Grid Computing

Adam Belloum
Institute of Informatics
University of Amsterdam
a.s.z.belloum@uva.nl

High Performance computing Curriculum, Jan 2015 http://www.hpc.uva.nl/

UvA-SURFsara

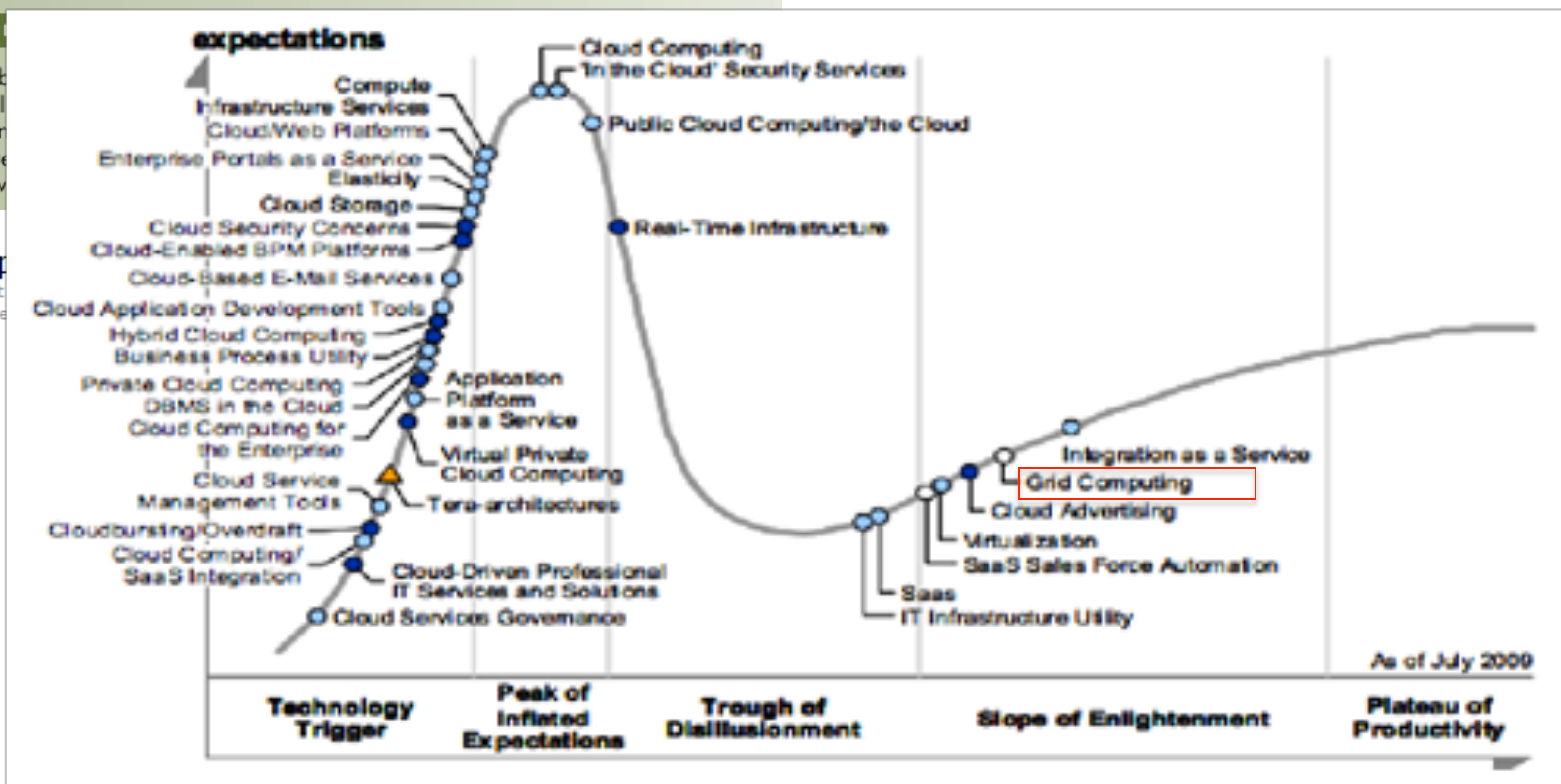
Grid Computing

RED HERRING

TOP TEN TRENDS 2001

DISTRIBUTE THE WEALTH

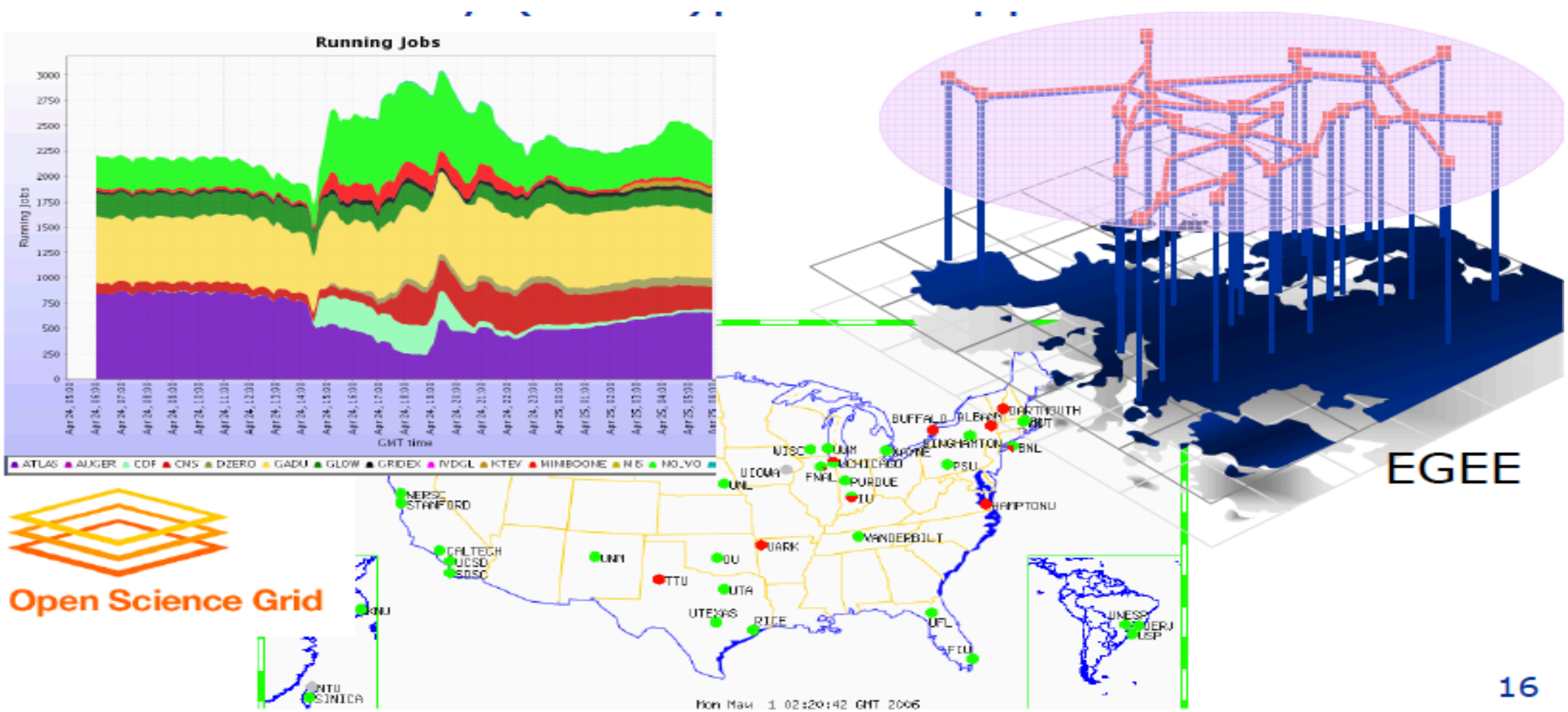
istributed computing initiatives.



