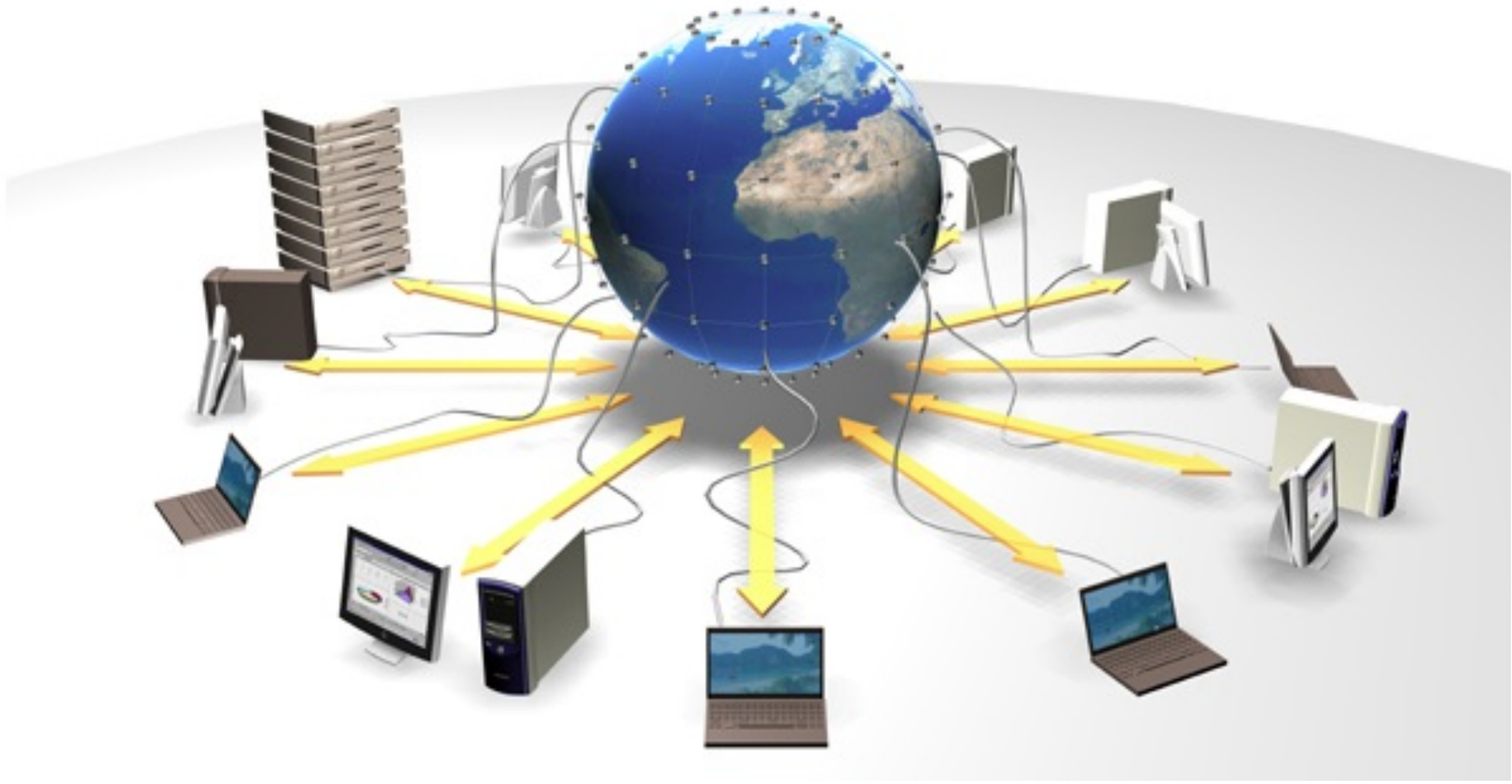
Grid Computing



Elements of Grid Computing

- Resource sharing
  - Computers, data, 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 organizations
  - Community overlays on classic org structures
  - Large or small, static or dynamic

Definitions

- Resource
  - An entity that may be shared
    - CPU, storage, data, software,…
  - Not necessarily a physical entity
    - Filesystem, bandwidth, thread pool…
  - Defined in terms of interfaces and capabilities
    - Open/close/read/write define the access methods to a filesystem
    - Copy/delete/move/create/cat define the methods to manipulate data

Virtual Organizations

- Components
  - set of individual/institutions
  - set of resources
  - set of sharing rules
- Dynamic set of individuals and/or institutions defined by a shared goal and a set of sharing rules
- May vary in size, scope, duration and structure
  - Example: class students for cooperative lecture writing
  - Example: industrial consortium building a new aircraft
- The sharing is highly controlled, with resource providers and consumers defining clearly and carefully just what is shared

Example of VOs

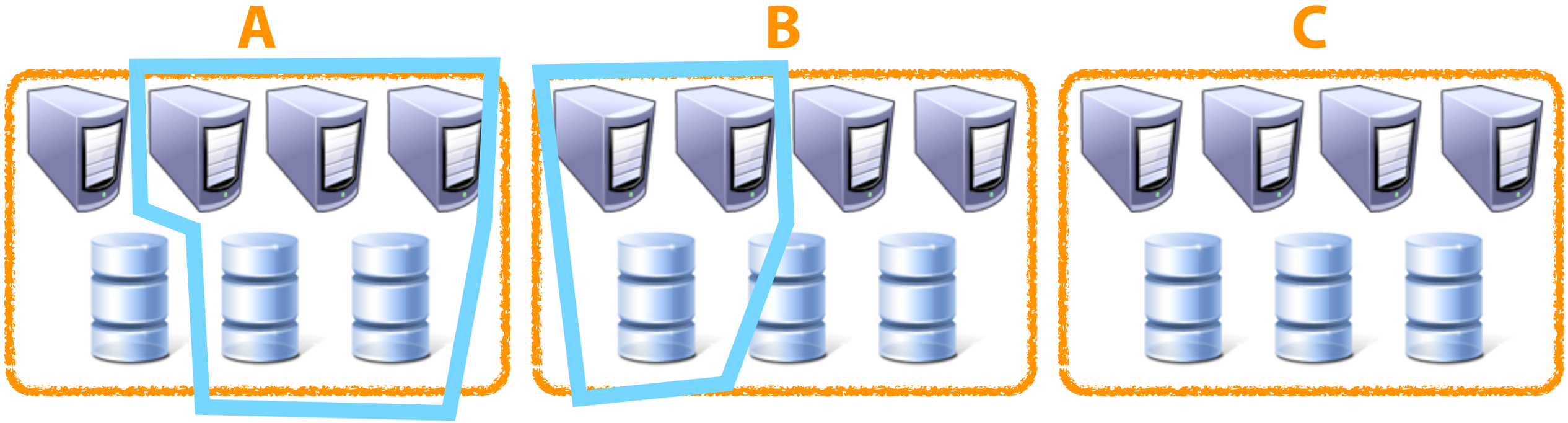
- Three physical organizations (A, B, C)
- Two virtual organizations (X, Y)

Example of VOs

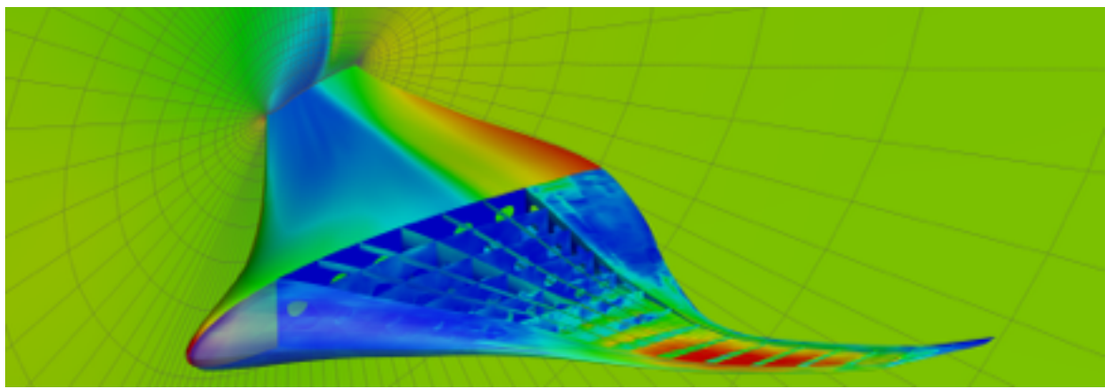
- Three physical organizations (A, B, C)
- Two virtual organizations (X, Y)

Example of VOs

- Three physical organizations (A, B, C)
- Two virtual organizations (X, Y)



