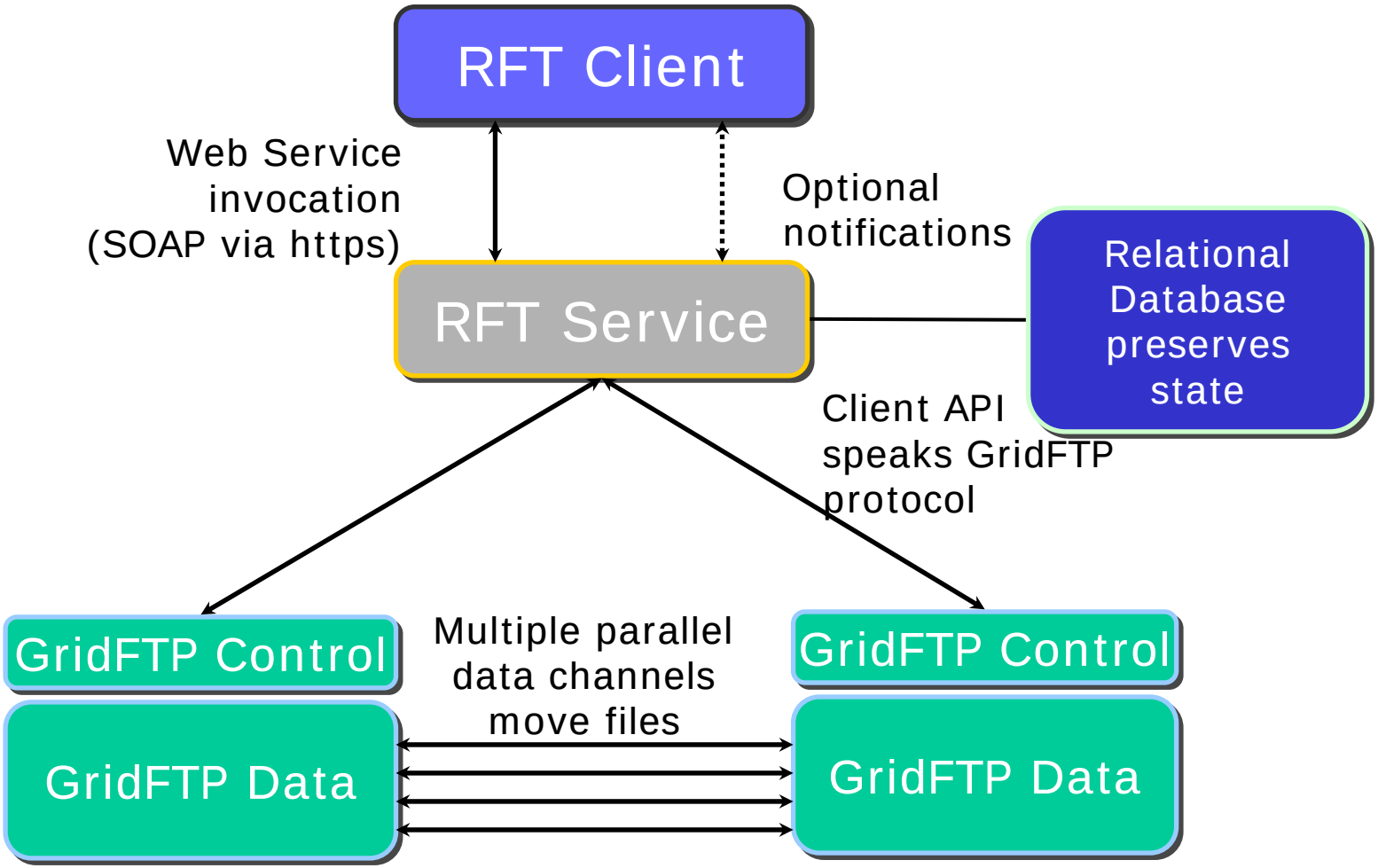
- Transport
  - Exactly one per stack
  - Must be on the bottom
- Transform
  - Zero or many per stack
- Control flows from user to the top of the stack, to the transport driver.



RFT - File Transfer Queuing

- A WSRF service for queuing file transfer requests
  - Server-to-server transfers
  - Checkpointing for restarts
  - Database back-end for failovers
- Allows clients to requests transfers and then “disappear”
  - No need to manage the transfer
  - Status monitoring available if desired

Reliable File Transfer (RFT)



Has transferred >900,000 files.

Globus RFT

- Supports concurrency
  - Multiple files in transit at any given time
  - Useful when transferring many small files
- Restart markers saved by service in DB
  - Failed transfers restarted from midpoint
- Client need not stay connected during transfers
- Clients check status in two ways
  - Subscribe to notifications from RFT service
  - Poll service to find status of transfers

Globus RFT (2)

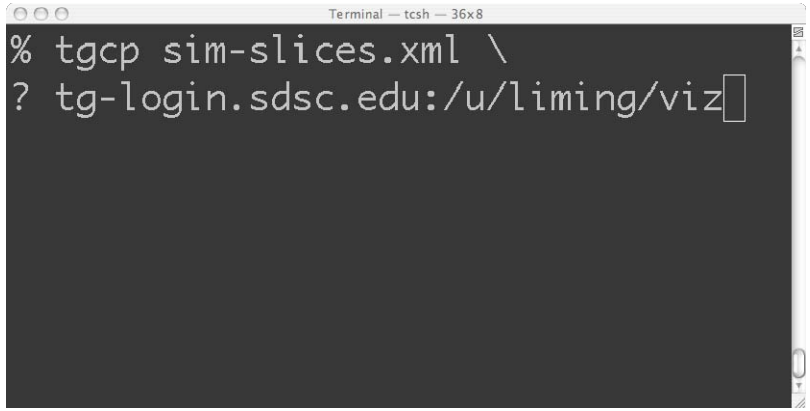
- Exposes WSRF compliant interface
  - Code RFT client using favorite Web services tools
- Single RFT service fronts multiple RFT resources
  - Each “user” can have separate resource
  - Each resource maintains own queue, notifications, lifetime
- Delete sets of files/directories on a GridFTP server
- Configurable exponential back off before retrying failed transfer
- Configurable number of retries for failed transfers per request
- Transfer all or none option

TGCP - TeraGrid Copy

- Applies a set of transformation rules to source and destination
  - Local admin supplies the rules
  - Adds host/port and appropriate path information, puts into GridFTP URL format
- When source/dest sites are identified, add network tuning parameters based on a table maintained by admins
- Invoke either g-u-c or rft to perform the transfer
  - Pass through command line options

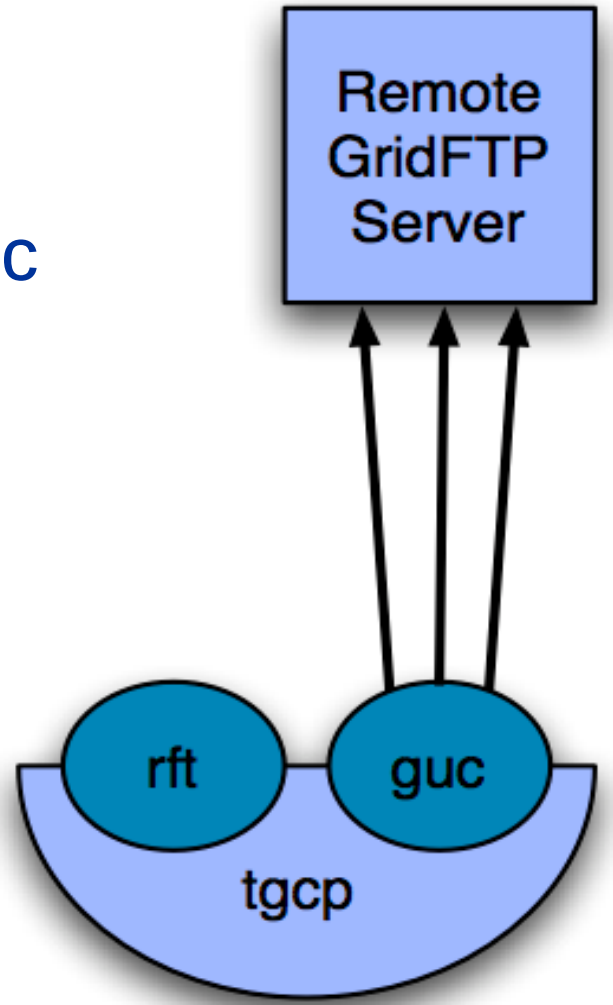
TGCP User Interface

- SCP-style source and destination
  - host:path
- Two options
  - -big - Use striped transfer
  - -rft - Manage the transfer
- If source is a directory, RFT will transfer the full contents of the directory



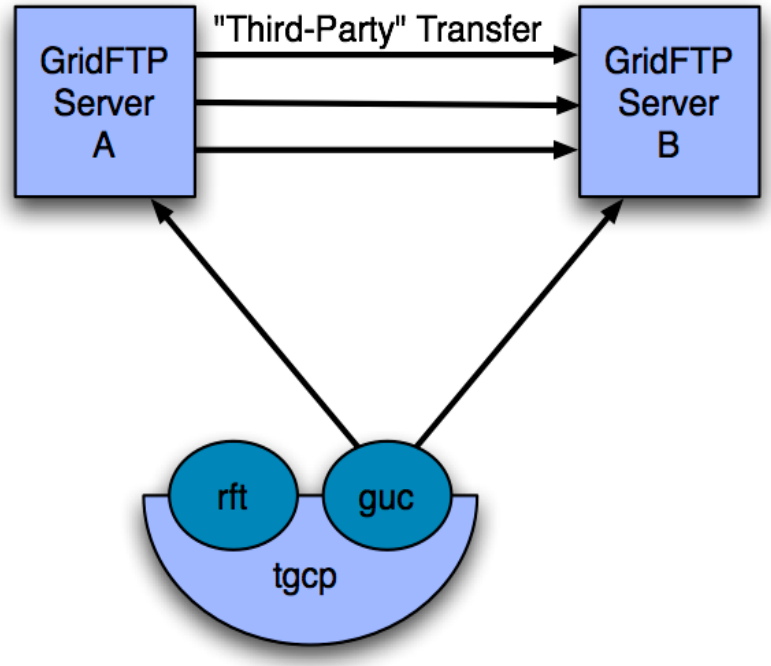
Scenario #1 - Non-shared Local Filesystem

- If the local GridFTP servers can't get to the local file, tgcp uses g-u-c as a GridFTP client
- Parallelism and network tuning parameters help to optimize transfer.



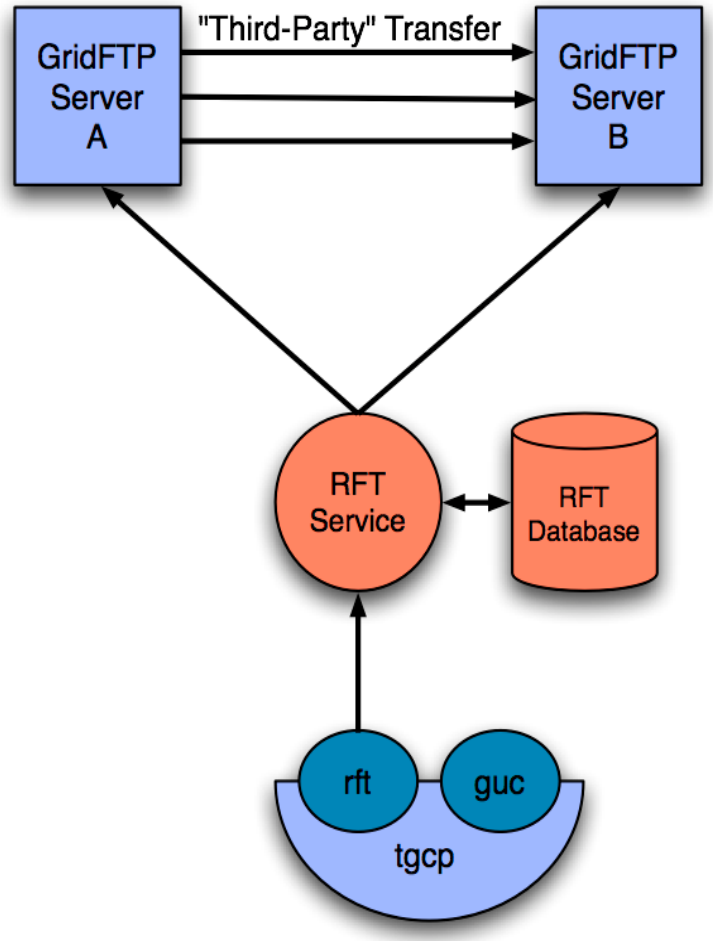
Scenario #2 - High-Performance Transfer

- If the file is accessible to servers at both ends, tgcp can invoke a third party transfer
  - Striping can be performed as well as parallelism
  - Servers typically have high-performance NICs



Scenario #3 - Managed High-perf. Transfer

- For many files to be transferred (e.g., a directory) or the user wants to “fire and forget”, tgcp uses rft
  - Requests stored in a persistent database.
  - Failure recovery
  - RFT can use channel caching for even better performance with small files.