Source: "Hype Cycle for Cloud Computing, 2009," Gartner

First Generation Grids: Batch Computing

Focus on aggregation of many resources for massively (data-)parallel applications



Second Generation Grids: Service-Oriented Science

- Empower many more users by enabling
  - on-demand access to services
- Grids become an enabling technology for service oriented science (or business)
  - Grid infrastructures host services
  - Grid technologies used to build services

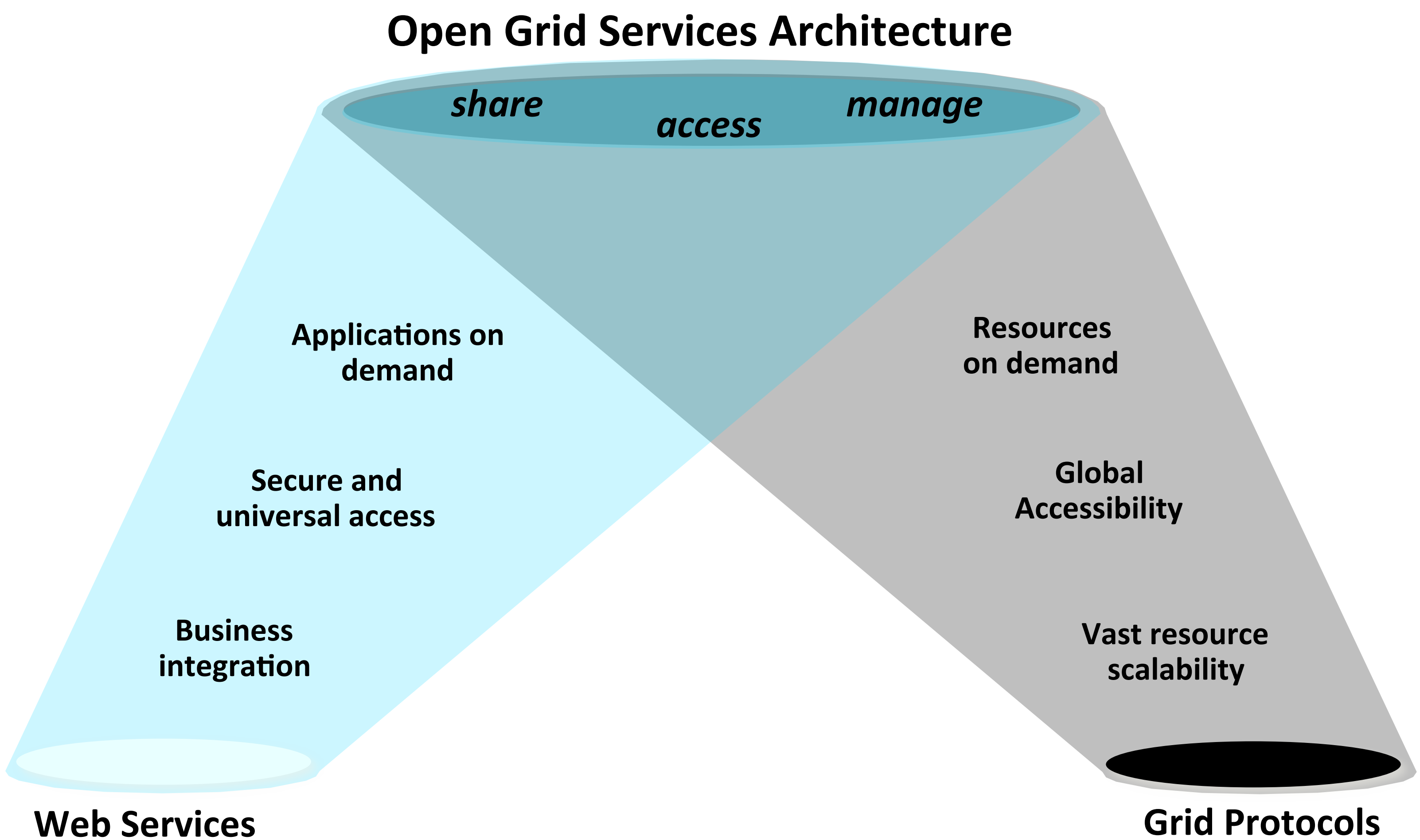
Science Gateways

TeraGrid™
EMPOWERING DISCOVERY

caBIG

"Service-Oriented Science", *Science*, 2005

Second Generation Grids: Service-Oriented Science (Best of Two Worlds)



'Open Grid Services Architecture Evolution, J.P. Prost, IBM Montpellier, France, Ecole Bruide 2004

Second Generation Grids: Service-Oriented Science (Transient Service Instances)

- "Web services" address discovery & invocation of persistent services
  - Interface to persistent state of entire enterprise
- In Grids, must also support transient service instances, created/destroyed dynamically
  - Significant implications for how services are managed, named, discovered, and used

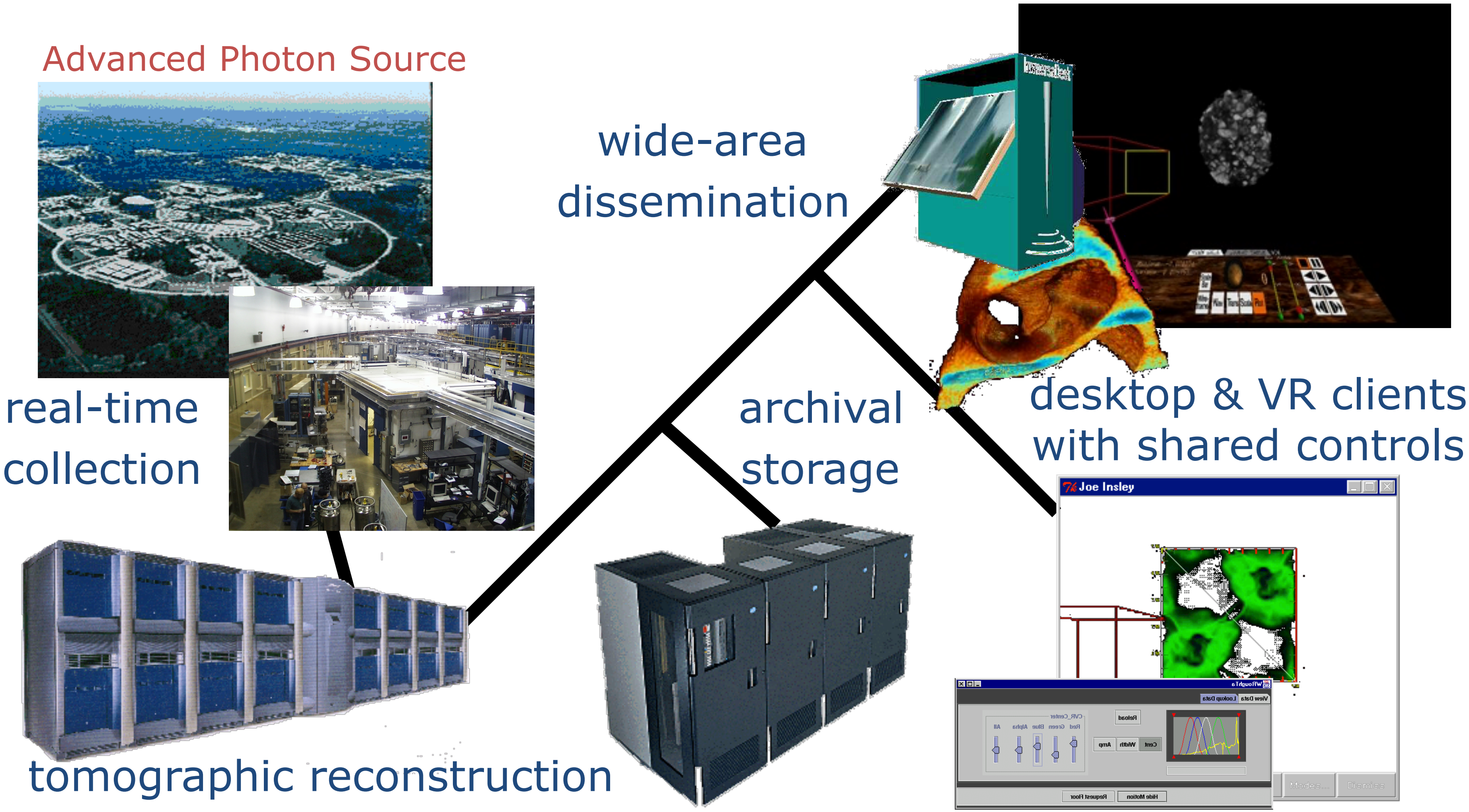
"Globus Toolkit Futures: An Open Grid Services Architecture" Ian Foster et al. Globus Tutorial, Argonne National Laboratory, January 29, 2002