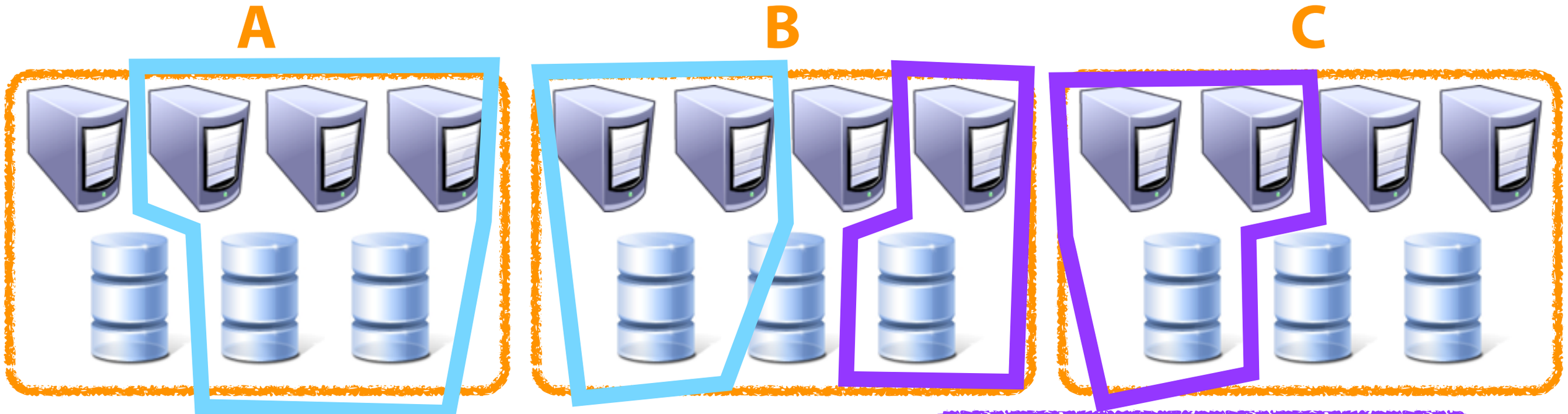
X



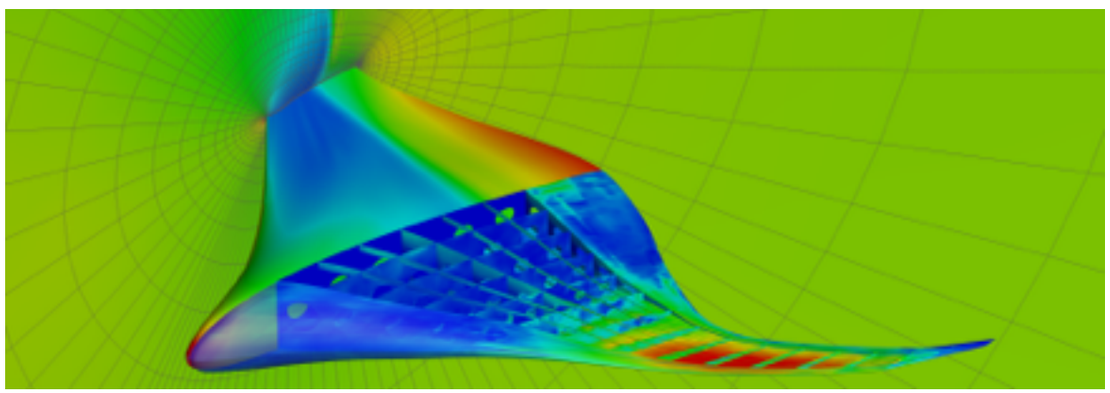
Multidisciplinary Design

Example of VOs

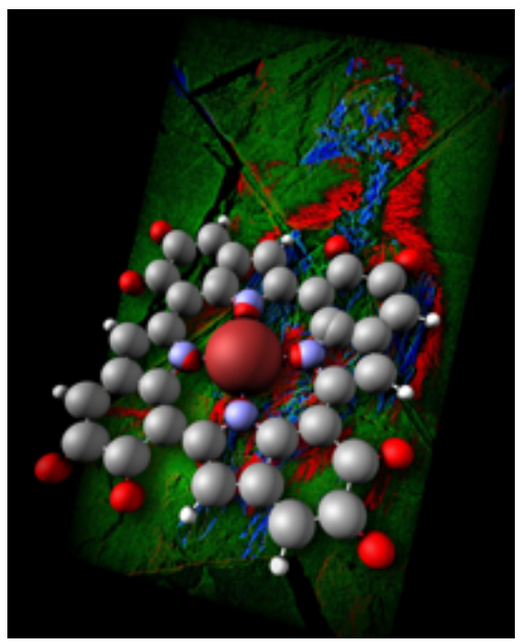
- Three physical organizations (A, B, C)
- Two virtual organizations (X, Y)