So we can…

- Move files between servers
- Reliably
- With or without a Web service interface

- But we wanted to work with replicas!

Globus Replica Location Service

- Maintains mappings between logical identifiers and target names
- Logical identifier or Logical File Name (LFN)
  - Location-independent identifier (name)
  - Example: `foo`
- Target name or Physical File Name (PFN)
  - Specific file identifier such as a URL
  - E.g.: `gsiftp://myserver.mycompany.com/foo`
- RLS maps between LFNs and PFNs
  - `foo` ⇒ `gsiftp://myserver.mycompany.com/foo`

LFNs and PFNs

- LFN to PFN mappings are often many-to-one
- Multiple PFNs may indicate different access to a file

Local Replica Catalog

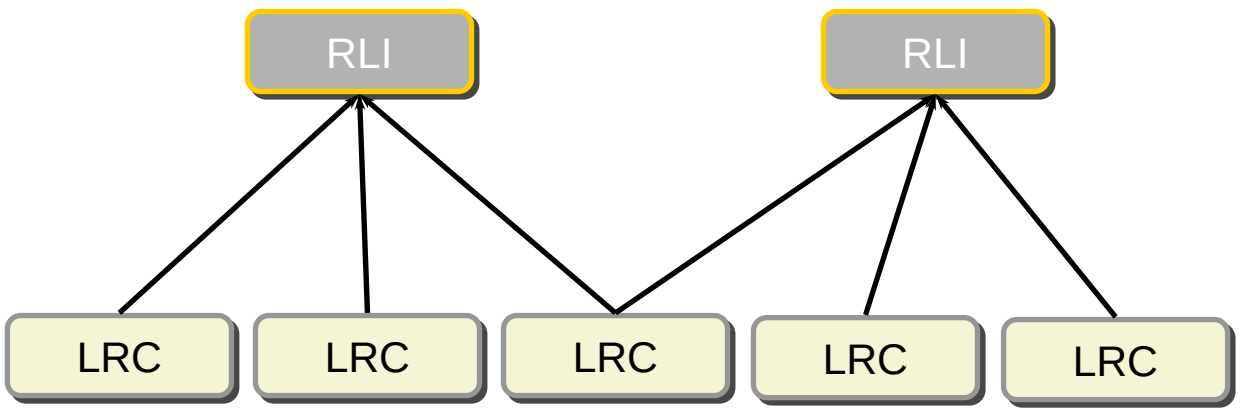
- Local replica catalog (LRC): Catalog of LFN to PFN mappings
- LRCs contain consistent information about local to target mappings

Local Replica Catalog (LRC)

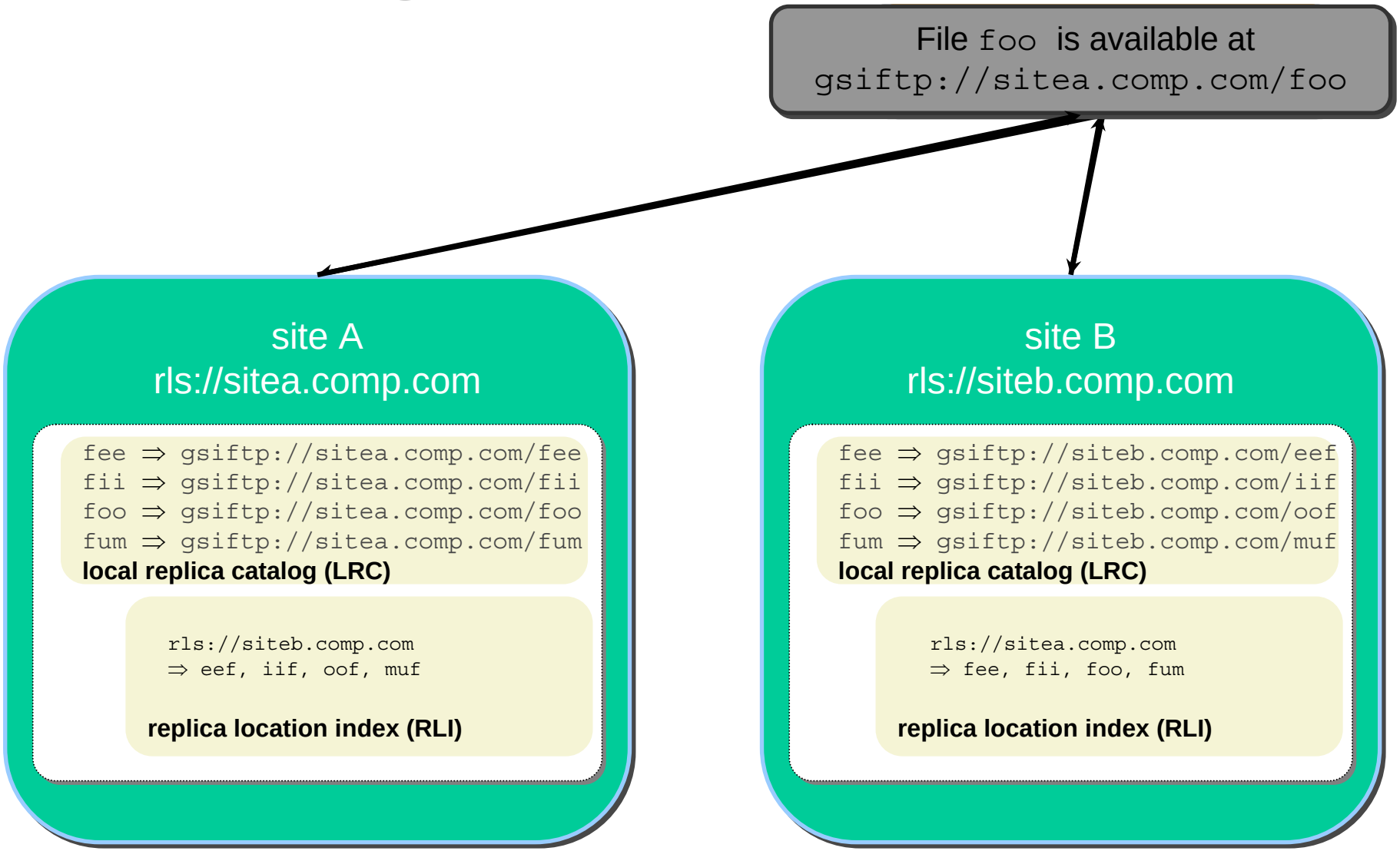
```
fee ⇒ gsiftp://dataserver.mycompany.com/fee
fii ⇒ file://nodeA.mycompany.com/fii
foo ⇒ file://nodeB.mycompany.com/foo
fum ⇒ https://www.mycompany.com/fum
```

Replica Location Index

- Replica Location Index (RLI): Aggregate information about one or more LRCs
- Only the LFN content for LRC is aggregated
  - Each configured LRC sends list of LFNs to LRCs
  - PFNs and mappings **not** aggregated



Finding Files Across the Grid

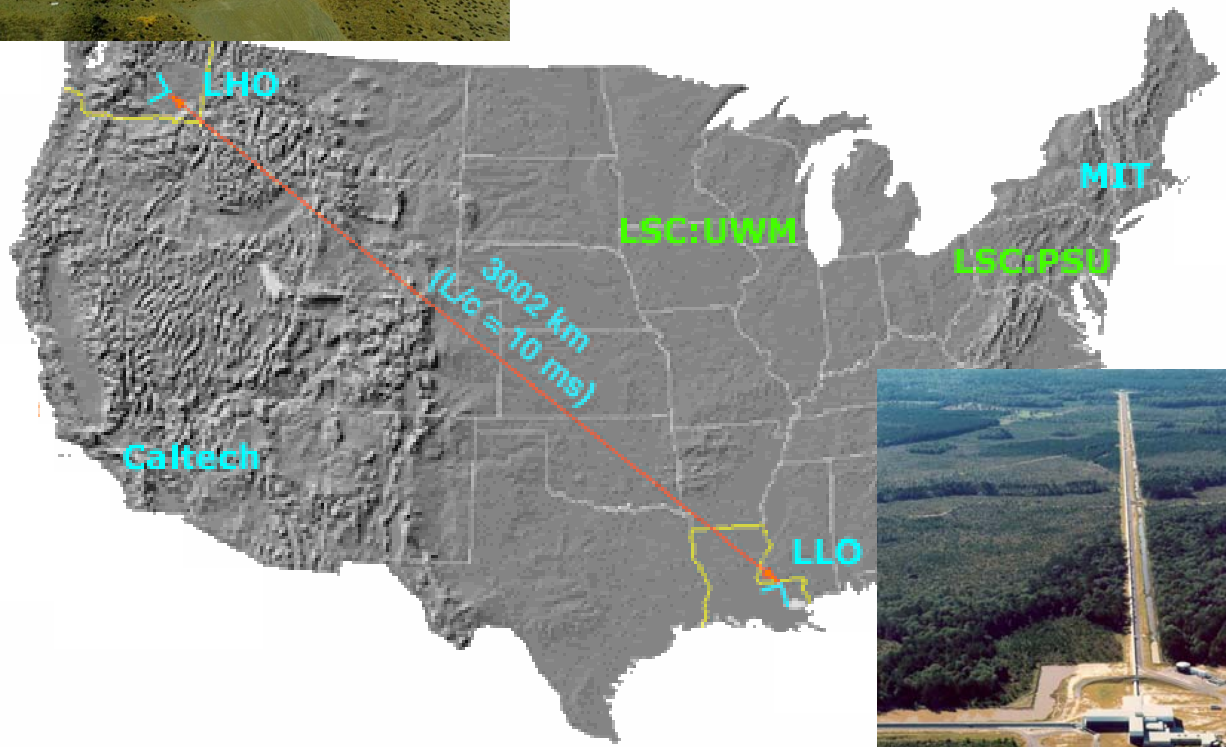


Globus RLS

- **Soft state update** from LRCs to RLIs
  - Relaxed consistency of index
  - Tunable depending on desired load
- Two alternative update methods supported
  - Full list updates send entire list of LFNs periodically, partial updates in between
    - > Complete list means always accurate
    - > Large lists put drain on network, CPU, storage
  - Optional compressed bloom filter or hash
    - > Compression relieves load on network, CPU, storage
    - > False positives are possible (tunable rate)

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

LSC

LIGO Data Grid: Before & After

Before:

- Data replication via "FedEx" Grid
- Ad-hoc site-by-site idioms for finding data in storage
- Ad-hoc error prone mapping from metadata to file names
- Workflow limited to a single compute resource site

After:

- 24 x 7 x 365 continuous fault tolerant data streaming
- Single client tool for scientists and applications to find data
- Scientists concentrate on metadata and forget file names
- Multi-site planning of workflows across LIGO Data Grid

LIGO scientists searching for signals from neutron stars and black holes run **more jobs** across **more resources** and access **more data** using the LIGO Data Grid built on Globus**.**

Papers are published faster due to Globus and the LIGO Data Grid.

Another Data Management Use Case

- Instead of accessing replicated files – what if you're working with a distributed database?