eScience: Applications that needs the Grid

- "eScience promotes innovation in collaborative, computationally or data intensive research across all disciplines, throughout the research lifecycle"

- Nowadays Scientific Applications are
  - CPU intensive
  - Produce/process Huge sets of Data
  - Requires access to *geographically distributed* and *expensive* instruments

Online Access to Scientific Instruments



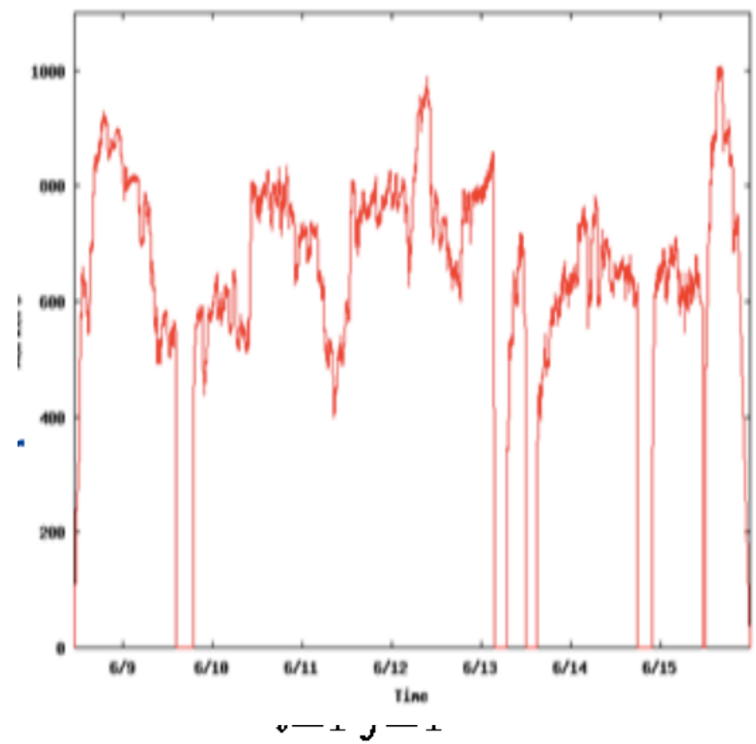
DOE X-ray grand challenge: ANL, USC/ISI, NIST, U.Chicago

From the Grid tutorials available at : http:.//www.globus.org

CPU intensive Science: Optimization problem NUG30

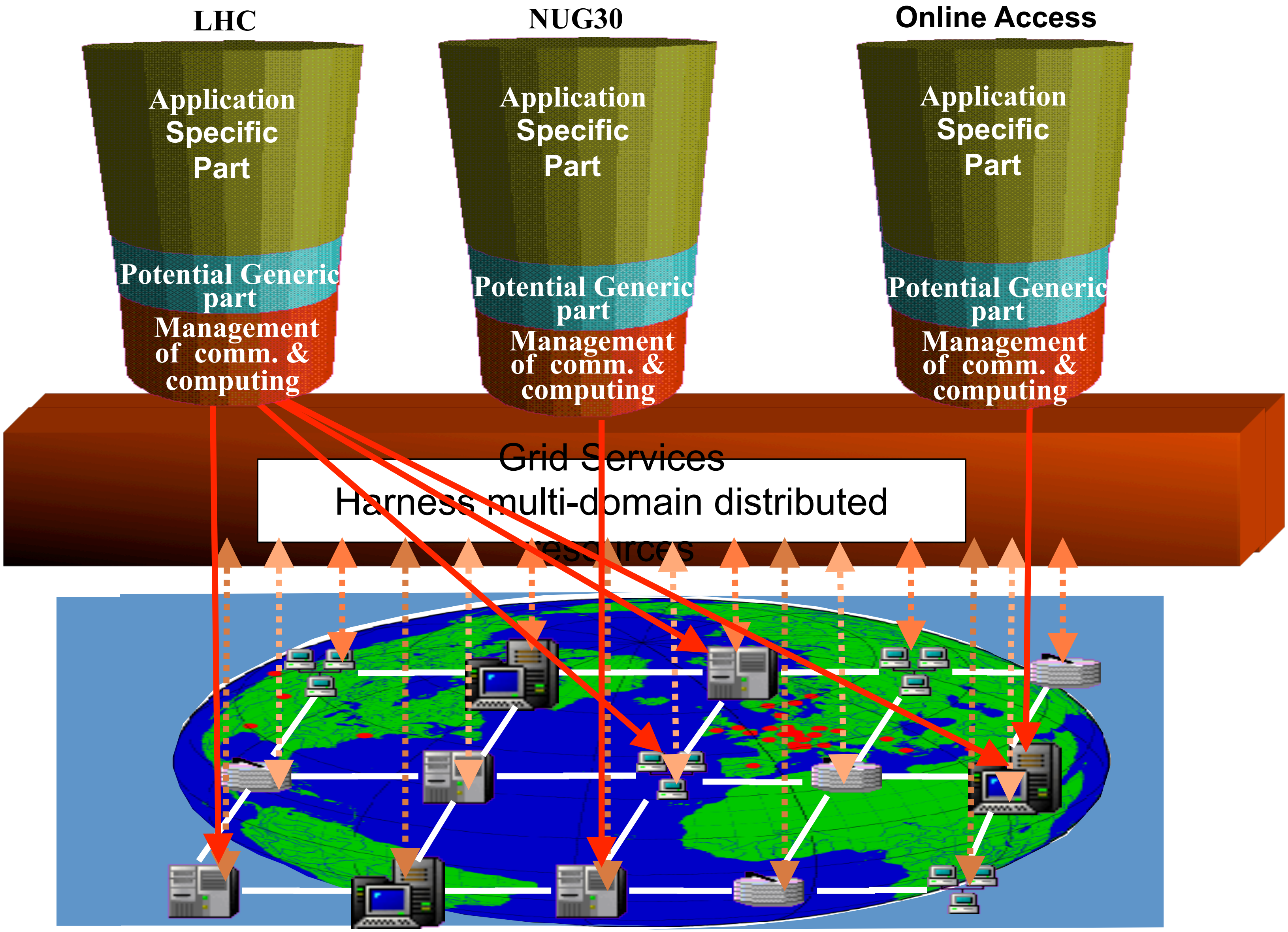
- a quadratic assignment problem (QAP) known as NUG30
  - given a set of n locations and n facilities, the goal is to assign each facility to a location.
  - There are **n!** possible assignments

- NUG30 proposed in 1968 as a test of computer capabilities, but remained unsolved because of its great complexity.