X



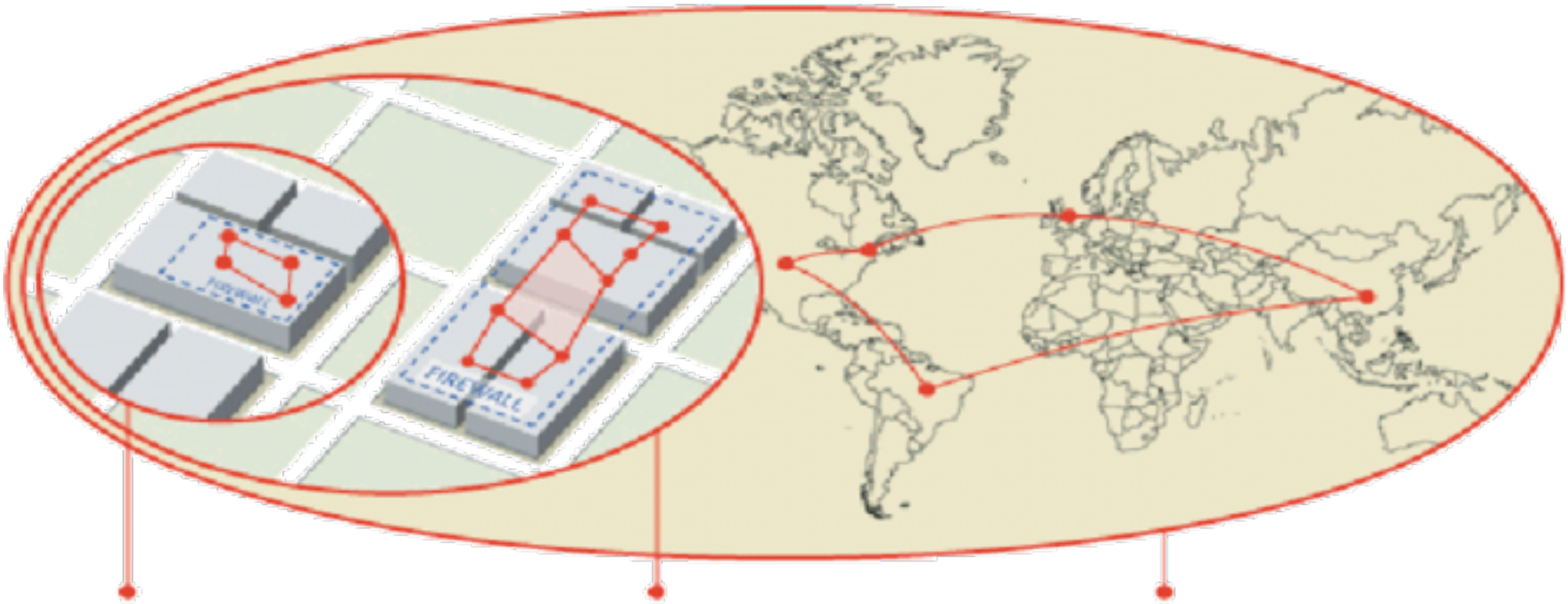
Multidisciplinary Design



Joint Drug Analysis

Y

Scope of Grids



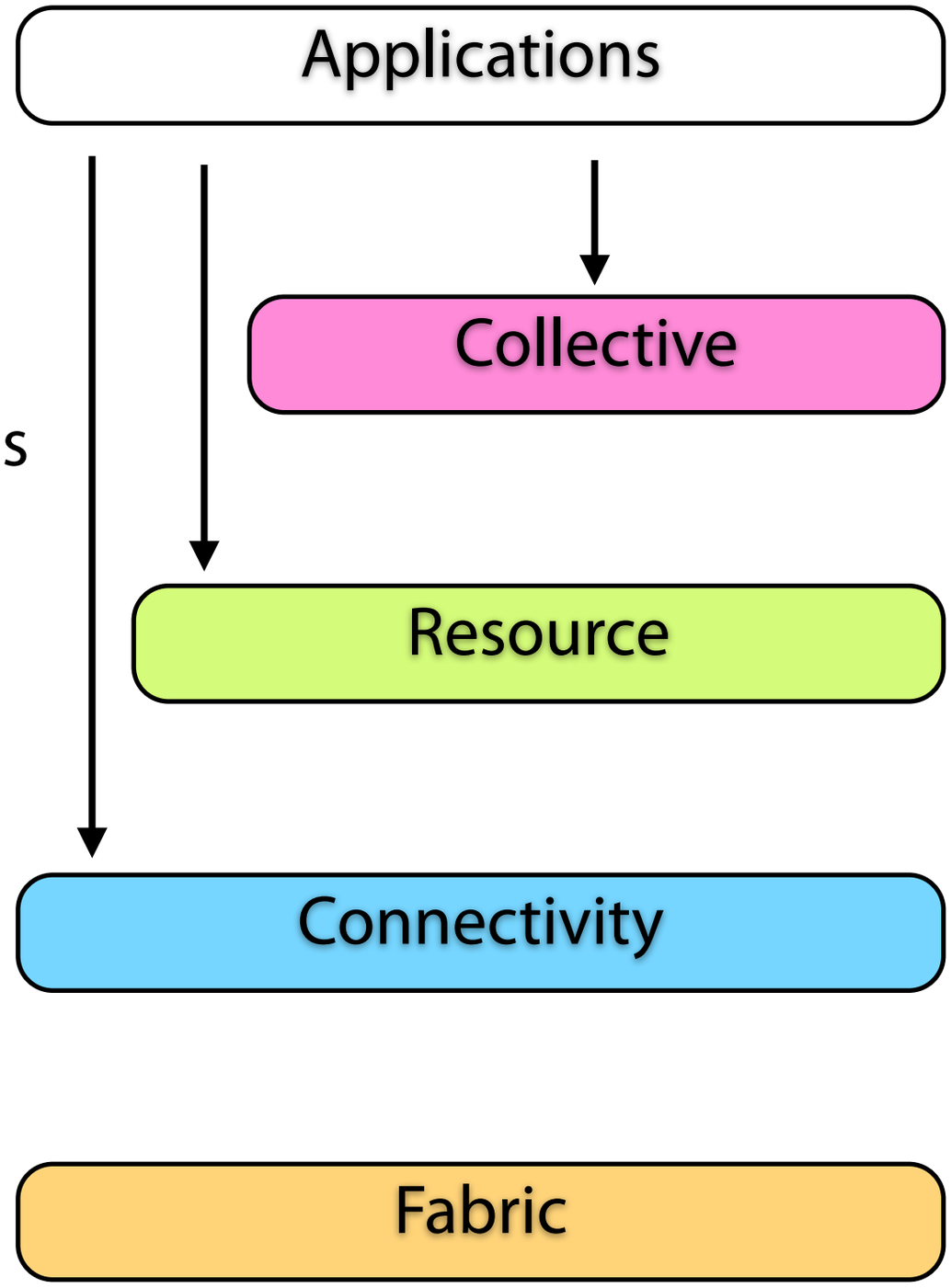
Grid Architecture

"Coordinating multiple resources":
Ubiquitous infrastructure services, application-specific distributed services

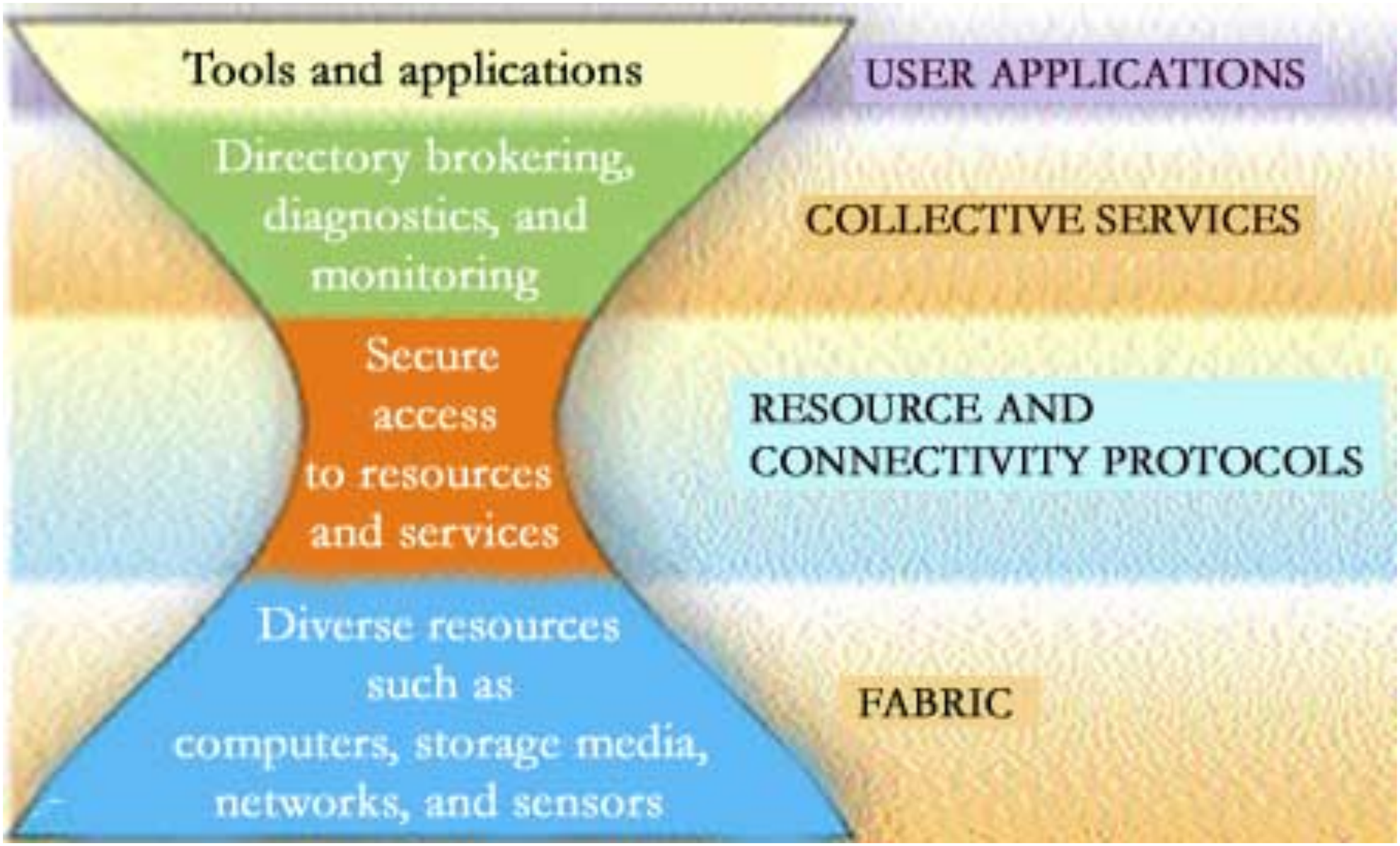
"Sharing single resources":
Negotiating access, controlling use

"Talking to things":
Communication (Internet protocols) & security

"Controlling things locally":
Access to & control of resources



The Hourglass Model



I. Foster, "The Grid: A New Infrastructure for 21st Century Science," Physics Today, vol. 55, no. 2, pp. 42–47, 2002.

Fabric Layer

- Just what you would expect: the diverse mix of resources that may be shared
  - Individual computers, Condor pools, file systems, archives, metadata catalogs, networks, sensors, etc.
- Few constraints on low-level technology: connectivity and resource level protocols form the “neck in the hourglass”
- Defined by interfaces not physical characteristics

Connectivity Layer

- Communication
  - Internet protocols: IP, DNS, routing, etc.
- Security: Grid Security Infrastructure (GSI)
  - Uniform authentication, authorization, and message protection mechanisms in multi-institutional setting
  - Single sign-on, delegation, identity mapping
  - Public key technology, SSL, X.509, GSS-API
  - Supporting infrastructure: Certificate Authorities, certificate & key management, …

Resource Layer

- Grid Resource Allocation Management (GRAM)
  - Remote allocation, reservation, monitoring, control of compute resources
- GridFTP protocol (FTP extensions)
  - High-performance data access & transport
- Grid Resource Information Service (GRIS)
  - Access to structure & state information
- Others emerging: Catalog access, code repository access, accounting, etc.
- All built on connectivity layer: GSI & IP

Collective Layer

- Index servers a.k.a. meta-directory services
  - Custom views on dynamic resource collections assembled by a community
- Resource brokers
  - Resource discovery and allocation
- Replica catalogs
- Replication services
- Co-reservation and co-allocation services
- Workflow management services
- etc…