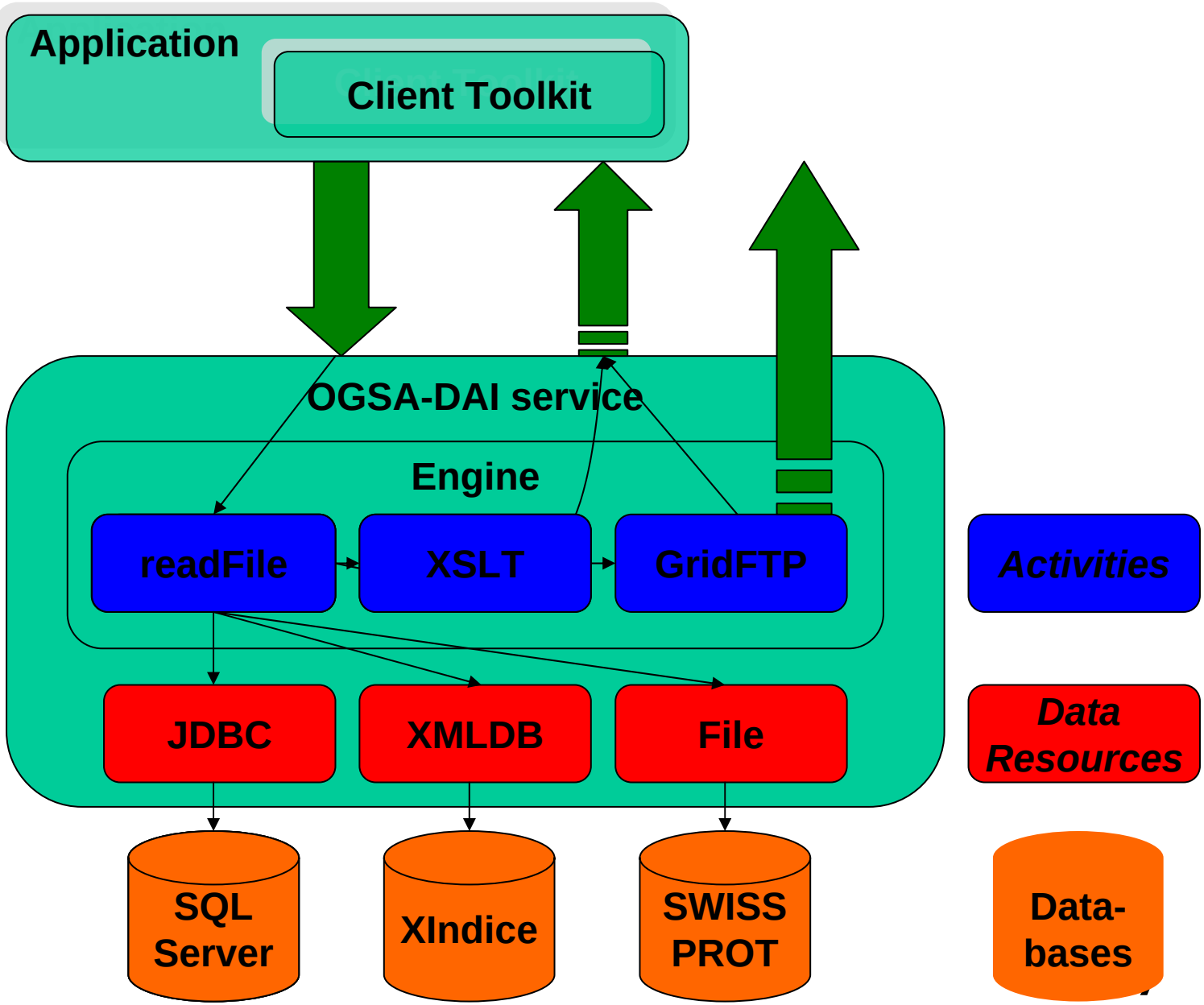
OGSA-DAI

- Grid Interfaces to Databases
  - Data access
    - > Relational & XML Databases, semi-structured files
  - Data integration
    - > Multiple data delivery mechanisms, data translation
- Extensible & Efficient framework
  - Request documents contain multiple tasks
    - > A task = execution of an activity
    - > Group work to enable efficient operation
  - Extensible set of activities
    - > > 30 predefined, framework for writing your own
  - Moves computation to data
  - Pipelined and streaming evaluation
  - Concurrent task evaluation

OGSA-DAI

- Provide service-based access to structured data resources as part of Globus
- Specify a selection of interfaces tailored to various styles of data access—starting with relational and XML

The OGSA-DAI Framework



OGSA-DAI: A Framework for Building Applications

- Supports data access, insert and update
  - Relational: MySQL, Oracle, DB2, SQL Server, Postgres
  - XML: Xindice, eXist
  - Files – CSV, BinX, EMBL, OMIM, SWISSPROT,...
- Supports data delivery
  - SOAP over HTTP
  - FTP; GridFTP
  - E-mail
  - Inter-service
- Supports data transformation
  - XSLT
  - ZIP; GZIP
- Supports security
  - X.509 certificate based security

OGSA-DAI: Other Features

- A framework for building data clients
  - Client toolkit library for application developers
- A framework for developing functionality
  - Extend existing activities, or implement your own
  - Mix and match activities to provide functionality you need
- Highly extensible
  - Customise our out-of-the-box product
  - Provide your own services, client-side support, and data-related functionality

Summary so far

- File transfer tools
  - GridFTP
  - RFT
- Replication with RLS
- OGSA-DAI to work with databases

More Specifically, I May Want To …

- Create a service for use by my colleagues
- Manage who is allowed to access my service (or my experimental data or …)
- Ensure reliable & secure distribution of data from my lab to my partners
- Run 10,000 jobs on whatever computers I can get hold of
- Monitor the status of the different resources to which I have access

Traditional Resource Management Approach

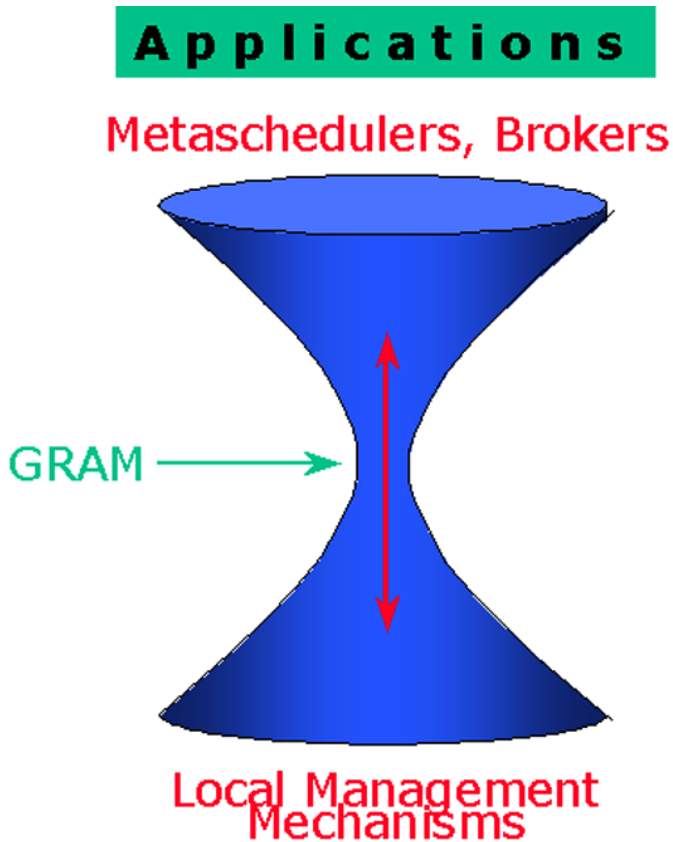
- Have access to numerous sites
  - Accounts, permissions, etc
- Use a Metascheduler to make resource selection decisions
  - GridWay
  - Metascheduler uses GRAM to contact the difference local queuing systems

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

GRAM - Basic Job Submission and Control Service

- A uniform service interface for remote job submission and control
  - Includes file staging and I/O management
  - Includes reliability features
  - Supports basic Grid security mechanisms
  - Available in Pre-WS and WS
- GRAM is *not* a scheduler.
  - No scheduling
  - No metascheduling/brokering
  - Often used as a front-end to schedulers, and often used to simplify metaschedulers/brokers



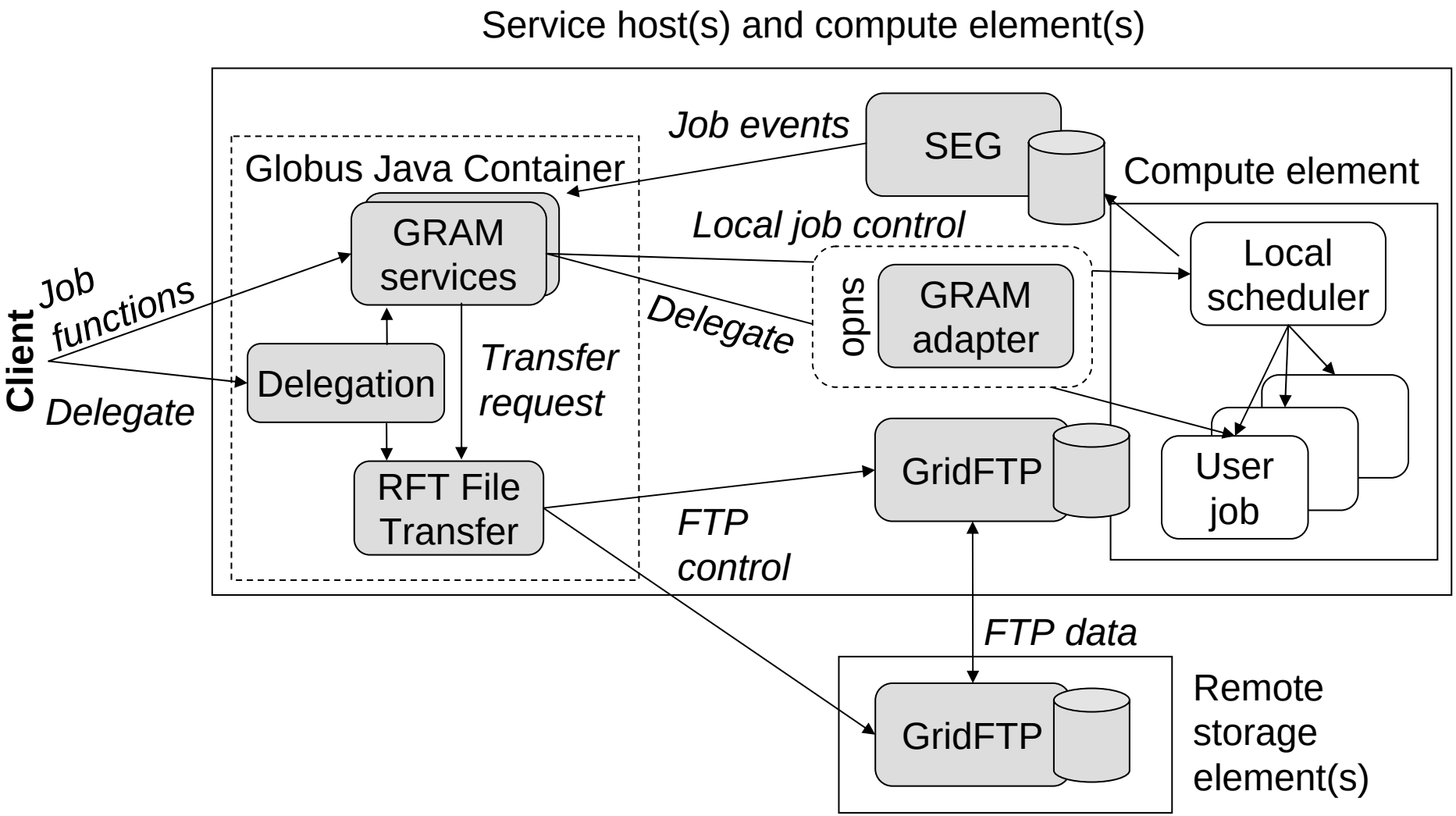
GRAM4 (aka WS GRAM)