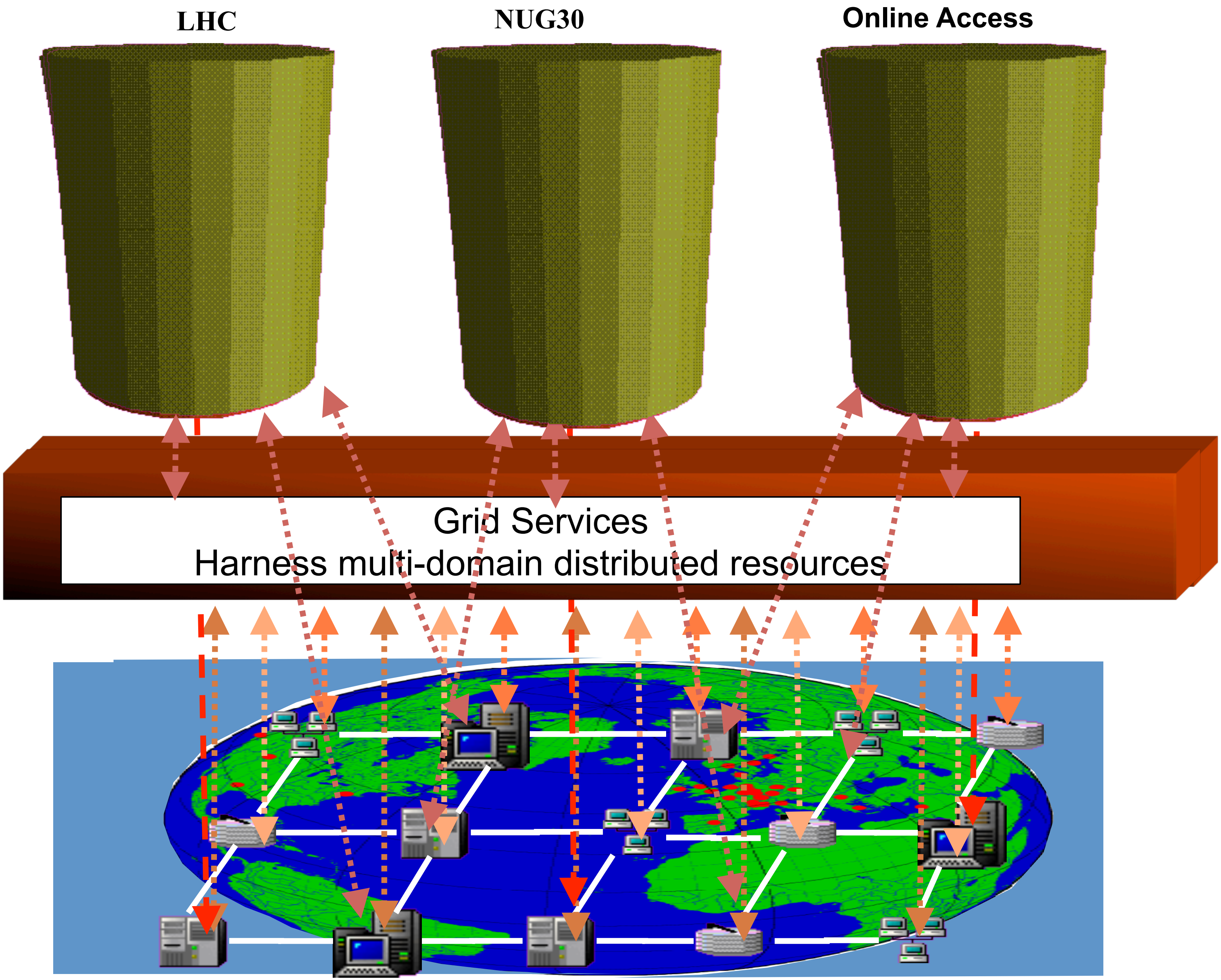


Nug30 Quadratic Assignment Problem Solved by 1,000 https://scout.wisc.edu/archives/r7125

To solve these problems?



“VL-e project” UvA


LHC
NUG30
Online Access
Grid Services
Harness multi-domain distributed resources

The Grid Vision

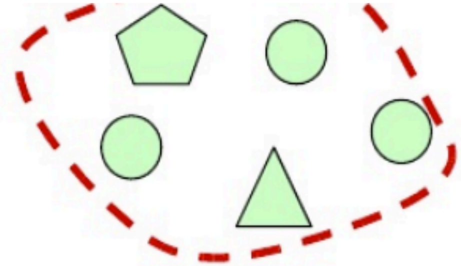
"Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations"

- On-demand, ubiquitous access to computing, data, and services
- New capabilities constructed dynamically and transparently from distributed services

"When the network is as fast as the computer's internal links, the machine disintegrates across the net into a set of special purpose appliances"

(George Gilder)