Globus Toolkit

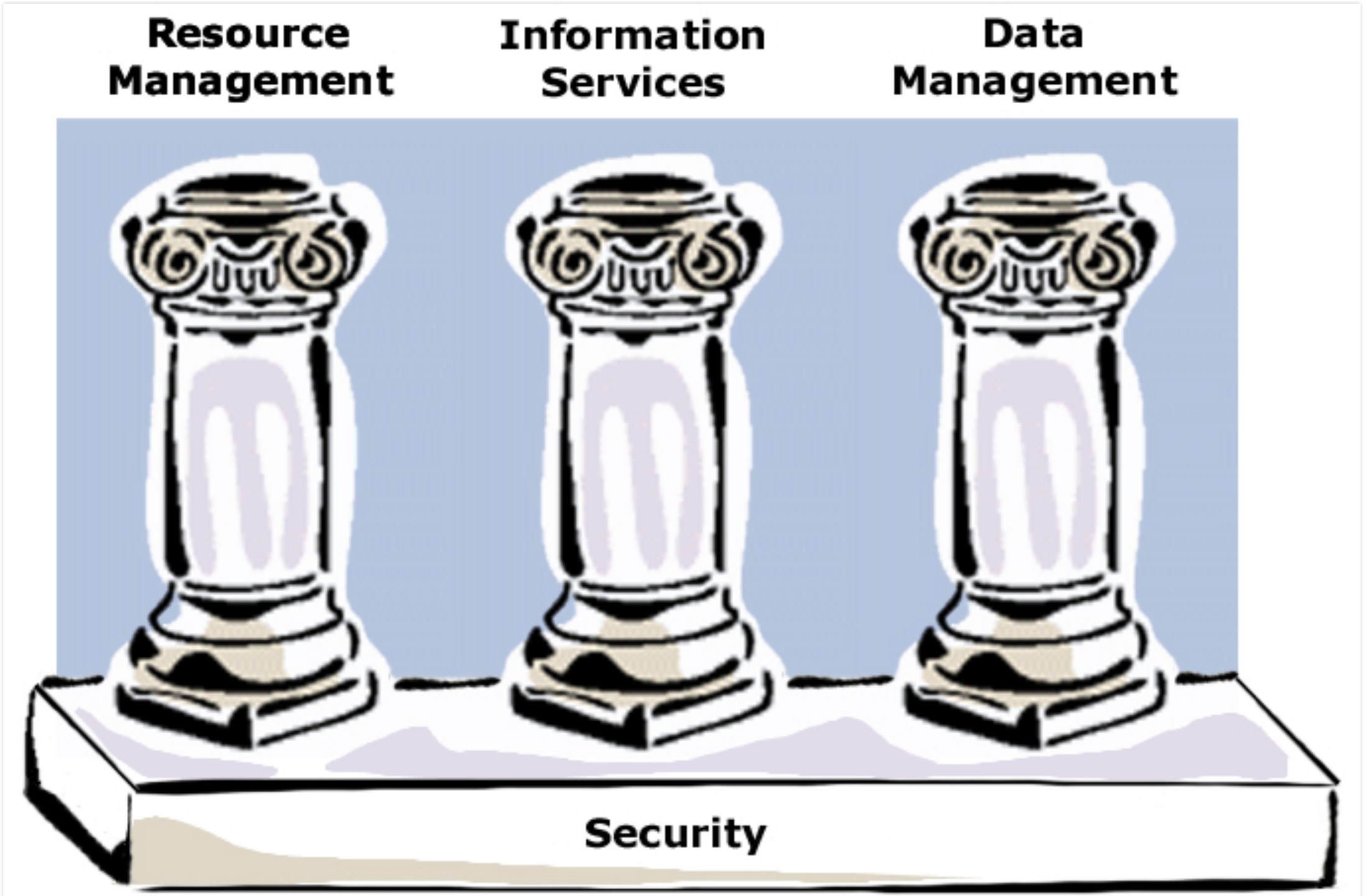
- An example Grid middleware

http://www.globus.org/toolkit/

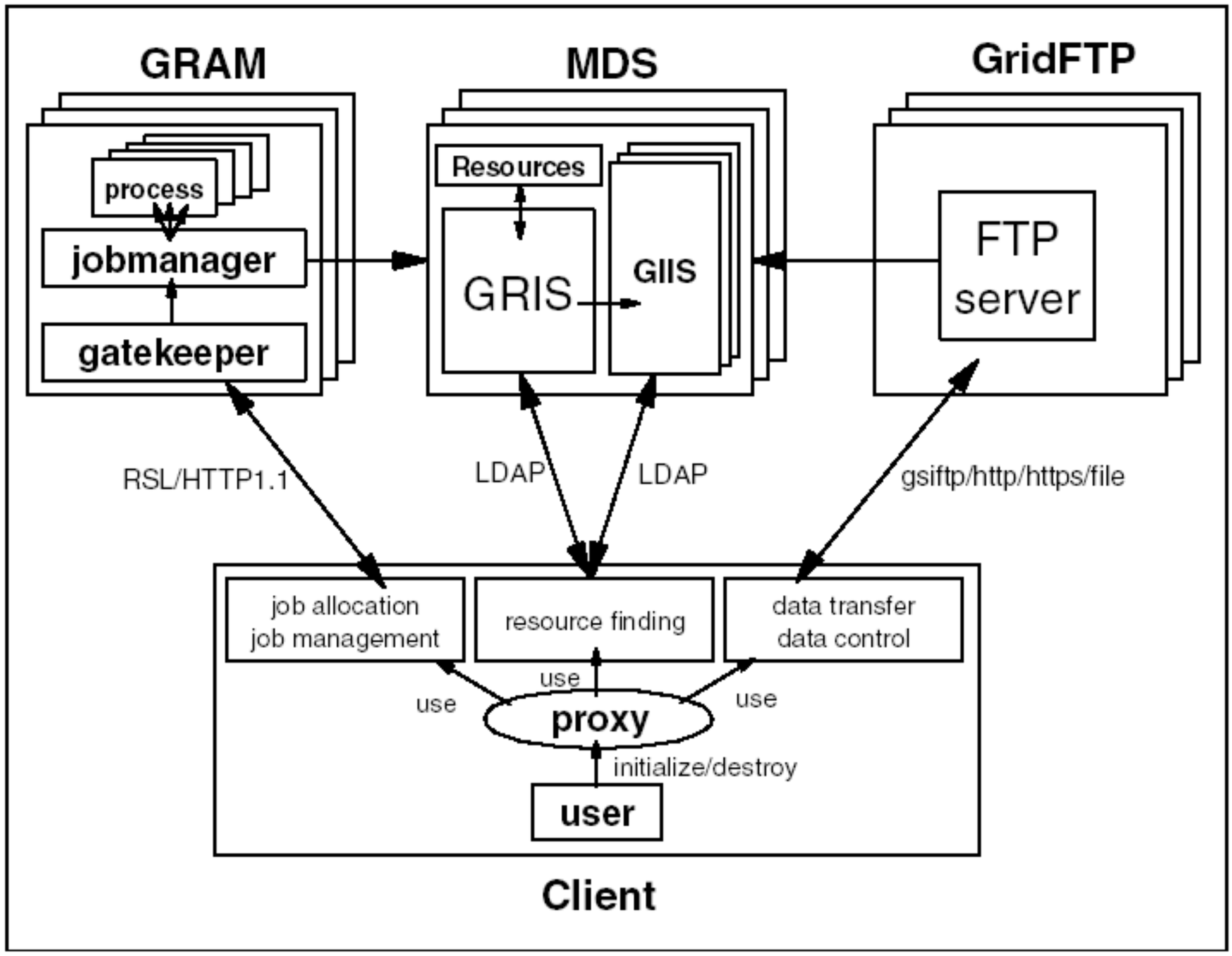
- A software toolkit addressing key technical problems in the development of Grid enabled tools, services, and applications
  - Offer a modular "bag of technologies"
  - Enable incremental development of Grid-enabled tools and applications
  - Implement standard Grid protocols and APIs (the "core" of the hourglass)
  - Is available under liberal open source license

- Now is evolving to Cloud middleware

Key Protocols



GT2 Protocols and Services

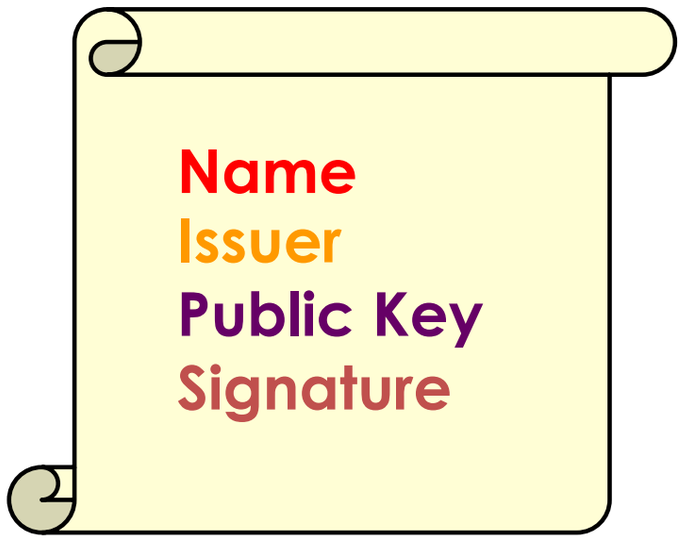


Grid Security

- Resources being used may be valuable & the problems being solved sensitive
- Resources are often located in distinct administrative domains
  - Each resource has own policies & procedures
- Set of resources used by a single computation may be large, dynamic, and unpredictable
  - Not just client/server, requires delegation
- It must be broadly available & applicable
- Standard, well-tested, well-understood protocols; integrated with wide variety of tools