- 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

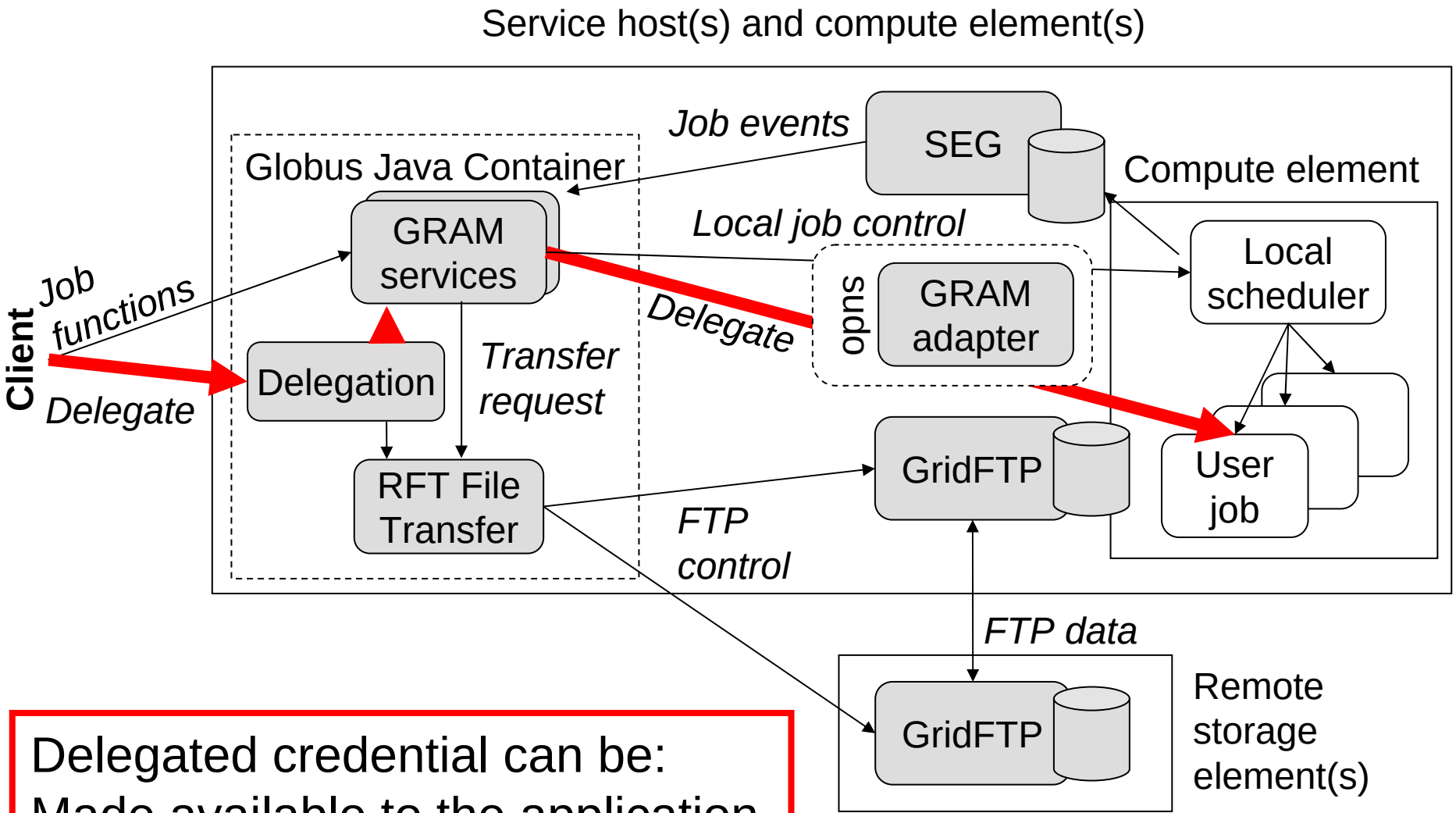
Using GRAM vs Building a Service

- GRAM is intended for jobs that
  - are arbitrary programs
  - need stateful monitoring or credential management
  - Where file staging is important
- If the application is lightweight, with modest input/output, may be a better candidate for hosting directly as a WSRF service

GRAM4 Architecture

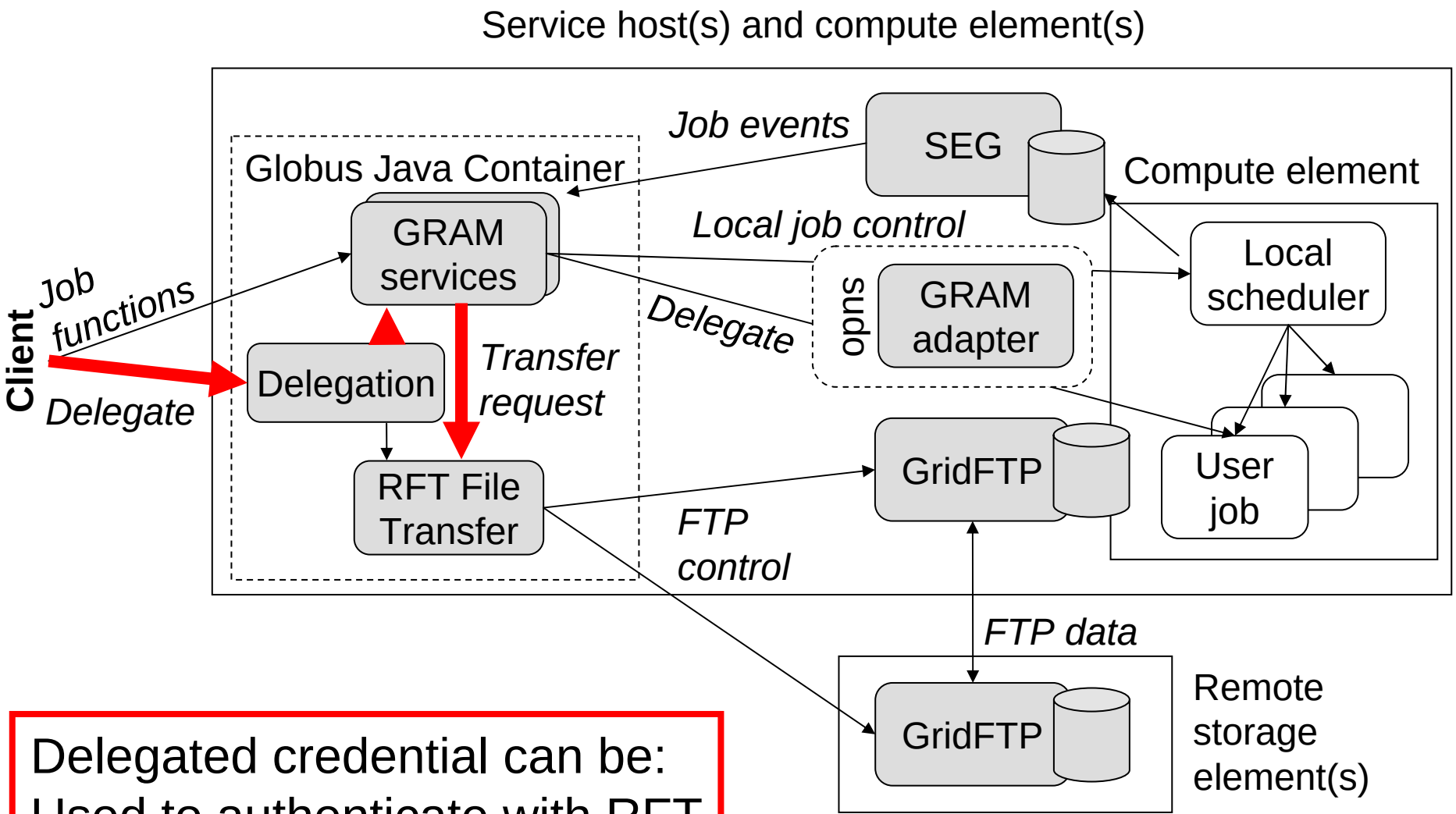


GRAM4 Architecture



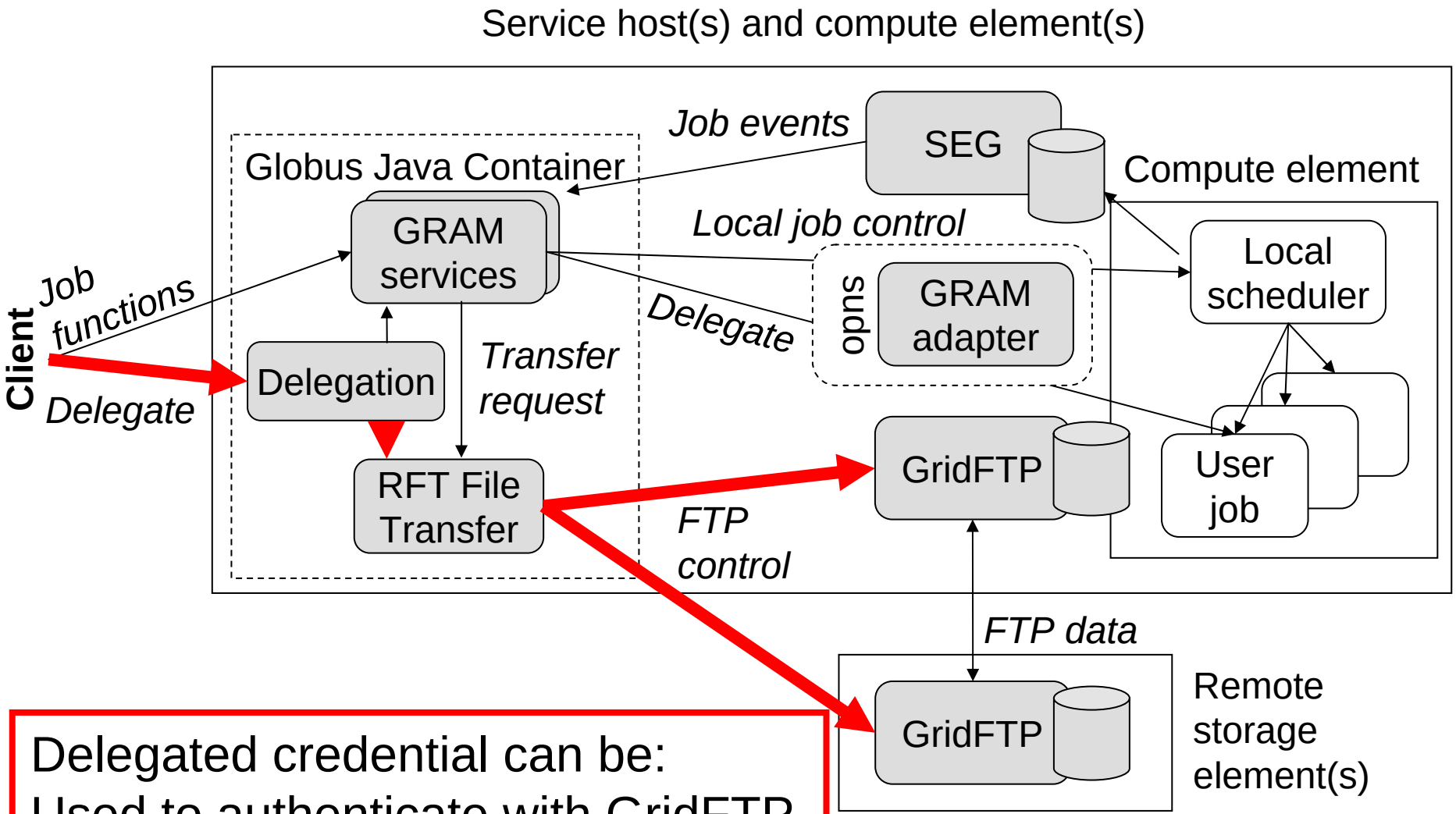
Delegated credential can be:
Made available to the application

GRAM4 Architecture



Delegated credential can be:
Used to authenticate with RFT

GRAM4 Architecture



Delegated credential can be:
Used to authenticate with GridFTP

Submitting a Sample Job

- Specify a remote host with –F

```
globusrun-ws –submit –F host2
                    –job-command /bin/true
```

- The return code will be the job's exit code if supported by the scheduler

Data Staging and Streaming

- Simplest stage-in/stage-out example is stdout/stderr

globusrun-ws –S –s –c /bin/date

- -S is short for “-submit”
- -s is short for –streaming
- The output will be sent back to the terminal, control will not return until the job is done

Resource Specification Language (RSL)