The Grid paradigm

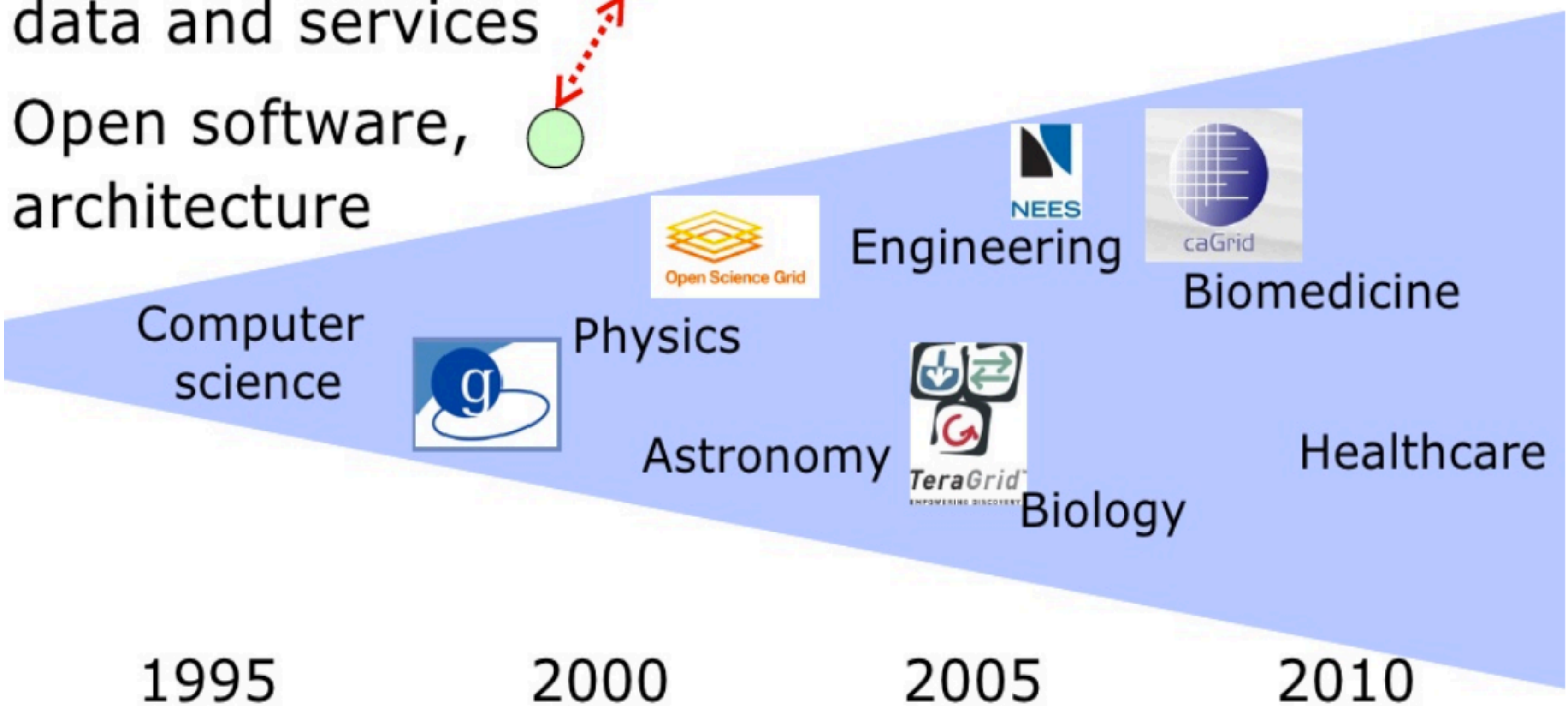


Principles and mechanisms for dynamic VOs

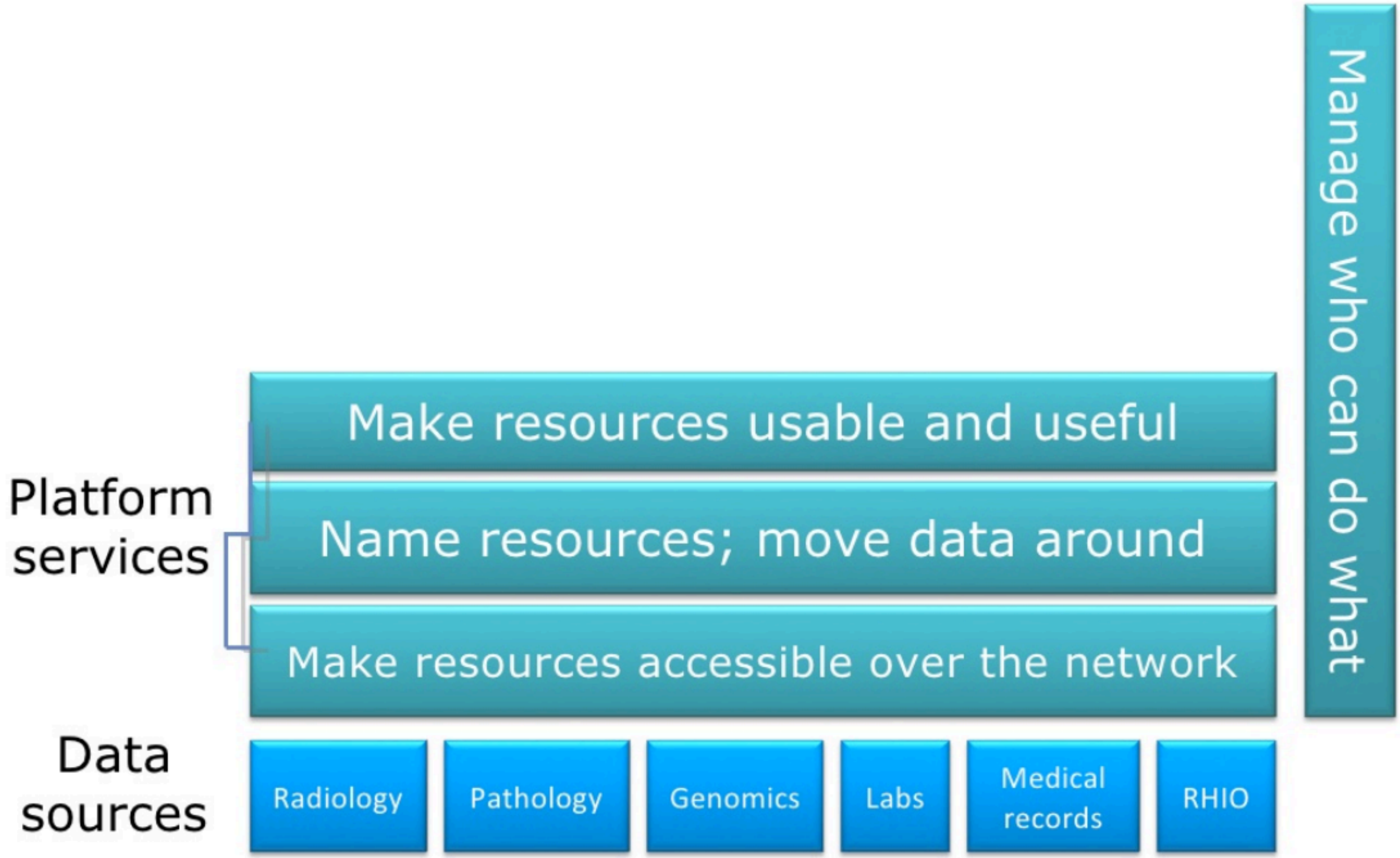
Leverage service oriented architecture (SOA)