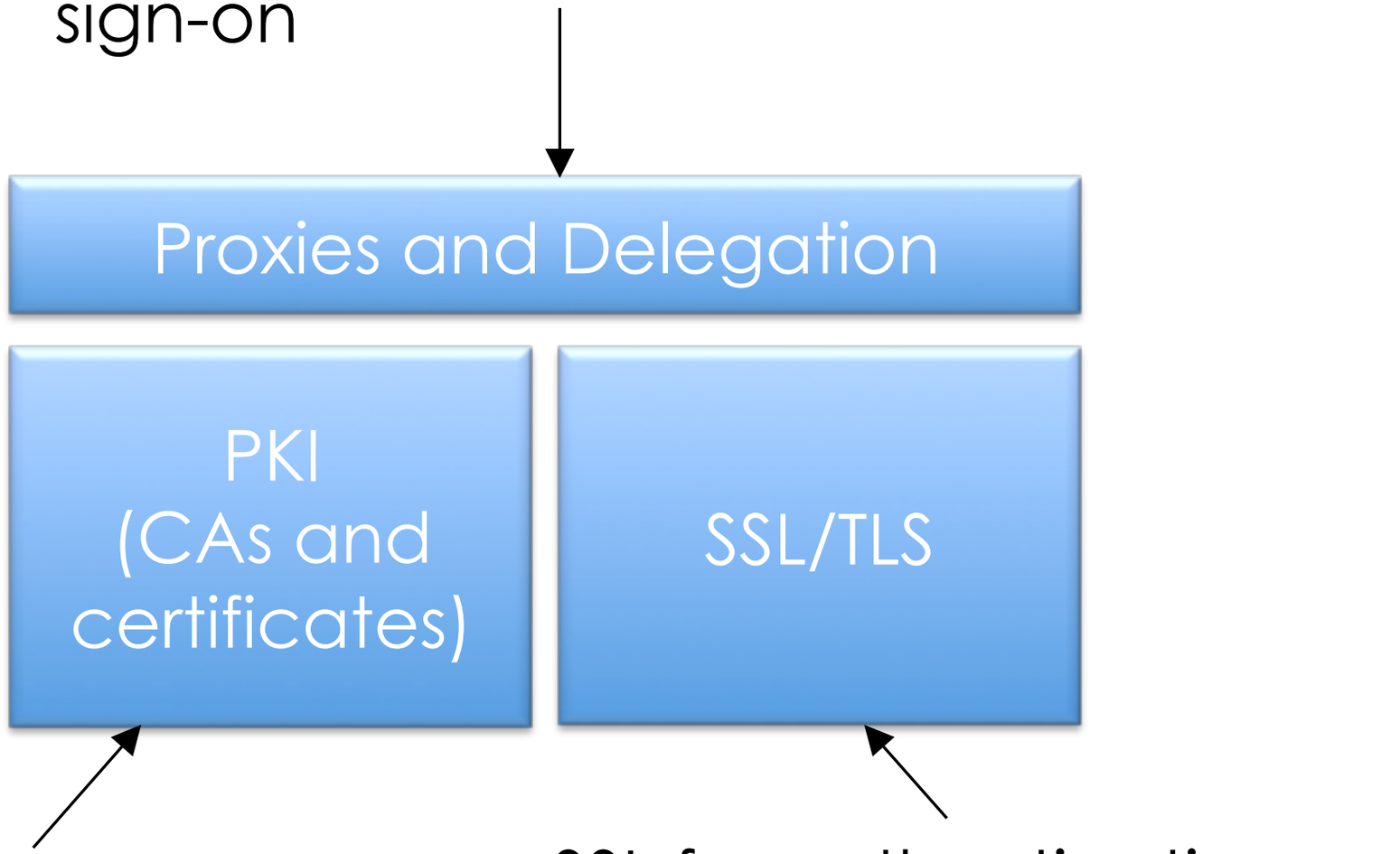
Public Key Infrastructure

- PKI allows you to know that a given public key belongs to a given user
- PKI builds upon asymmetric encryption:
  - Each entity has two keys: public and private
  - Data encrypted with one key can only be decrypted with the other
  - The private key is known only to the owner
- The public key is given to the world encapsulated in a X.509



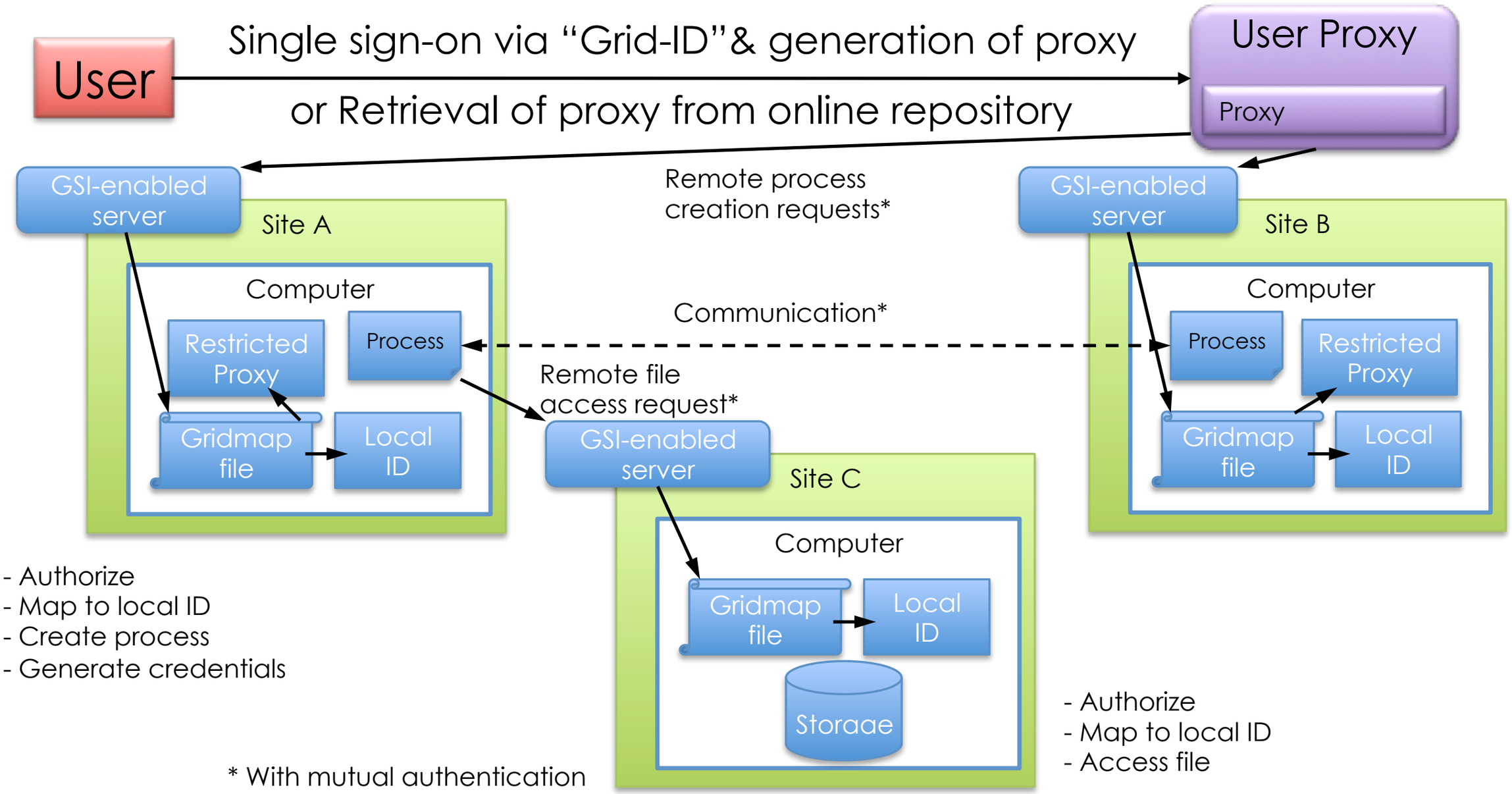
Grid Security Infrastructure

Proxies and delegation (GSI extensions) for secure single sign-on



GSI in Action

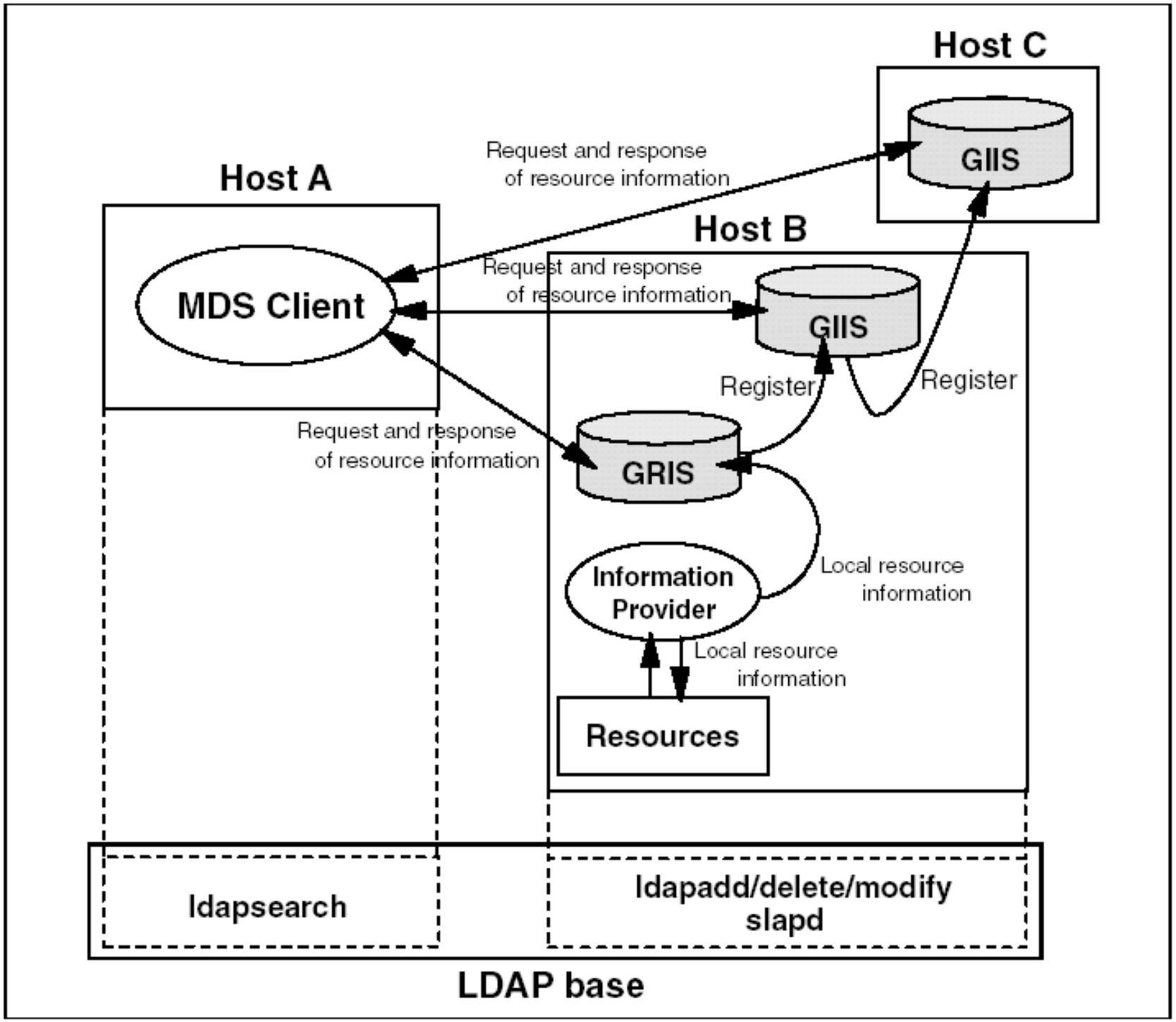
"Create Processes at A and B that Communicate & Access Files at C"