- For more complicated jobs, we'll use RSL to specify the job

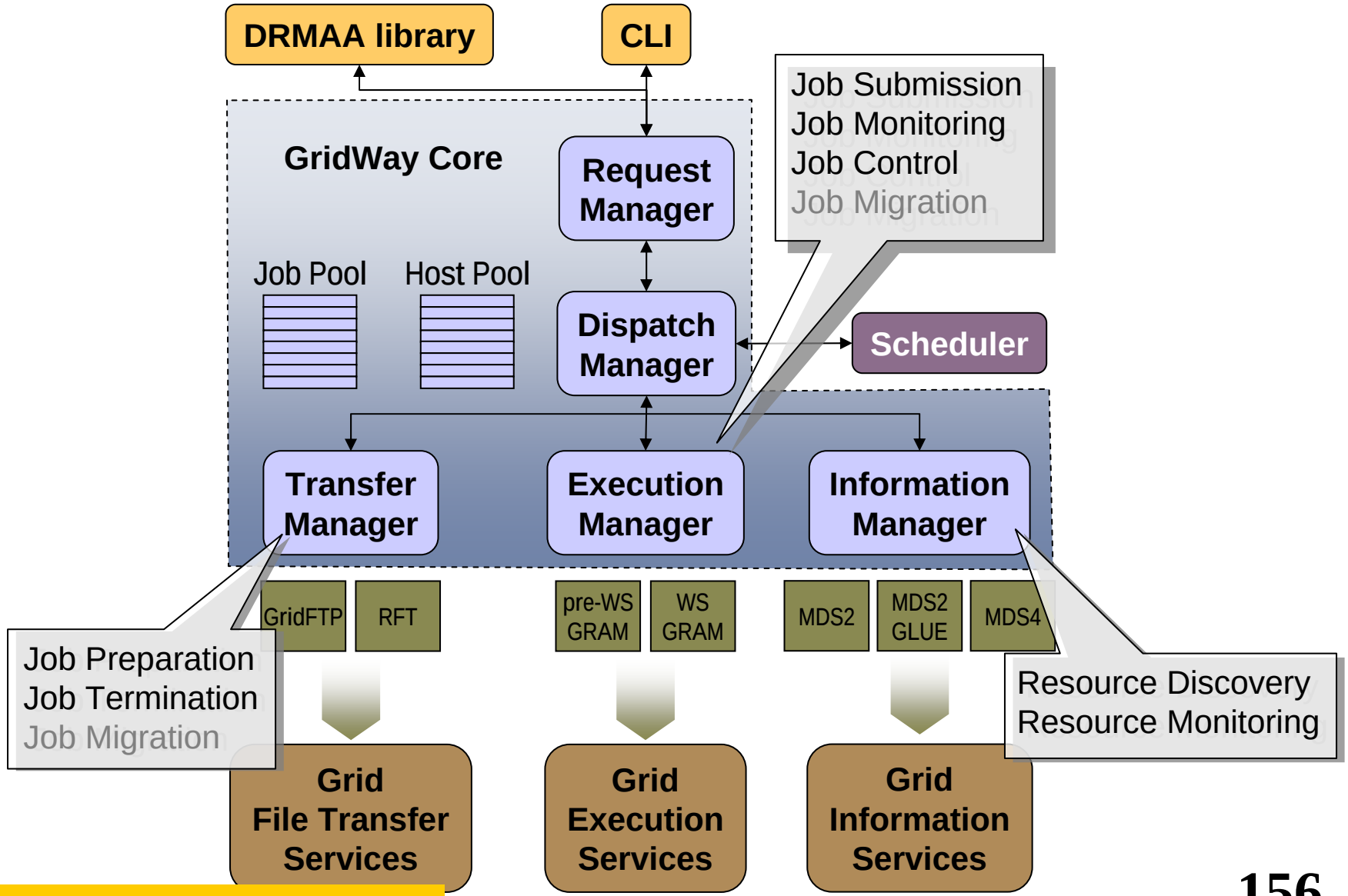
```
<job>
<executable>/bin/echo</executable>
<argument>this is an example_string </argument>
<argument>Globus was here</argument>
<stdout>${GLOBUS_USER_HOME}/stdout</stdout>
 <stderr>${GLOBUS_USER_HOME}/stderr</stderr>
</job>
```

GRAM4: A Big Advance over GRAM2

- Big scalability/performance improvements
  - 32,000 active jobs (GRAM2 max ~100)
  - Ability to manage load on control node
  - Reuse delegated credentials
- New functionality
  - Flexible authorization
  - Modular LRM interface
  - Notifications
  - JSDL support
  - Advance reservation service – in development

GridWay Meta-Scheduler

- Scheduler virtualization layer on top of Globus services
  - A LRM-like environment for submitting, monitoring, and controlling jobs
  - A way to submit jobs to the Grid, without having to worry about the details of exactly which local resource will run the job
  - A policy-driven job scheduler, implementing a variety of access and Grid-aware load balancing policies
  - Accounting

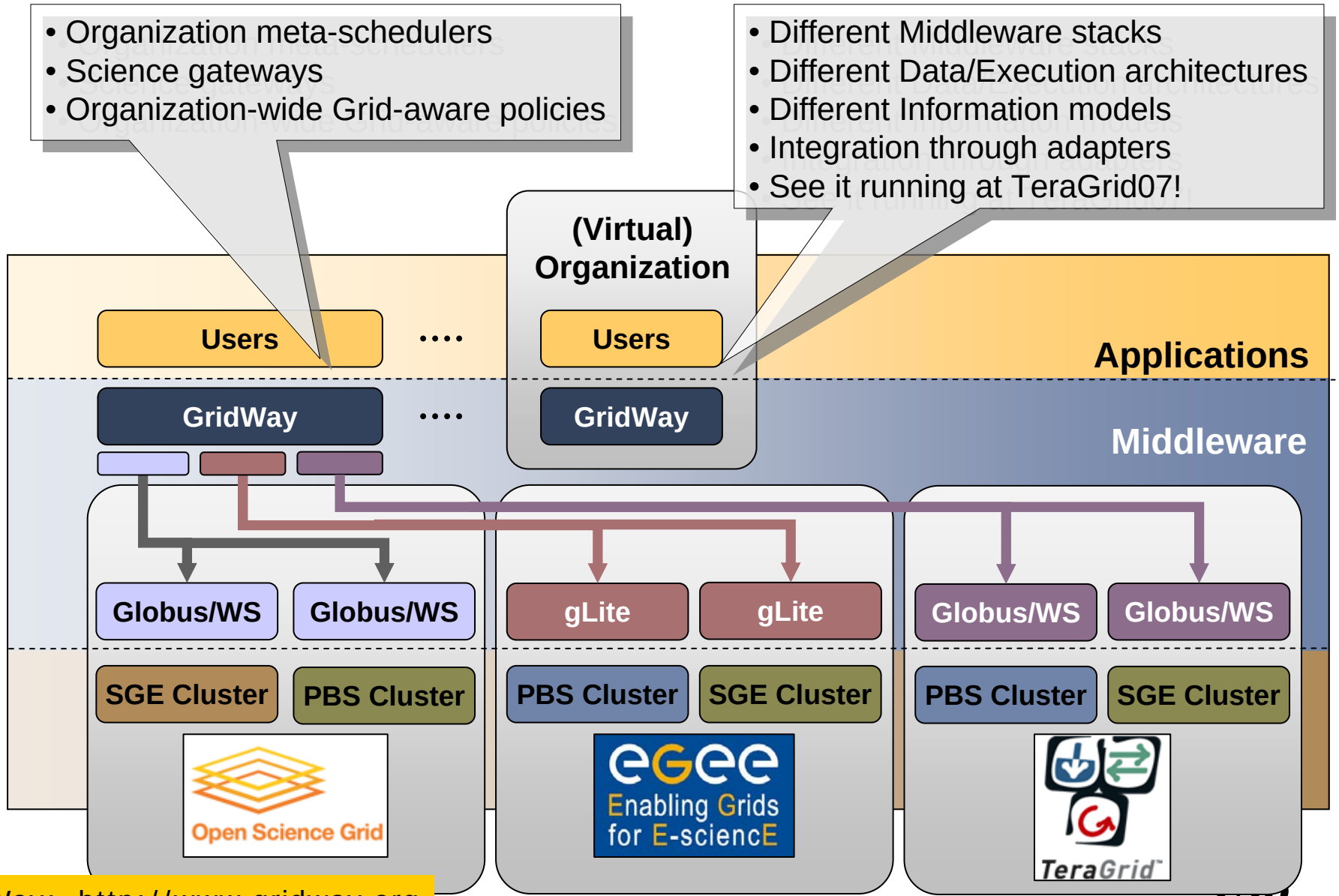

DRMAA library
CLI
GridWay Core
Request Manager
Job Pool
Host Pool
Dispatch Manager
Scheduler
Job Submission
Job Monitoring
Job Control
Job Migration
Transfer Manager
Execution Manager
Information Manager
GridFTP
RFT
pre-WS GRAM
WS GRAM
MDS2
MDS2 GLUE
MDS4
Job Preparation
Job Termination
Job Migration
Resource Discovery
Resource Monitoring
Grid File Transfer Services
Grid Execution Services
Grid Information Services

GridWay 5.2 Features

- Workload management
  - Advanced (Grid-specific) scheduling policies
  - Fault detection & recovery
  - Accounting
  - Array jobs, DAG workflows, and MPI jobs
- User Interface
  - OGF standards: JSDL (POSIX Profile) & DRMAA (C and JAVA)
  - Analysis of trends in resource usage
  - Command line interface, similar to that found on local LRM Systems
  - Easier installation through the auto-tools framework

Next Release GridWay 5.2.1

Interoperability: OSG, EGEE, TG & NorduGrid



Summary so far

- GRAM gives you a unified way to interface to different LRMs
- GridWay can be used to make the higher level scheduling decisions

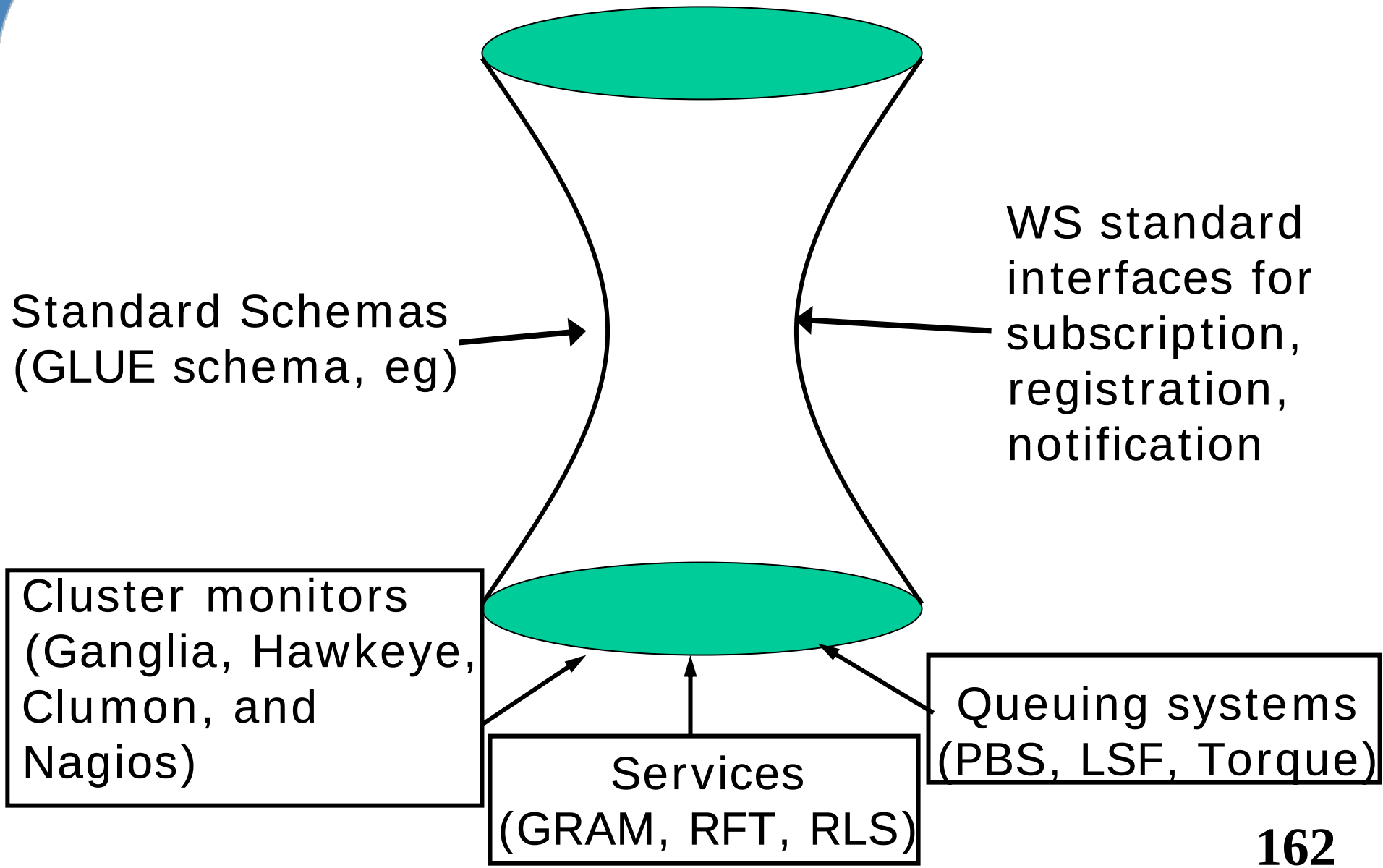
More Specifically, I May Want To …

- Create a service for use by my colleagues
- Manage who is allowed to access my service (or my experimental data or …)
- Ensure reliable & secure distribution of data from my lab to my partners
- Run 10,000 jobs on whatever computers I can get hold of
- Monitor the status of the different resources to which I have access

Monitoring and Discovery System (MDS4)

- Grid-level monitoring system
  - Aid user/agent to identify host(s) on which to run an application
  - Warn on errors
- Uses standard interfaces to provide publishing of data, discovery, and data access, including subscription/notification
  - WS-ResourceProperties, WS-BaseNotification, WS-ServiceGroup
- Functions as an hourglass to provide a common interface to lower-level monitoring tools

Information Users :
Schedulers, Portals, Warning Systems, etc.