Loose coupling of data and services

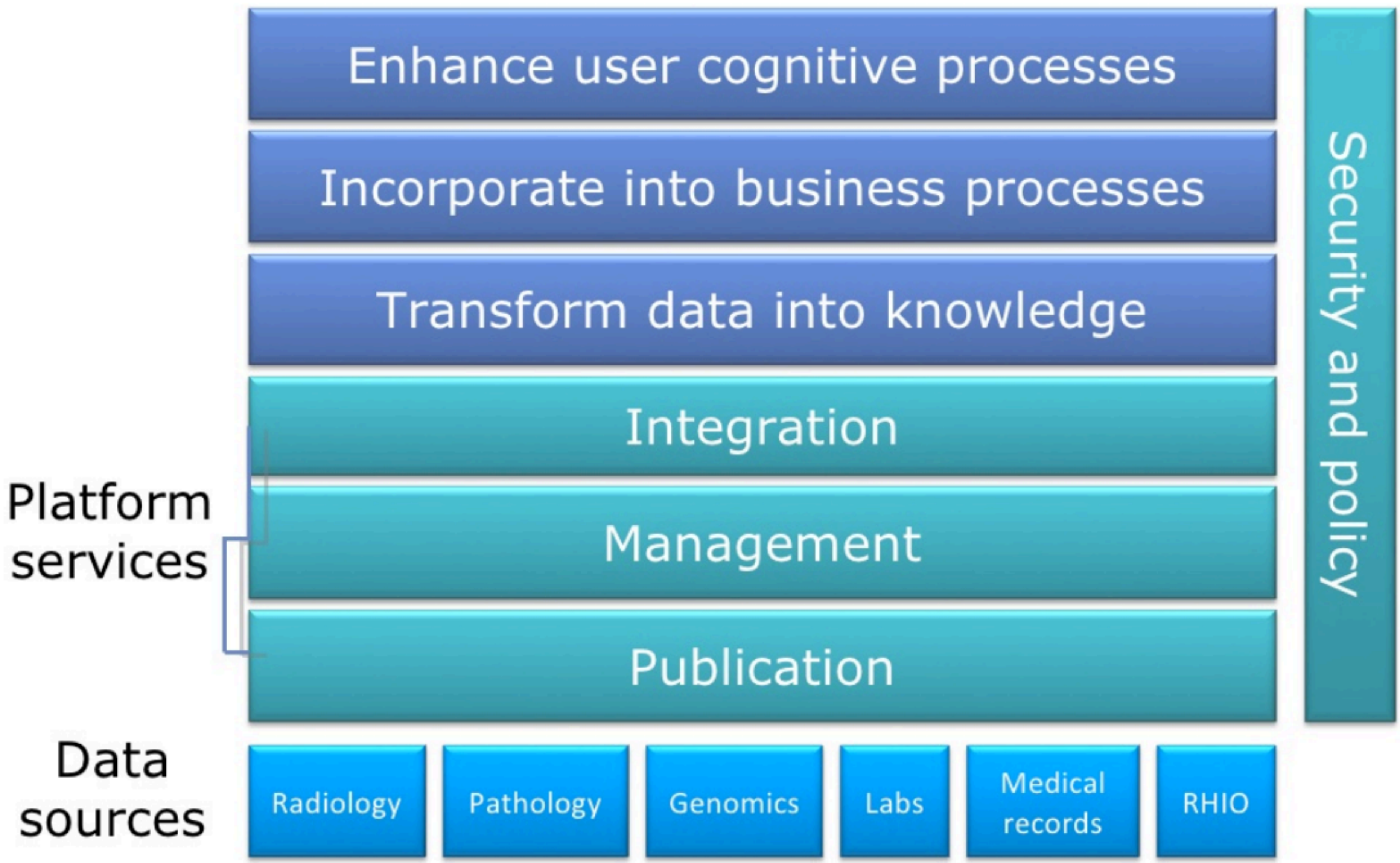
Open software, architecture



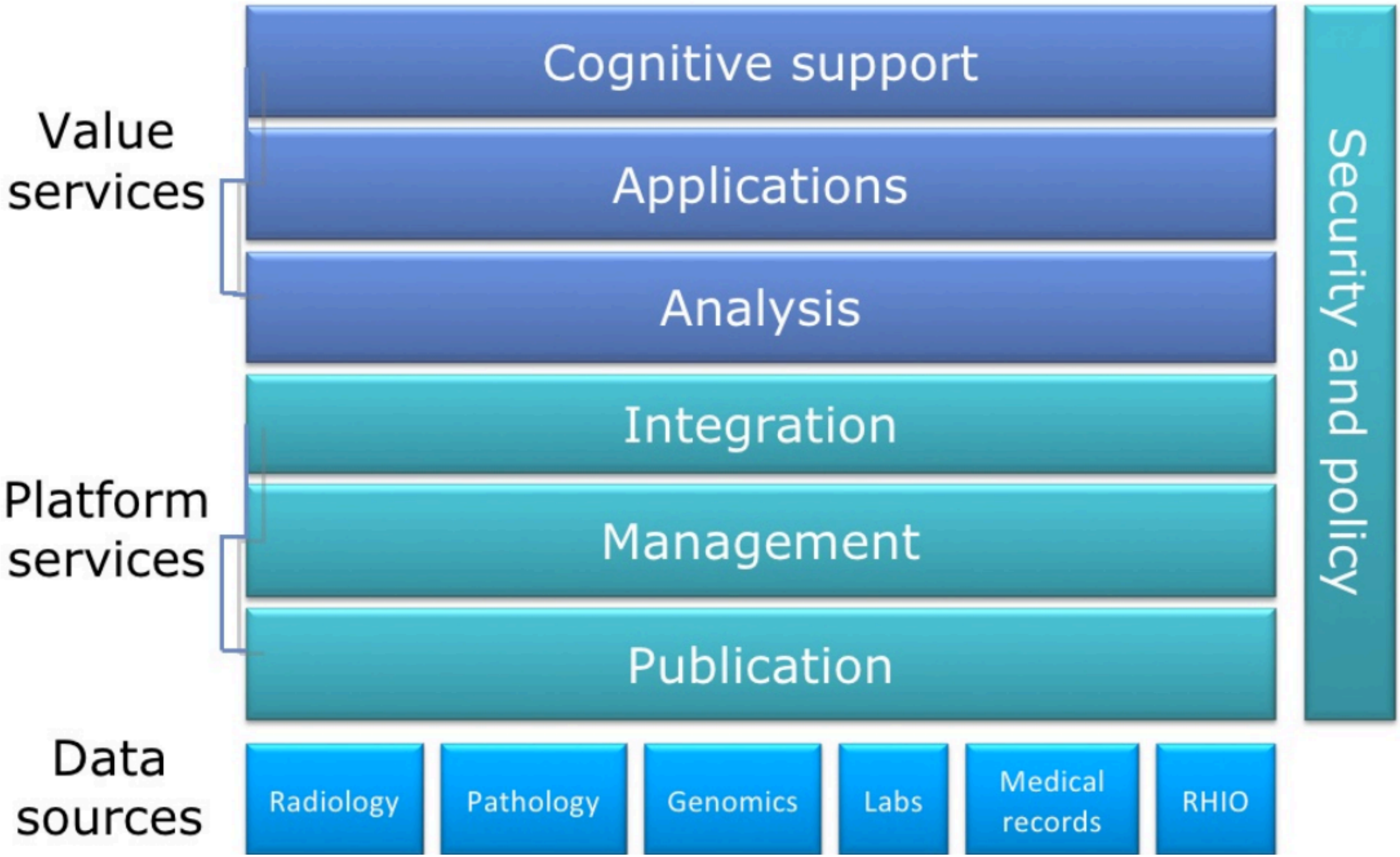
The Grid paradigm and information integration



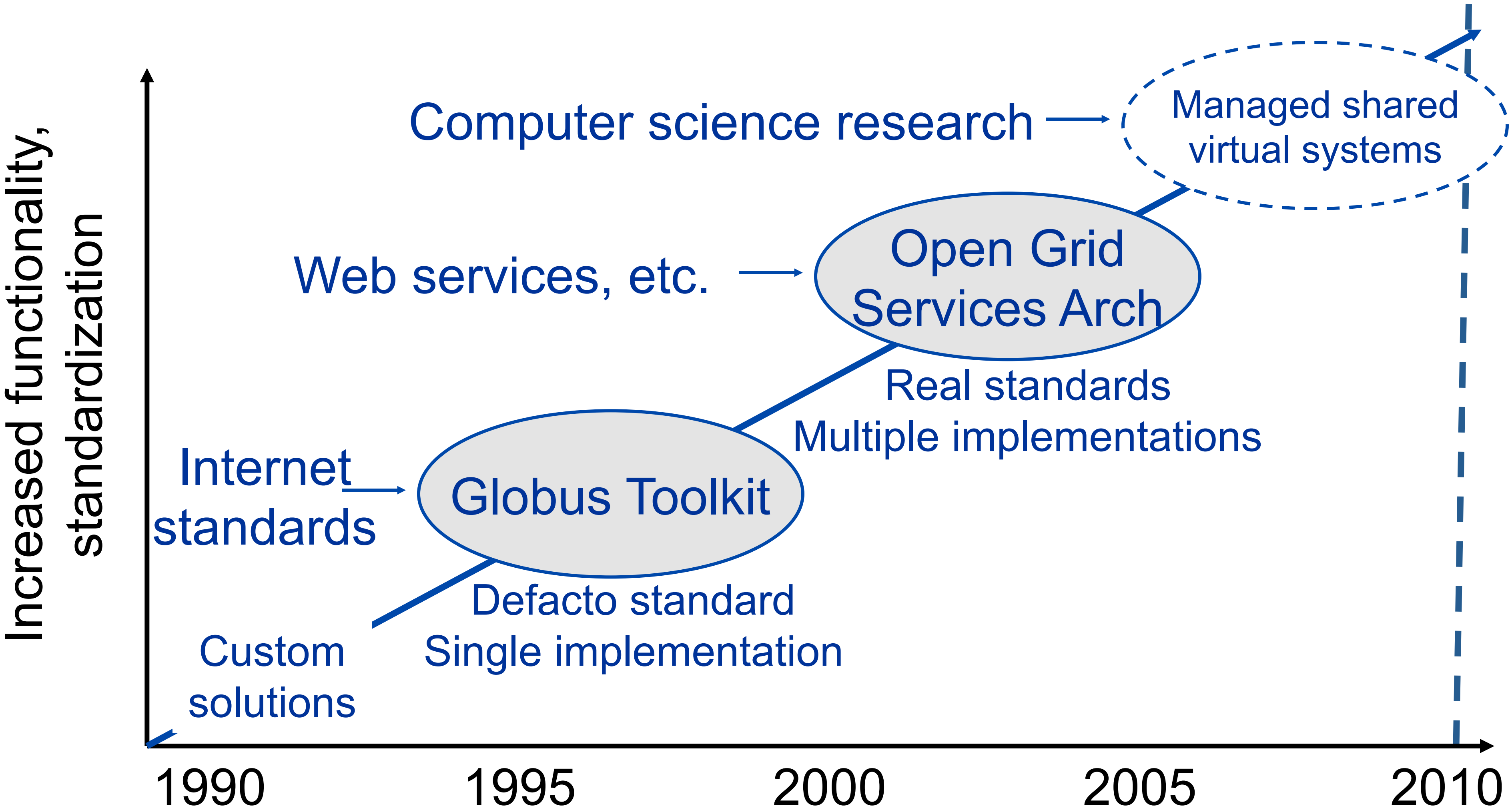
The Grid paradigm and information integration