Grid Information Service

- Provide access to static and dynamic information regarding system components
- A basis for configuration and adaptation in heterogeneous, dynamic environments
- Resource Description Services
  - Supplies information about a specific resource
- Aggregate Directory Services
  - Supplies collection of information which was gathered from multiple resource description services
  - Customized naming and indexing

MDS Protocols and Services



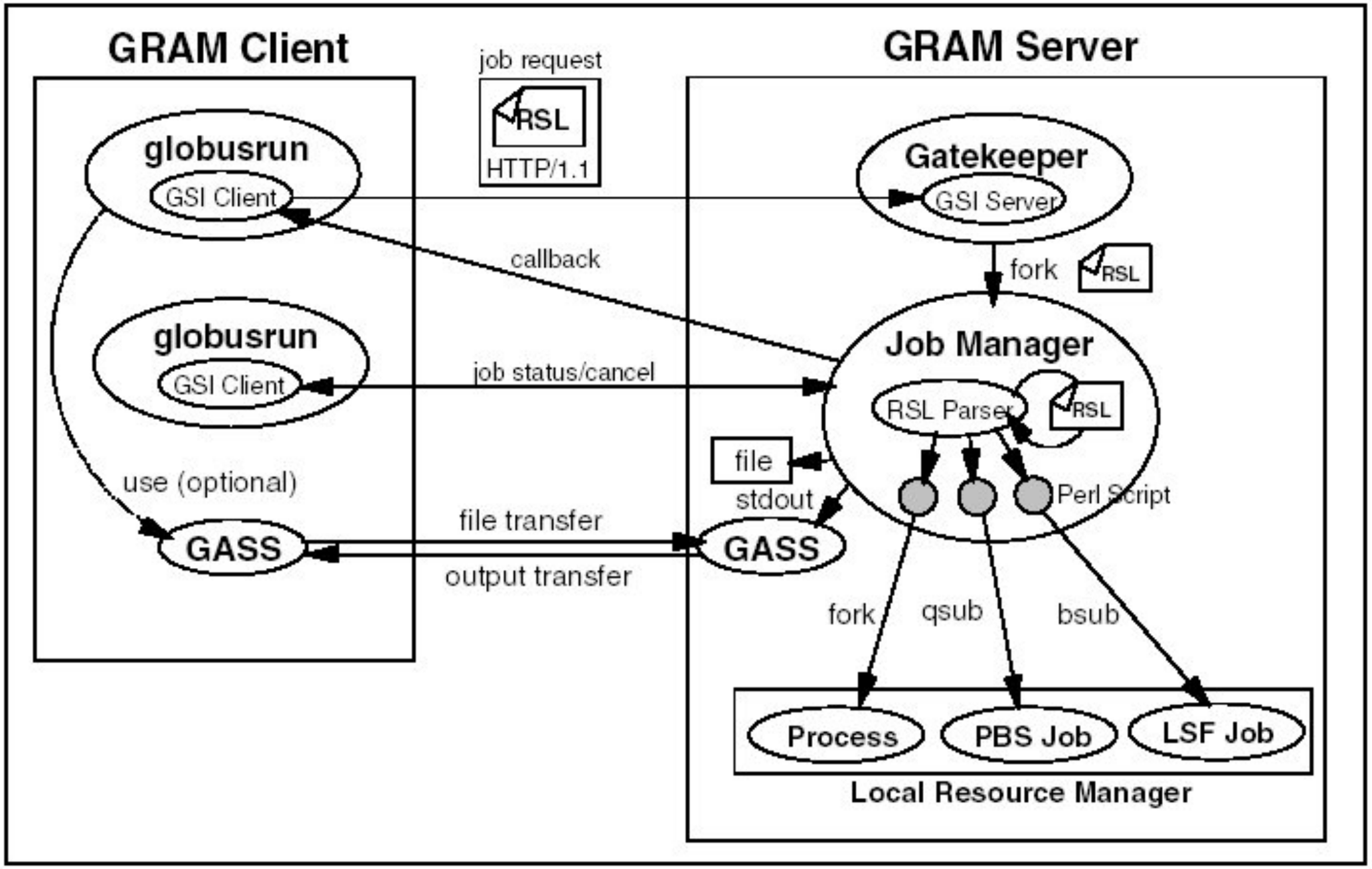
Grid Resource Management

- Grid Resource Management System consists of :
  - Local resource management system (Resource Layer)
    - Basic resource management unit
    - Provide a standard interface for using remote resources
    - Grid Resource Allocation Manager (GRAM)
  - Global resource management system (Collective Layer)
    - Coordinate all Local resource management system within multiple or distributed Virtual Organizations (VOs)
    - Provide high-level functionalities to efficiently use all of resources
      - Job Submission
      - Resource Discovery and Selection
      - Scheduling
      - Co-allocation
      - Job Monitoring, etc.
    - e.g. Meta-scheduler, Resource Broker, etc.

Definitions

- **Resource**: entity able to execute one or more jobs on the behalf of the user
- **Client**: process using GRAM protocol to submit a job request
- **Job**: one or more processes being part of a job request
- **Job request**: a message containing the request and the specification for a job execution on a remote resource. A typical job request specifies:
  - When and where processes should be created
  - How and what processes to create
  - How to execute and terminate processes
- **Gatekeeper**: remote resources service managing incoming job requests (GT2)
- **Job Manager**: service instantiated by the gatekeeper to manage the execution and monitor the job's processes (GT2)