MDS4 Components

- Information providers
  - Monitoring is a part of every WSRF service
  - Non-WS services are also be used
- Higher level services
  - Index Service – a way to aggregate data
  - Trigger Service – a way to be notified of changes
  - Both built on common aggregator framework
- Clients
  - WebMDS
- All of the tool are schema-agnostic, but interoperability needs a well-understood common language

Information Providers

- Data sources for the higher-level services
- Some are built into services
  - Any WSRF-compliant service publishes some data automatically
  - WS-RF gives us standard Query/Subscribe/Notify interfaces
  - Globus services: ServiceMetaDataInfo element includes start time, version, and service type name
  - Most of them also publish additional useful information as resource properties

Information Providers: Globus Services

- Reliable File Transfer Service (RFT)
  - Service status data, number of active transfers, transfer status, information about the resource running the service
- Community Authorization Service (CAS)
  - Identifies the VO served by the service instance
- Replica Location Service (RLS)
  - Note: not a WS
  - Location of replicas on physical storage systems (based on user registrations) for later queries

Information Providers

- Other sources of data
  - Any executables
  - Other (non-WS) services
  - Interface to another archive or data store
  - File scraping
- Just need to produce a valid XML document

Information Providers: Cluster and Queue Data

- Interfaces to Hawkeye, Ganglia, CluMon, Nagios
  - Basic host data (name, ID), processor information, memory size, OS name and version, file system data, processor load data
  - Some condor/cluster specific data
  - This can also be done for sub-clusters, not just at the host level
- Interfaces to PBS, Torque, LSF
  - Queue information, number of CPUs available and free, job count information, some memory statistics and host info for head node of cluster

Higher-Level Services

- Index Service
  - Caching registry
- Trigger Service
  - Warn on error conditions
- Archive Service
  - Database store for history (in development)

- All of these have common needs, and are built on a common framework

Common Aggregator Framework

- Basic framework for higher-level functions
  - Subscribe to Information Provider(s)
  - Do some action
  - Present standard interfaces

Aggregator Framework Features

1) Common configuration mechanism
   - Specify what data to get, and from where
2) Self cleaning
   - Services have lifetimes that must be refreshed
3) Soft consistency model
   - Published information is recent, but not guaranteed to be the absolute latest
4) Schema Neutral
   - Valid XML document needed only

MDS4 Index Service

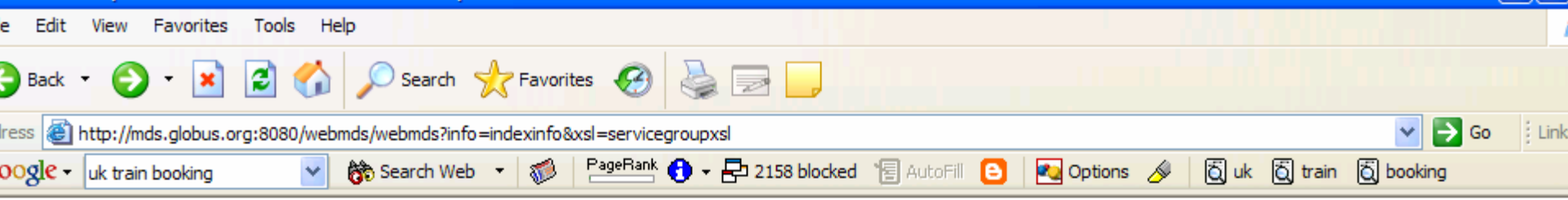
- Index Service is both registry and cache
  - Datatype and data provider info, like a registry (UDDI)
  - Last value of data, like a cache
- In memory default approach
  - DB backing store currently being developed to allow for very large indexes
- Can be set up for a site or set of sites, a specific set of project data, or for user-specific data only
- Can be a multi-rooted hierarchy
  - No *global* index

MDS4 Trigger Service

- Subscribe to a set of resource properties
- Evaluate that data against a set of pre-configured conditions (triggers)
- When a condition matches, action occurs
  - Email is sent to pre-defined address
  - Website updated
- Similar functionality in Hawkeye

WebMDS User Interface

- Web-based interface to WSRF resource property information
- User-friendly front-end to Index Service
- Uses standard resource property requests to query resource property data
- XSLT transforms to format and display them
- Customized pages are simply done by using HTML form options and creating your own XSLT transforms
- Sample page:
  - http://mds.globus.org:8080/webmds/webmds?info=indexinfo&xsl=servicegroupxsl



erviceGroup Overview

s page provides a brief overview of Web Services and/or WS-Resources that are members of a WS-ServiceGroup.

s WS-ServiceGroup has 4 direct entries, 33 in whole hierarchy.

esource Type	ID	Information	
Unknown	128.9.72.106	Aggregator entry with no content from `https://128.9.72.106:8443/wsrf/services/ReliableFileTransferFactoryService`	detail
GRAM	128.9.72.106	0 queues, submitting to 0 cluster(s) of 0 host(s).	detail
erviceGroup	128.9.72.140	This WS-ServiceGroup has 11 direct entries, 29 including descendants.	detail
erviceGroup	128.9.72.178	This WS-ServiceGroup has 4 direct entries, 4 including descendants.	detail
RFT	128.9.72.178	0 active transfer resources, transferring 0 files. 40.55 GB transferred in 173769 files since start of database.	detail
GRAM	128.9.72.178	0 queues, submitting to 1 cluster(s) of 10 host(s).	detail
GRAM	128.9.72.178	1 queues, submitting to 1 cluster(s) of 10 host(s).	detail
GRAM	128.9.72.178	2 queues, submitting to 1 cluster(s) of 10 host(s).	detail
erviceGroup	128.9.72.106	This WS-ServiceGroup has 3 direct entries, 3 including descendants.	detail
GRAM	128.9.72.106	0 queues, submitting to 0 cluster(s) of 0 host(s).	detail
GRAM	128.9.72.106	1 queues, submitting to 0 cluster(s) of 0 host(s).	detail
RFT	128.9.72.106	0 active transfer resources, transferring 0 files. 8.28 GB transferred in 8595 files since start of database.	detail
erviceGroup	128.9.64.179	This WS-ServiceGroup has 4 direct entries, 4 including descendants.	detail
GRAM	128.9.64.179	1 queues, submitting to 1 cluster(s) of 15 host(s).	detail
GRAM	128.9.64.179	5 queues, submitting to 1 cluster(s) of 15 host(s).	detail
RFT	128.9.64.179	0 active transfer resources, transferring 0 files. 63.16 GB transferred in 108704 files since start of database.	detail
GRAM	128.9.64.179	0 queues, submitting to 1 cluster(s) of 15 host(s).	detail
erviceGroup	128.9.128.168	This WS-ServiceGroup has 3 direct entries, 3 including descendants.	detail
GRAM	128.9.128.168	0 queues, submitting to 0 cluster(s) of 0 host(s).	detail
RFT	128.9.128.168	0 active transfer resources, transferring 0 files. 10.52 GB transferred in 23489 files since start of database.	detail

Working with TeraGrid

- Large US project across 9 different sites
  - Different hardware, queuing systems and lower level monitoring packages
- Starting to explore MetaScheduling approaches
- Need a common source of data with a standard interface for basic scheduling info

Data Collected

- Provide data at the subcluster level
  - Sys admin defines a subcluster, we query one node of it to dynamically retrieve relevant data
- Can also list per-host details
- Interfaces to Ganglia, Hawkeye, CluMon, and Nagios available now
  - Other cluster monitoring systems can write into a .html file that we then scrape
- Also collect basic queuing data, some TeraGrid specific attributes

Queue Overview

Name	UniqueId	Gram Information			LRMS		CPUs		Status	Jobs			Policy Limits			
		Version	Host	Port/URL	Type	Version	Total	Free		Total	Running	Waiting	Wall Clock Time	CPU Time	Total Jobs	Running Jobs
big	big	4.0.1	tg-login1.ncsa.teragrid.org	2019	PBS-Torque	2.0.0p7	891	538	enabled	0	0	0	2880	-1	-1	-1
dque	dque	4.0.1	tg-login1.ncsa.teragrid.org	2019	PBS-Torque	2.0.0p7	891	538	enabled	171	50	121	1440	-1	-1	-1
long	long	4.0.1	tg-login1.ncsa.teragrid.org	2019	PBS-Torque	2.0.0p7	891	538	enabled	0	0	0	5760	-1	-1	-1
priority	priority	4.0.1	tg-login1.ncsa.teragrid.org	2019	PBS-Torque	2.0.0p7	891	538	enabled	0	0	0	1440	-1	-1	-1
debug	debug	4.0.1	tg-login1.ncsa.teragrid.org	2019	PBS-Torque	2.0.0p7	891	538	enabled	0	0	0	30	-1	-1	-1
quake	quake	4.0.1	tg-login1.ncsa.teragrid.org	2019	PBS-Torque	2.0.0p7	891	538	enabled	4	0	4	2880	-1	-1	-1
gpfs-wan	gpfs-wan	4.0.1	tg-login1.ncsa.teragrid.org	2019	PBS-Torque	2.0.0p7	891	538	enabled	0	0	0	1440	-1	-1	-1

Cluster / Subcluster Overview

Type	Name	UniqueID	Processor		Total Memory	Operating System	SMP Size	Storage Device			TeraGrid Extensions
			Type	Clock Speed				Name	Size	Available Space	Total Nodes
Cluster	NCSA-TeraGrid	NCSA-TG									891
SubCluster	NCSA-TG-IA64CPU13-FASTIO-HIMEM	IA64CPU13-FASTIO-HIMEM	IA-64	1296	4061	Linux2.4.21.SuSE_292.til#1 SMP Fri Jun 3 07	2	entire-system	353385	91439	128
SubCluster	NCSA-TG-IA64CPU13-FASTIO-LOMEM.ncsa.teragrid.org	IA64CPU13-FASTIO-LOMEM	IA-64	1296	4101	Linux2.4.21.SuSE_292.til#1 SMP Fri Jun 3 07	2	entire-system	353384	91435	128
SubCluster	NCSA-TG-IA64CPU15-FASTCPU-GPFSWAN.ncsa.teragrid.org	IA64CPU15-FASTCPU-GPFSWAN	IA-64	1496	4106	Linux2.4.21.SuSE_292.til#1 SMP Fri Jun 3 07	2	entire-system	260036	10620	16
SubCluster	NCSA-TG-IA64CPU15-FASTCPU.ncsa.teragrid.org	IA64CPU15-FASTCPU	IA-64	1496	4106	Linux2.4.21.SuSE_292.til#1 SMP Fri Jun 3 07	2	entire-system	260036	10619	615
SubCluster	NCSA-TG-IA64CPU13-FASTIO-HIMEM-SPARE	IA64CPU13-FASTIO-HIMEM-SPARE	IA-64	1296	4056	Linux2.4.21.SuSE_292.til#1 SMP Fri Jun 3 07	2	entire-system	353372	91423	1
SubCluster	NCSA-TG-IA64CPU13-FASTIO-LOMEM-SPARE	IA64CPU13-FASTIO-LOMEM-SPARE	IA-64	1296	4061	Linux2.4.21.SuSE_292.til#1 SMP Fri Jun 3 07	2	entire-system	353385	91439	1
SubCluster	NCSA-TG-IA64CPU15-PHASE2-FASTCPU-SPARE2	IA64CPU15-PHASE2-FASTCPU-SPARE2	IA-64	1496	4106	Linux2.4.21.SuSE_292.til#1 SMP Fri Jun 3 07	2	entire-system	260036	10620	2