The Grid paradigm and information integration



Emergence of Open Grid Standards



"Grid Computing and Scaling Up the Internet" I. Foster, IPv6 Forum, an

The Grid Middleware

- Software toolkit addressing key technical areas
  - Offer a modular "bag of technologies"
  - Enable incremental development of grid-enabled tools and applications
  - Define and standardize grid protocols and APIs

- Focus is on inter-domain issues, not clustering
  - Collaborative resource use spanning multiple organizations
  - Integrates cleanly with intra-domain services
  - Creates a "collective" service layer

"Basics Globus Toolkit™ Developer Tutorial" Globus Team, 2003

Grid Middleware Definition

- Architecture identifies the fundamental system components, specifies purpose and function of these components, and indicates how these components interact with each other.
- Grid architecture is a protocol architecture, with protocols defining the basic mechanisms by which VO users and resources negotiate, establish, manage and exploit sharing relationships.
- Grid architecture is also a service standard-based open architecture that facilitates extensibility, interoperability, portability and code sharing.

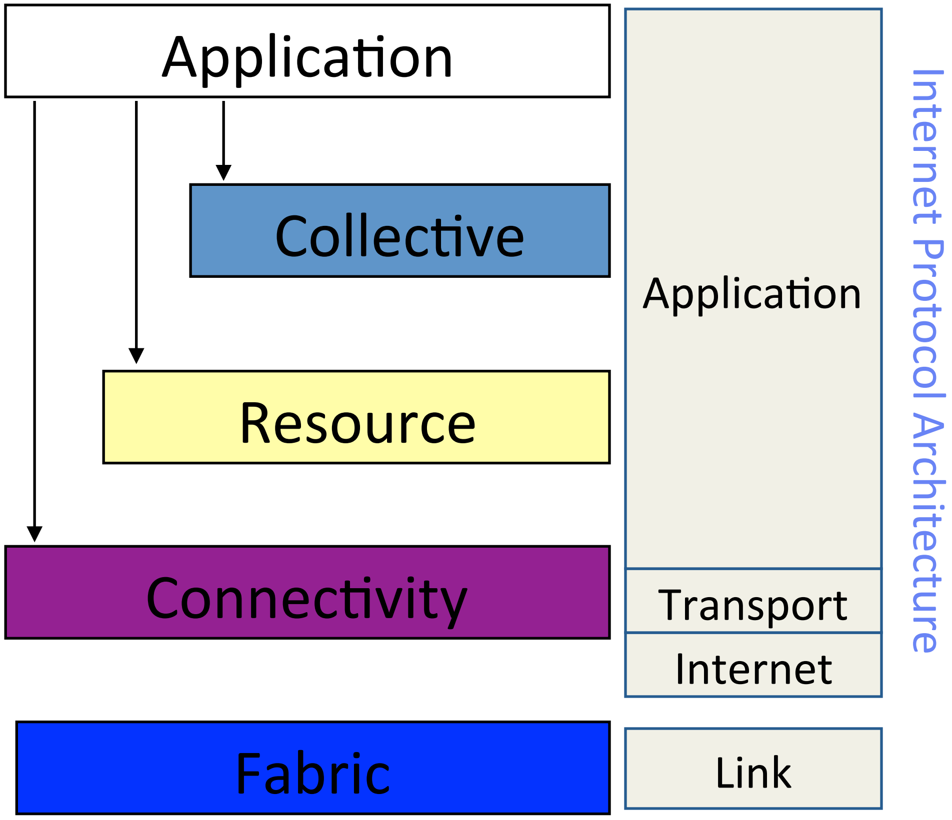
Architecture

"Coordinating multiple resources": ubiquitous infrastructure services, app-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



Examples of Grid Middleware

- Globus Toolkit (GT4.X) now (GT5.X)
  - www.globus.org
- Legion/Avaki
  - http://www.avaki.com/
  - http://legion.virginia.edu/
- Grid Sun engine
  - http://www.sun.com/service/sungrid/overview.jsp
- Unicore
  - http://www.unicore.org