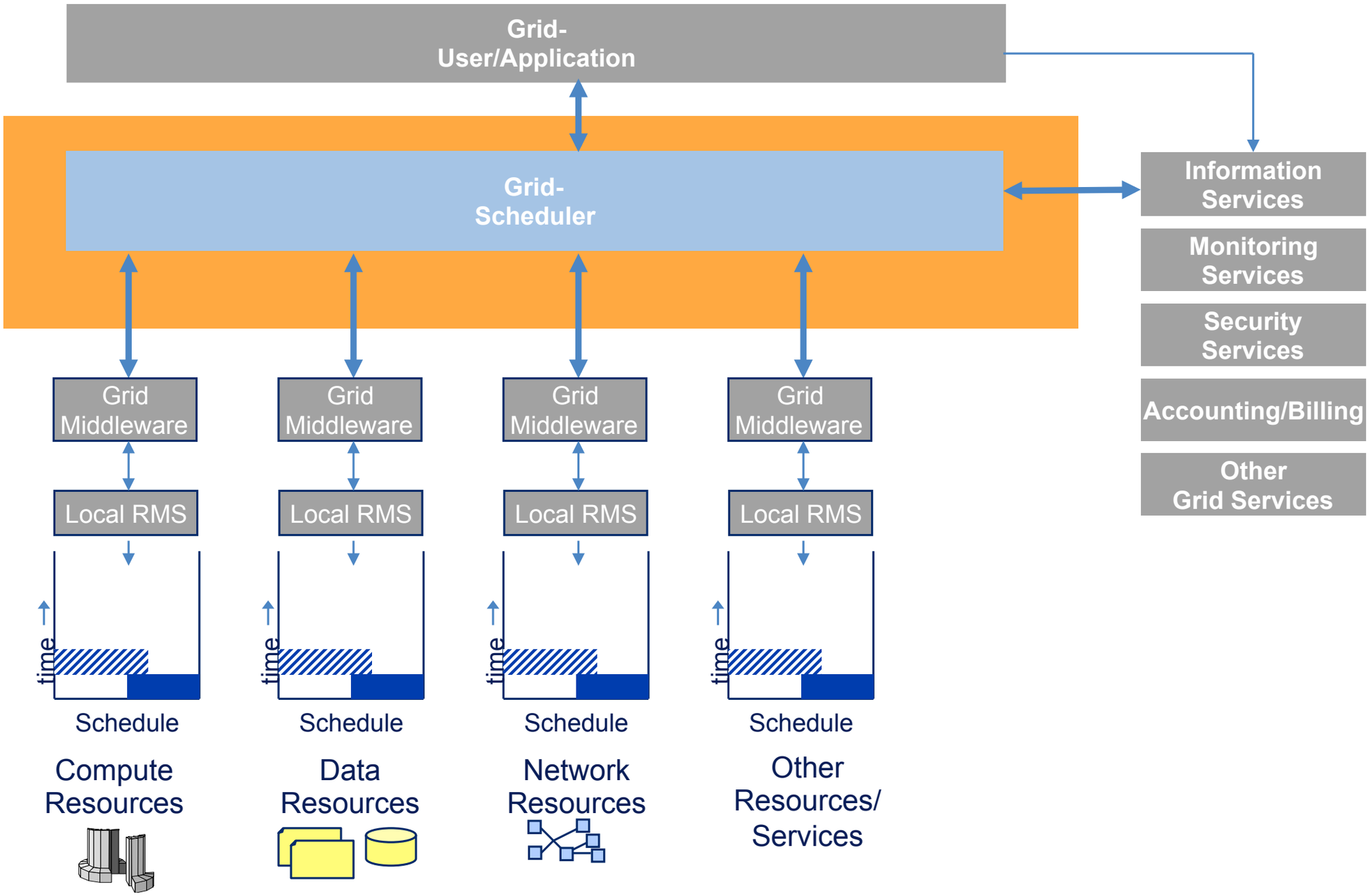
GRAM Protocols and Services



Grid Scheduling Levels

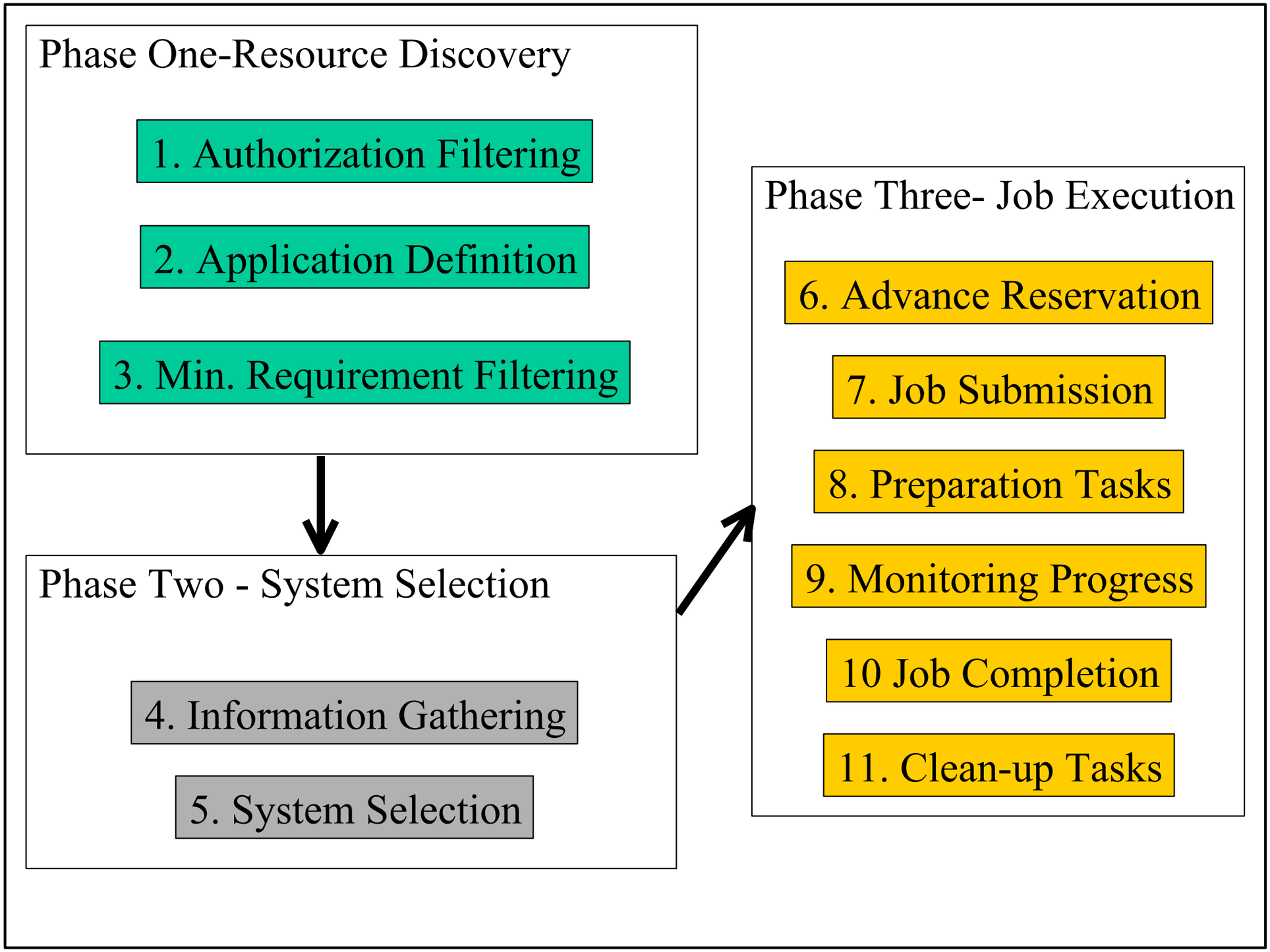
- Resource-level scheduler
  - low-level scheduler, local scheduler, local resource manager
  - scheduler close to the resource, controlling a supercomputer, cluster, or network of workstations, on the same local area network
  - Examples: Open PBS, PBS Pro, LSF, SGE
- Enterprise-level scheduler
  - Scheduling across multiple local schedulers belonging to the same organization
  - Examples: PBS Pro peer scheduling, LSF Multicluster
- Grid-level scheduler
  - Also known as super-scheduler, broker, community scheduler
  - Discovers resources that can meet a job's requirements
  - Schedules across lower level schedulers

Grid Scheduler



A Grid scheduler allows the user to specify the required resources and environment of the job without having to indicate the exact location of the resources

Grid Scheduler Activities



Issues

- Resources may dynamically join and leave the Grid
- Not all currently unused resources are available to grid jobs
- Resource owner local policies can restrict maximum number of grid jobs allowed
- Hard to predict how long jobs will wait in a queue
- User information not accurate as mentioned before
- New jobs arrive that may surpass current queue entries due to higher priority
- Local jobs have typically higher priority than Grid jobs
- Limited information about the local schedulers is available (privacy)
- Data Management
- Network Management

Example of Grid Scheduling Process

Scheduling Service:

1. receives job description
2. queries Information Service for static resource information
3. prioritizes and pre-selects resources
4. queries for dynamic information about resource availability
5. queries Data and Network Management Services
6. generates schedule for job
7. reserves allocation if possible
   otherwise selects another allocation
8. delegates job monitoring to Job Supervisor

Job Supervisor/Network and Data Management:
service, monitor and initiate allocation

Example:

40 resources of requested type are found.

12 resources are selected.

8 resources are available.

Network and data dependencies are detected.

Utility function is evaluated.

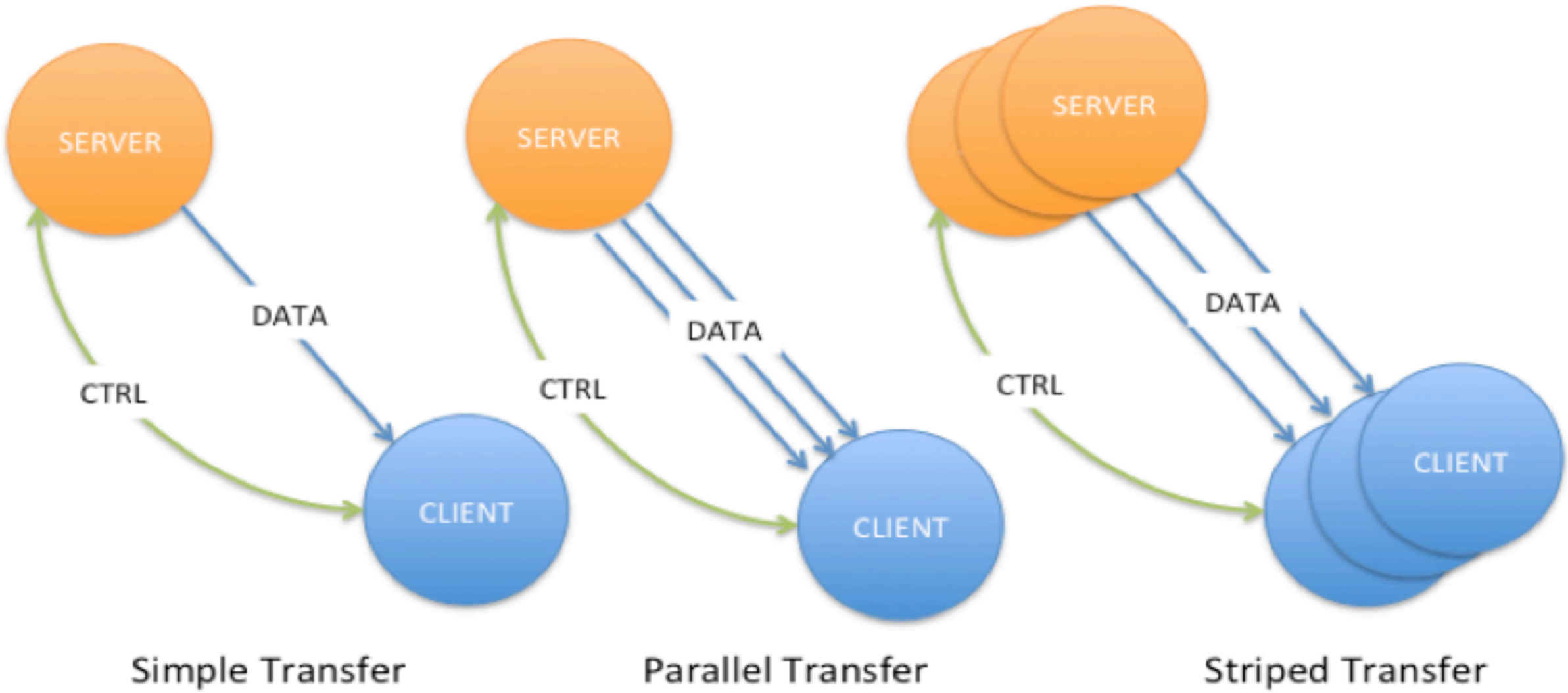
6th tried allocation is confirmed.