Hosts in Subcluster NCSA-TG-IA64CPU13-FASTIO-HIMEM

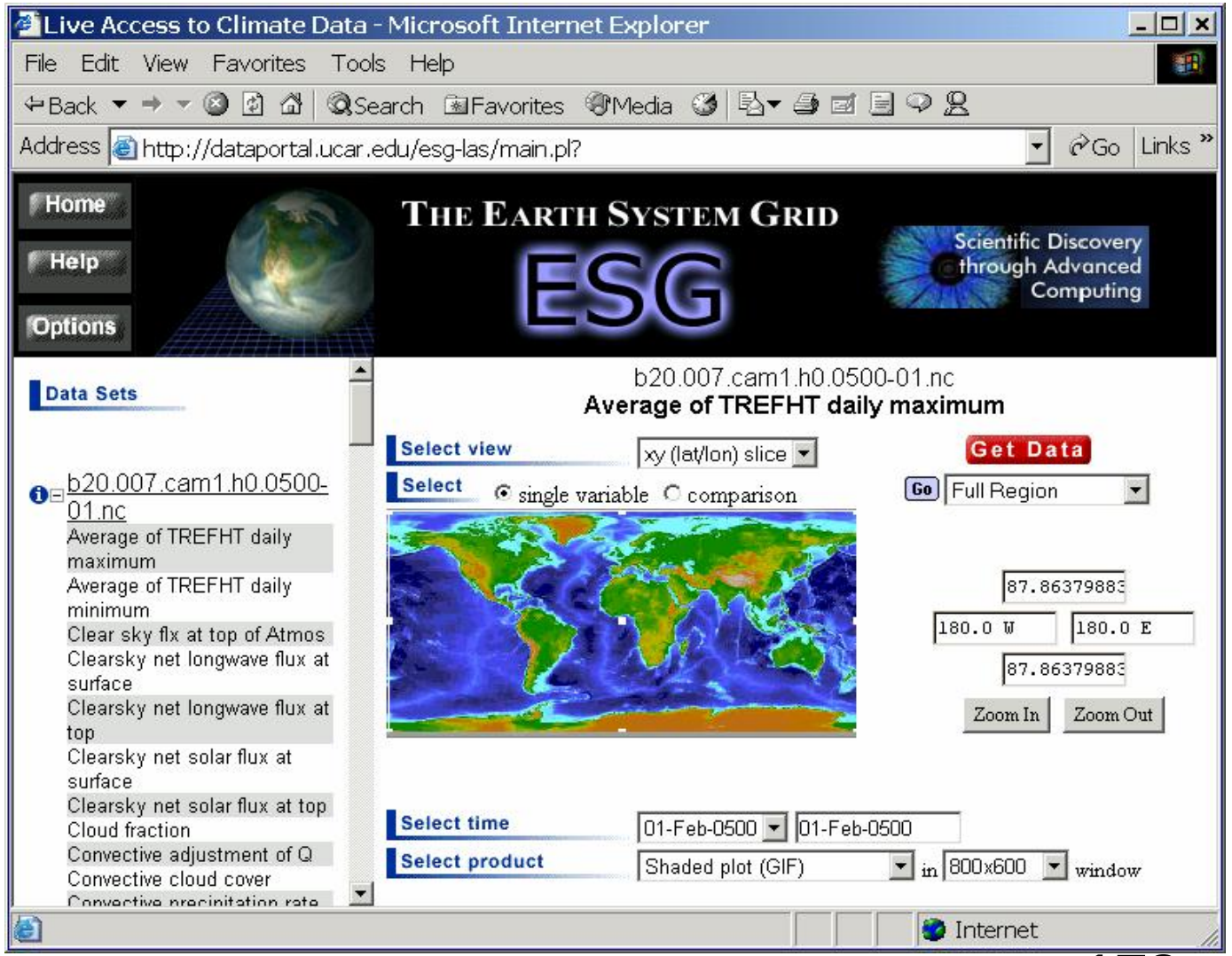
Name	UniqueId	TeraGrid Extensions
		Node Properties
tg-c001.ncsa.teragrid.org	tg-c001	all,ia64-compute,compute,ia64-cpu13,fastio,himem,rack40,clos12,stage
tg-c002.ncsa.teragrid.org	tg-c002	all,ia64-compute,compute,ia64-cpu13,fastio,himem,rack40,clos12
tg-c003.ncsa.teragrid.org	tg-c003	all,ia64-compute,compute,ia64-cpu13,fastio,himem,rack40,clos12
tg-c004.ncsa.teragrid.org	tg-c004	all,ia64-compute,compute,ia64-cpu13,fastio,himem,rack40,clos12
tg-c005.ncsa.teragrid.org	tg-c005	all,ia64-compute,compute,ia64-cpu13,fastio,himem,rack40,clos12

the globus alliance
www.globus.org

Earth System Grid

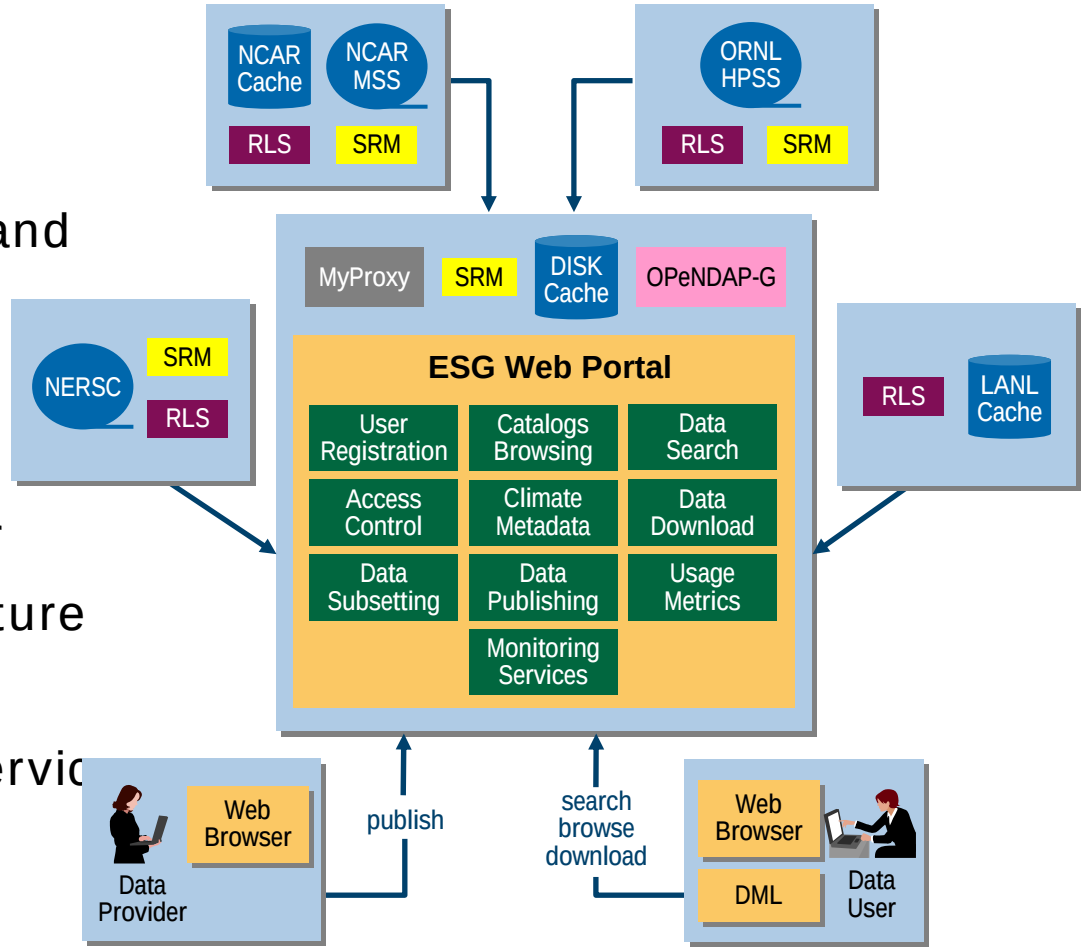
DOE Earth System Grid

Goal: Enable sharing & analysis of high-volume data from advanced earth system models



ESG Technologies

- Climate data
  - Metadata catalog
  - OPenDAP-G (aggregation and subsetting)
- Data management
  - Data Mover Lite
  - Storage Resource Manager
  - Globus Security Infrastructure
  - GridFTP
  - Globus Replica Location Servic
- Security services
  - Access control
  - MyProxy
  - PURSE User registration



MSS, HPSS: **Tertiary data storage systems**

Monitoring Overall System Status

- Monitored data are collected in MDS4 Index service
- Information providers check resource status at a configured frequency
  - Currently, every 10 minutes
- Report status to Index Service
- Information in Index Service is queried by ESG Web portal
- Used to generate overall picture of state of ESG resources
- Displayed on ESG Web portal page

ESG: Warning on Errors Sample

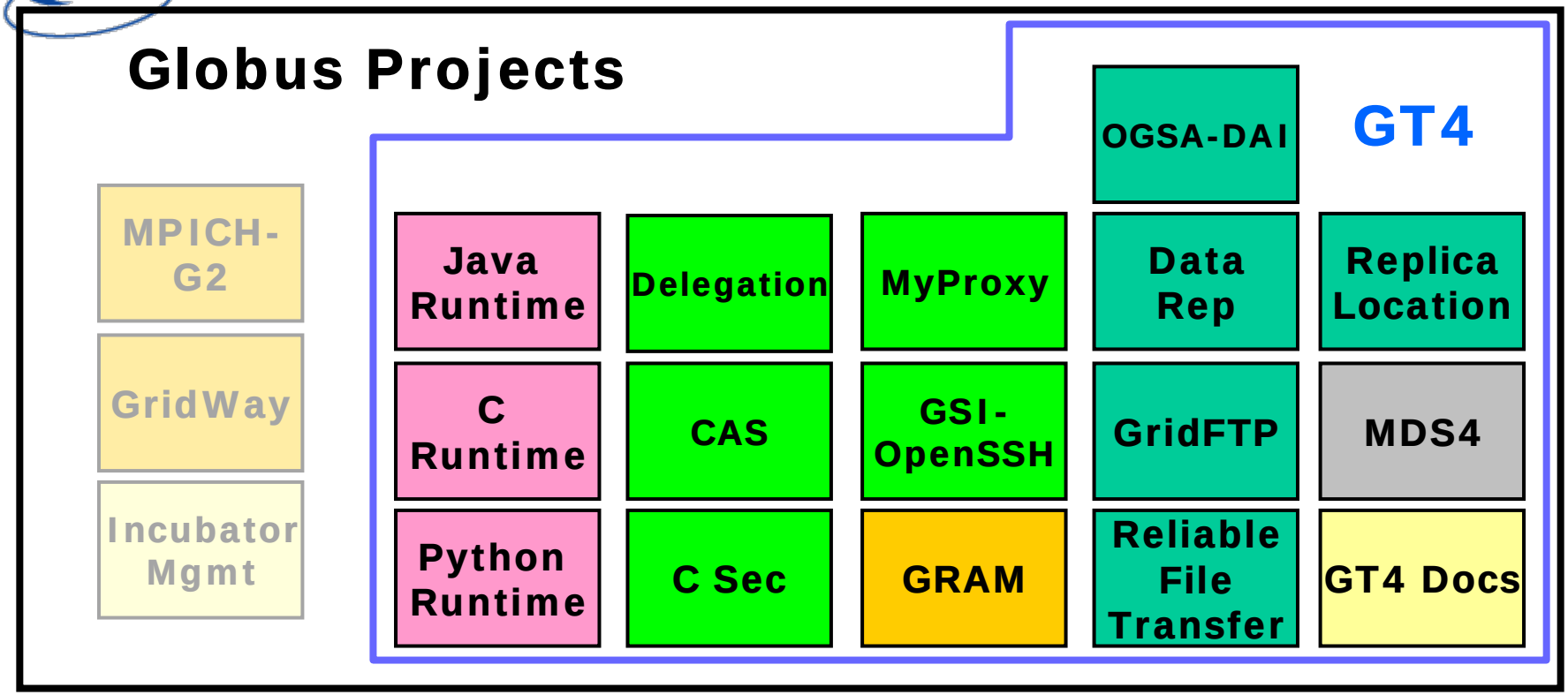
Total error messages for May 2006	**47**
Messages related to certificate and configuration problems at LANL	38
Failure messages due to brief interruption in network service at ORNL on 5/13	2
HTTP data server failure at NCAR 5/17	1
RLS failure at LLNL 5/22	1
Simultaneous error messages for SRM services at NCAR, ORNL, LBNL on 5/23	3
RLS failure at ORNL 5/24	1
RLS failure at LBNL 5/31	1

Summary to now

- MDS gives you a way to do resource discovery and warnings on errors

The GT4 Distribution

Globus Software: dev.globus.org



Incubator Projects

		GAARDS	MEDICUS	Cog WF	Virt WkSp		
GDTE	GridShib	OGRO	UGP	Dyn Acct	Gavia JSC	DDM	Metrics
Introduce	PURSE	HOC-SA	LRMA	WEEP	Gavia MS	SGGC	ServMark

Common Runtime | Security | Execution Mgmt | Data Mgmt | Info Services | Other

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

GT2 vs GT4

- Pre-WS Globus is in GT4 release
  - Both WS and pre-WS components (ala 2.4.3) are shipped
  - These do NOT interact, but both can run on the same resource independently
- Basic functionality is the same
  - Run a job
  - Transfer a file
  - Monitoring
  - Security
- Code base is completely different

Why Use GT4?