Data/network provided and job is started

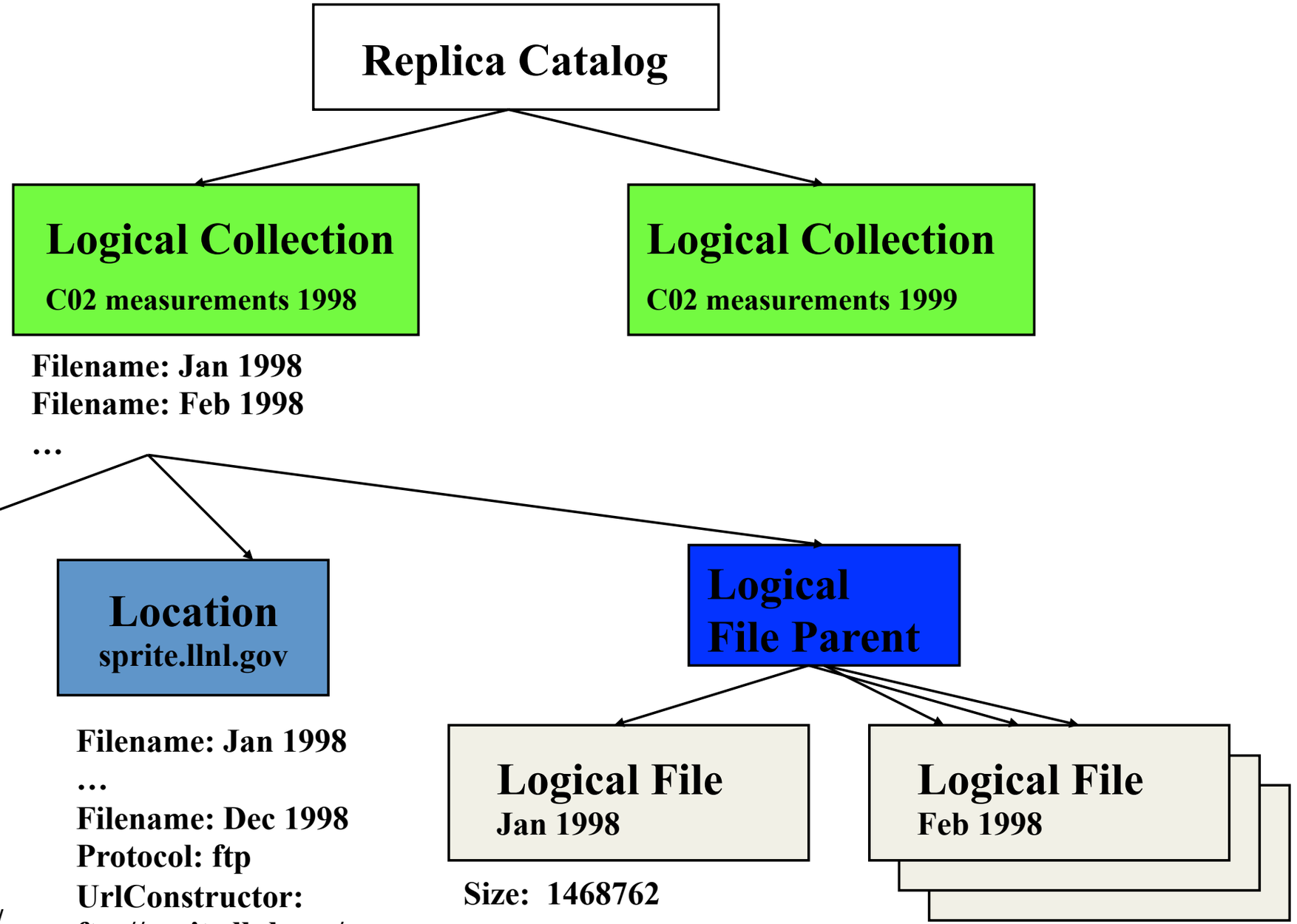
Grid Data Management

- Data access and transfer
  - **GASS**: Simple multi-protocol tool to transfer 'normal' files; integrated in GRAM
  - **GridFTP**: Reliable and high-performance file transfer protocol for 'big' files in computer networks
- Replica Management
  - **Replica Catalog**: Service to keep updated information on sets of replicated data
  - **Replica Management**: Service to create and manage sets of replicated data

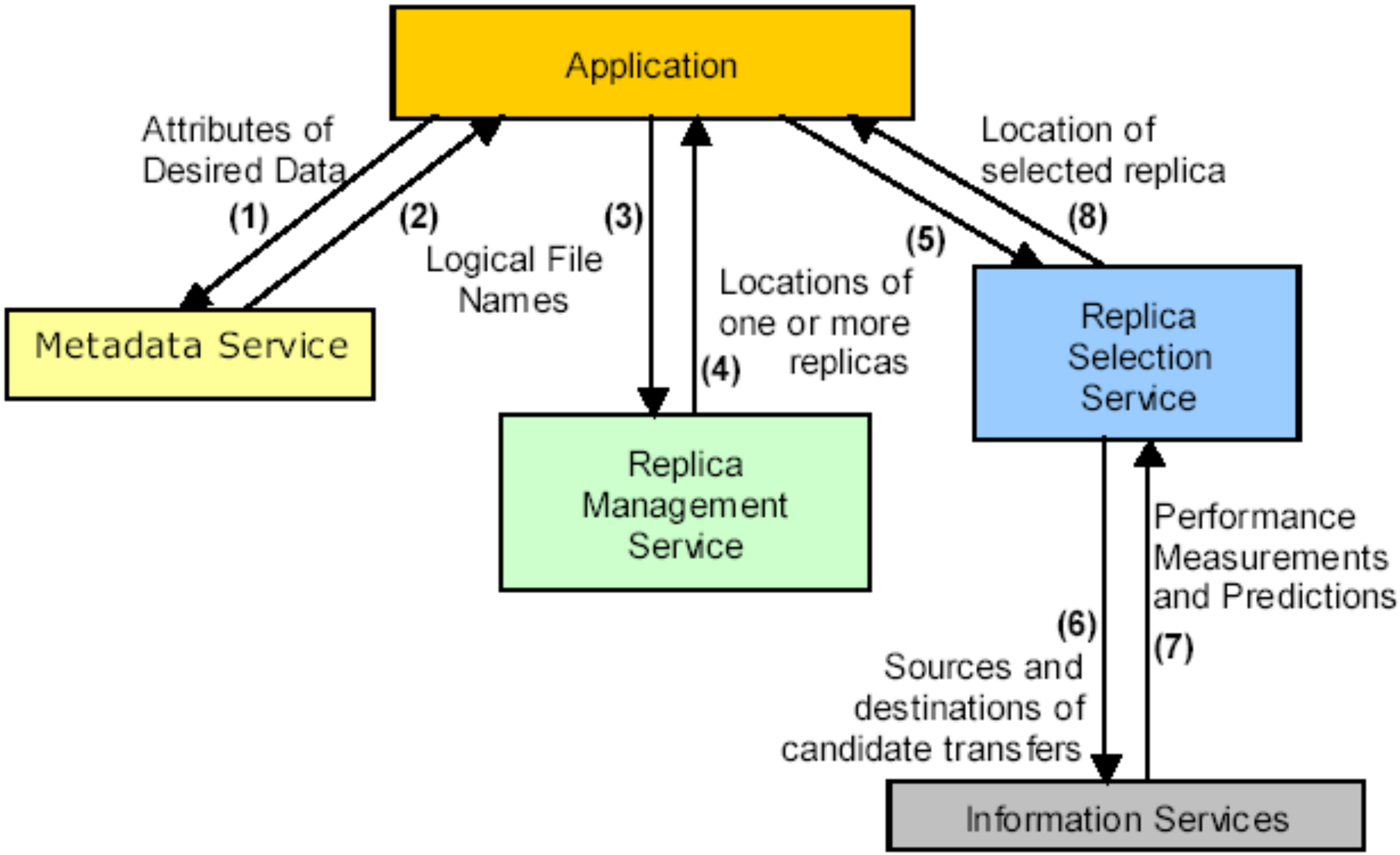
GridFTP


SERVER
DATA
CTRL
CLIENT
SERVER
DATA
CTRL
CLIENT
SERVER
DATA
CTRL
CLIENT
Simple Transfer
Parallel Transfer
Striped Transfer

Replica Catalog



Replica Management



Reading Assignments

- C. Kesselman, et al., The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International Journal of Supercomputing Applications, pp. 1-25, 2001.

  http://www.globus.org/alliance/publications/papers/anatomy.pdf

- IBM Redbooks paper, Fundamentals of Grid Computing

  http://www.redbooks.ibm.com/redpapers/pdfs/redp3613.pdf

- IBM Redbooks, Introduction to Grid Computing

  http://www.redbooks.ibm.com/redbooks/pdfs/sg246778.pdf

- Links and additional references provided at:

  http://www.cli.di.unipi.it/doku/doku.php/magistraleinformaticanetworking/cpa/start