- Performance and reliability
  - Literally millions of tests and queries run against GT4 services
- Scalability
  - Many lessons learned from GT2 have been addressed in GT4
- Support
  - This is our active code base, much more attention
- Additional functionality
  - New features are here
  - Additional GRAM interfaces to schedulers, MDS Trigger service, GridFTP protocol interfaces, etc
- Easier to contribute to

Versioning and Support

- Versioning
  - Evens are production (4.0.x, 4.2.x),
  - Odds are development (4.1.x)
- We support this version and the one previous
  - Currently stable version 4.0.4
  - We support 3.2.x and 4.0.x
  - We've also got the 4.1.2 dev release available (1 June '07)

Several "Next" Versions

- 4.0.5 – stable release
  - 100% same interfaces
  - Expected mid June
- 4.1.3 – development release(s)
  - New functionality
  - Expected every 6-8 weeks (mid July)
- 4.2.0 - stable release
  - Tested, documented 4.1.x branch
  - Likely late summer or early fall
  - Discussed on gt-dev@globus.org
- 5.0 – substantial code base change
  - With any luck, not for years :)

Tested Platforms

- Debian
- Fedora Core
- FreeBSD
- HP/UX
- IBM AIX
- Red Hat
- Sun Solaris
- SGI Altix (IA64 running Red Hat)
- SuSE Linux
- Tru64 Unix
- Apple MacOS X (no binaries)
- Windows – Java components only

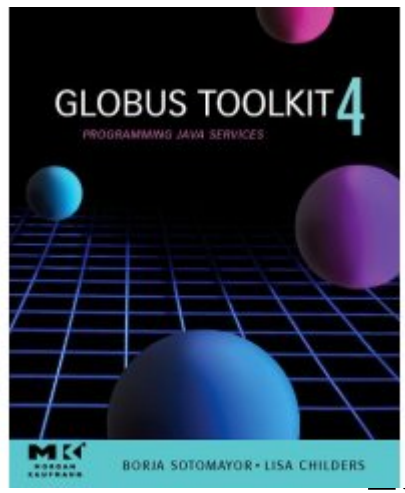
List of binaries and known platform-specific install bugs at

http://www.globus.org/toolkit/docs/4.0/admin/docbook/ ch03.html

Documentation Overview

- Current document significantly more detailed than earlier versions
  - http://www.globus.org/toolkit/docs/4.0/
- Tutorials available for those of you building a new service
  - http://www-unix.globus.org/toolkit/tutorials/BAS/

- Globus® Toolkit 4: Programming Java Services (The Morgan Kaufmann Series in Networking), by Borja Sotomayor, Lisa Childers (Available through Amazon, £19.99 or $20)



Installation in a nutshell

- Quickstart guide is very useful

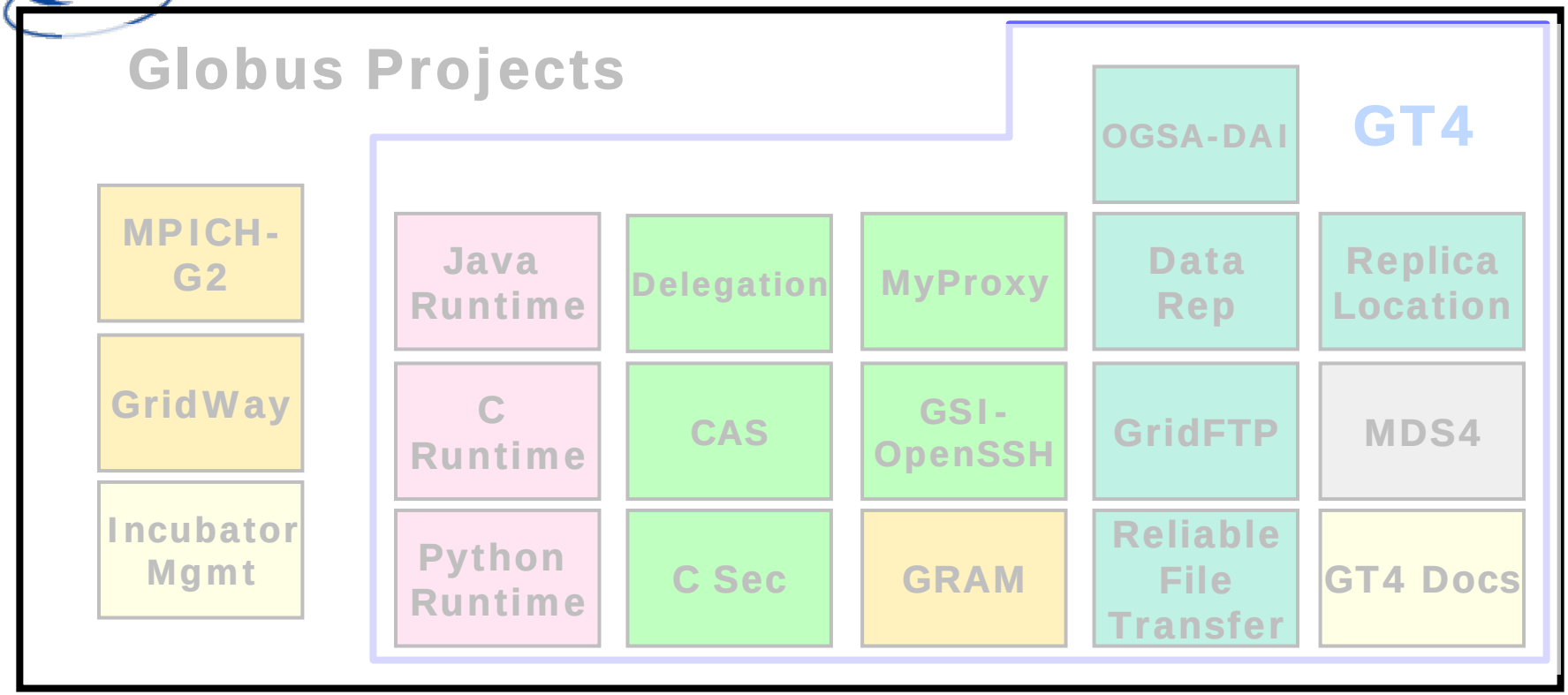
  http://www.globus.org/toolkit/docs/4.0/admin/docbook/quickstart.html

- Verify your prereqs!
- Security – check spellings and permissions
- Globus is system software – plan accordingly

General Globus Help and Support

- Globus toolkit help lists list
  - gt-user@globus.org
  - gt-dev@globus.org
  - http://dev.globus.org/wiki/Mailing_Lists
- Each project has specific lists
- Bugzilla
  - bugzilla.globus.org

Globus Software: dev.globus.org



Incubator Projects

		GAARDS	MEDICUS	Cog WF	Virt WkSp		
GDTE	GridShib	OGRO	UGP	Dyn Acct	Gavia JSC	DDM	Metrics
Introduce	PURSE	HOC-SA	LRMA	WEEP	Gavia MS	SGGC	ServMark

Common Runtime | Security | Execution Mgmt | Data Mgmt | Info Services | Other

Incubator Process in dev.globus

- Entry point for new Globus projects
- Incubator Management Project (IMP)
  - Oversees incubator process form first contact to becoming a Globus project
  - Quarterly reviews of current projects
  - Process being debugged by “Incubator Pioneers”

http://dev.globus.org/wiki/Incubator/Incubator_Process

Incubator Process (1 of 3)

- Project proposes itself as a *Candidate*
  - A proposed name for the project;
  - A proposed project chair, with contact info;
  - A list of the proposed committers for the project;
  - An overview of the aims of the project;
  - An overview of any current user base or user community, if applicable;
  - An overview of how the project relates to other parts of Globus;
  - A summary of why the project would enhance and benefit Globus.

Incubator Process (2 of 3)

- IMP meet, discuss, and accept project as a *Incubator Project*
  - Project is now part of the Incubator framework
  - Get assigned a Mentor to help
    - Member of IMP
    - Bridge between Globus and new Incubator Project
  - Opportunity to get up to speed on Globus Development process

Incubator Process (3 of 3)

- Quarterly reviews by IMP determine
  - Stay an Incubator Project
  - Retire
  - Escalate to a full Globus project
- Escalation when Project passes checklist
  - Legal
  - Meritocracy
  - Alignment/Synergy
  - Infrastructure

Current Incubator Projects dev.globus.org

- Distributed Data Management (DDM)
- Dynamic Accounts
- Gavia-Meta Scheduler
- Gavia- Job Submission Client
- Grid Authentication and Authorization with Reliably Distributed Services (GAARDS)
- Grid Development Tools for Eclipse (GDTE)
- GridShib
- Grid Toolkit Handle System (gt-hs)
- Higher Order Component Service Architecture (HOC-SA)
- Introduce
- Local Resource Manager Adaptors (LRMA)
- Metrics
- MEDICUS
- Open GRid OCSP (Online Certificate Status Protocol)
- Portal-based User Registration Service (PURSe)
- ServMark
- SJTU GridFTP GUI Client (SGGC)
- UCLA Grid Portal Software (UGP)
- WEEP
- Cog Workflow
- Virtual Workspaces

How Can You Contribute? Create a New Project

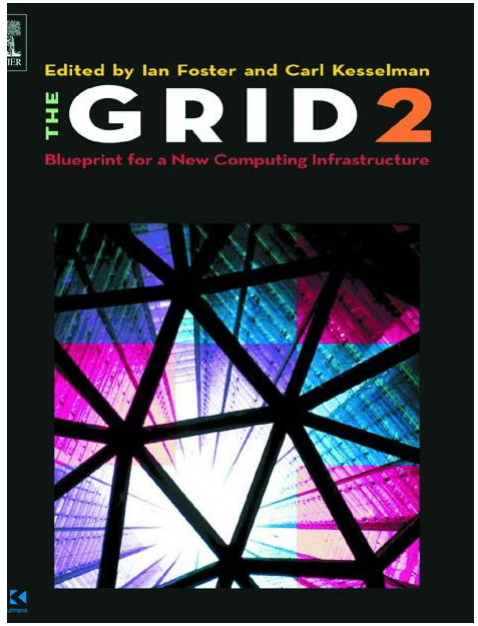
- Do you have a project you'd like to contribute?
- Does your software solve a problem you think the Globus community would be interested in?
- Contact incubator-committers@globus.org

Contribute to an Existing Project

- Contribute code, documentation, design ideas, and feature requests
- Joining the mailing lists
  - *-dev, *-user, *-announce for each project
  - See the project wiki page at dev.globus.org
- Chime in at any time
- Regular contributors can become committers, with a role in defining project directions

http://dev.globus.org/wiki/How_to_contribute

Summary: Grids are About …



Enabling *"coordinated resource sharing & problem solving in dynamic, multi-institutional virtual organizations."*

(Source: **"The Anatomy of the Grid"**)

- Access to shared resources
  → Virtualization, allocation, management
- With predictable behaviors
  → Provisioning, quality of service
- In dynamic, heterogeneous environments
  → Standards-based interfaces and protocols

… By Providing Open Infrastructure

- Web services standards
  - State, notification, security, …
- Services that enable access to resources
  - Service-enable new & existing resources
  - E.g., GRAM on computer, GridFTP on storage system, custom application services
  - Uniform abstractions & mechanisms
- Tools to build applications that exploit this infrastructure
  - Registries, security, data management, …
- A rich tool & service ecosystem

More Specifically, Making it Possible to …

- Create a service for use by my colleagues
- Manage who is allowed to access my service (or my experimental data or …)
- Ensure reliable & secure distribution of data from my lab to my partners
- Run 10,000 jobs on whatever computers I can get hold of
- Monitor the status of the different resources to which I have access
- And so on …

For More Information

- Jennifer Schopf
  - jms@mcs.anl.gov
  - http://www.mcs.anl.gov/~jms
- Globus Alliance
  - http://www.globus.org
- Dev.globus
  - http://dev.globus.org
- Upcoming Events
  - http://dev.globus.org/wiki/Outreach
- Globus Solutions
  - http://www.globus.org/solutions/

I should mention some open positions....

- Open positions as part of the Computation Institute, a joint institute between University of Chicago and Argonne National Laboratory
- Scientific research programmer, Grid computing research and applications
  - (requisition # 075544)
- Computer systems programmer, Grid computing research and workflow
  - (requisition # 075338)
- Software and senior software developers
  - (requisition #075287 and 074800)

http://jobs.uchicago.edu/, click "Job Opportunities" and search for requisition number listed

Globus at TG '07

June 5, Tuesday

- 5:30-6:30: Globus BOF

June 6, Wednesday

- 9:30-10, Portal-based User Registration Service (PURSe)
- 10-10:30, Resource Discovery on TeraGrid with MDS4
- 10:30-11, Globus 4 GRAM & GRAM Audit
- 11:30-12, GT4 GRAM: A Functionality and Performance Study
- 2:45-3:30, GridFTP/RFT/tgcp

June 7, Thursday

- 9:30-10 GridWay: A Metascheduler for Globus-based Grids

For This Afternoon

- Network Connection
- Web Browser
- SSH client
  - If you don't have an SSH client get one from

http://www.chiark.greenend.org.uk/%7esgtatham/putty/download.html

  - get putty.exe
- Class notes are at:

http://www.ci.uchicago.edu/~benc/tg07.html