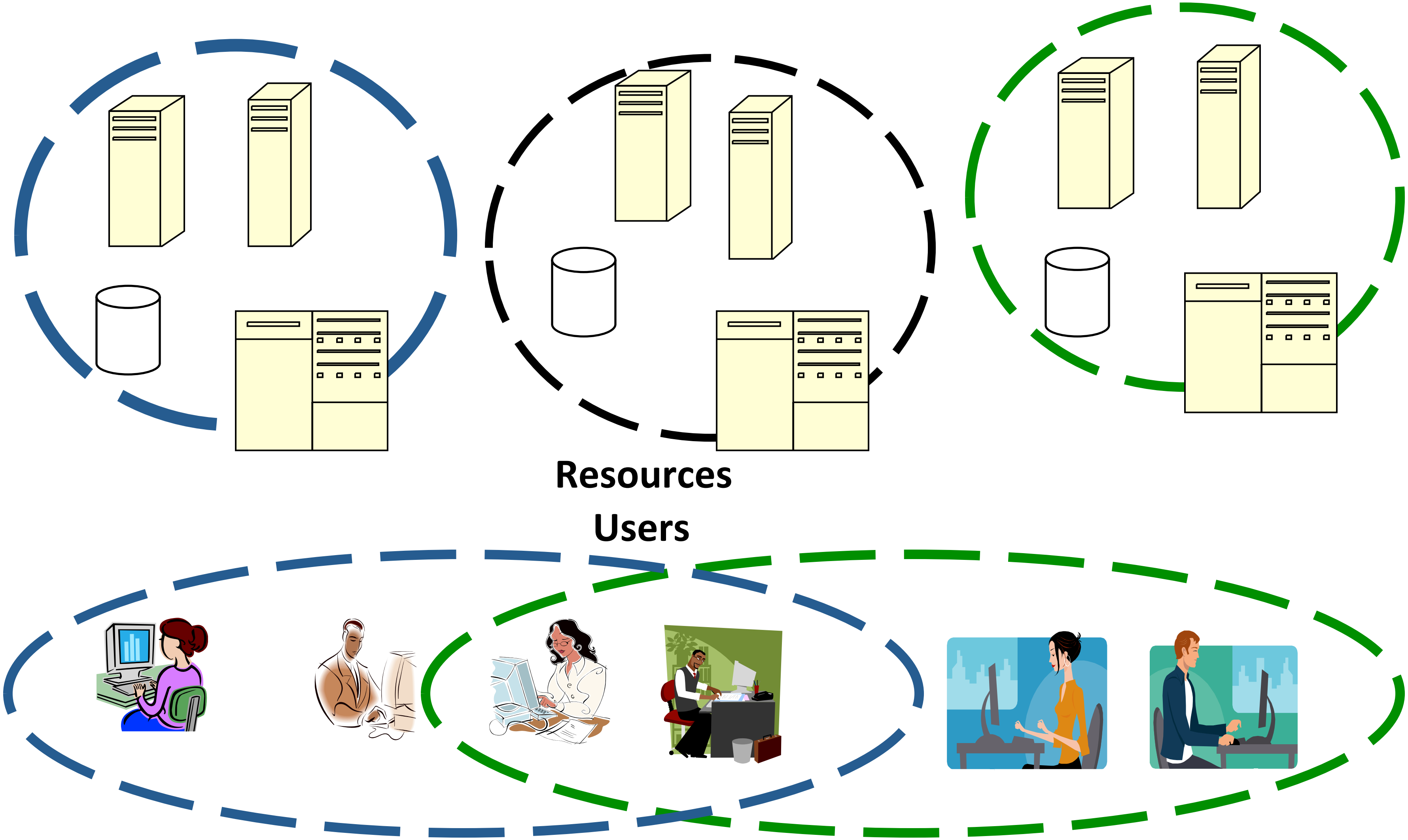
The Grid Approach and Problem

- Flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources

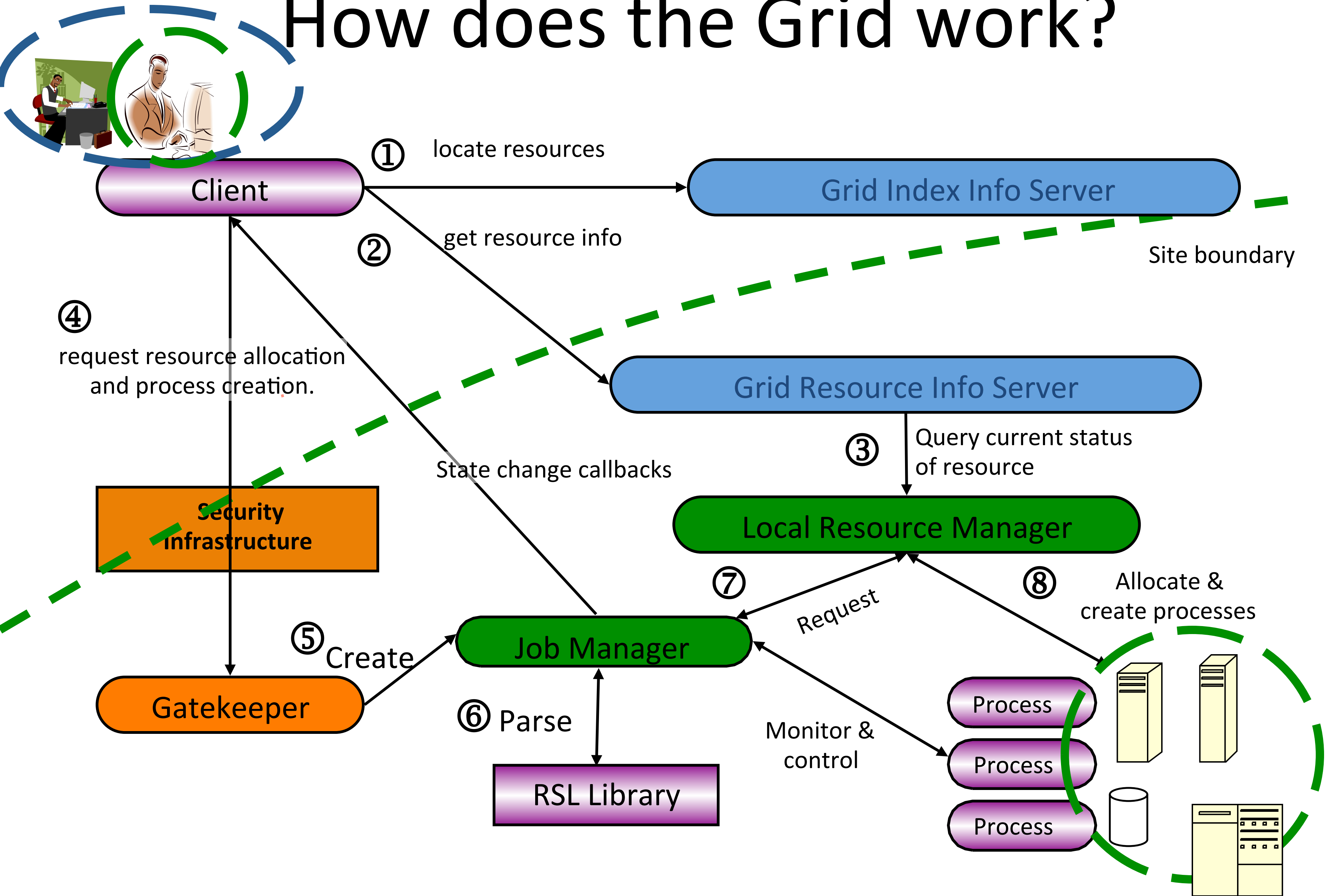
- Enable communities ("Virtual Organizations") to share geographically distributed resources as they pursue common goals -- *assuming the absence* of : central location, central control, existing trust relationships.

From the Grid tutorials available at : http:.//www.globus.org

Typical Grid Scenario



How does the Grid work?



The Four components of a Grid infrastructure

- Resource Management
- Information services
- Data Management
- Security

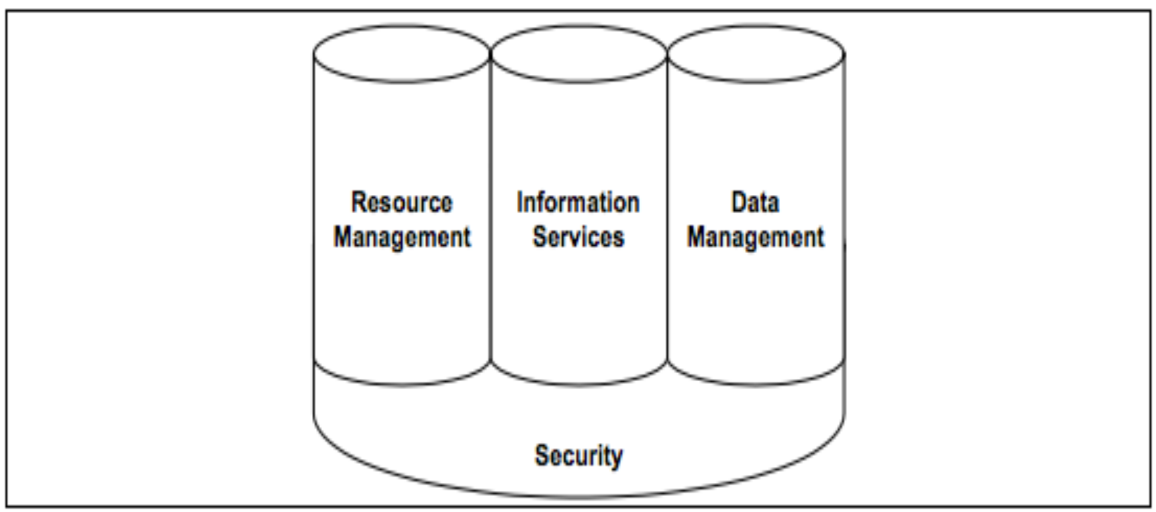
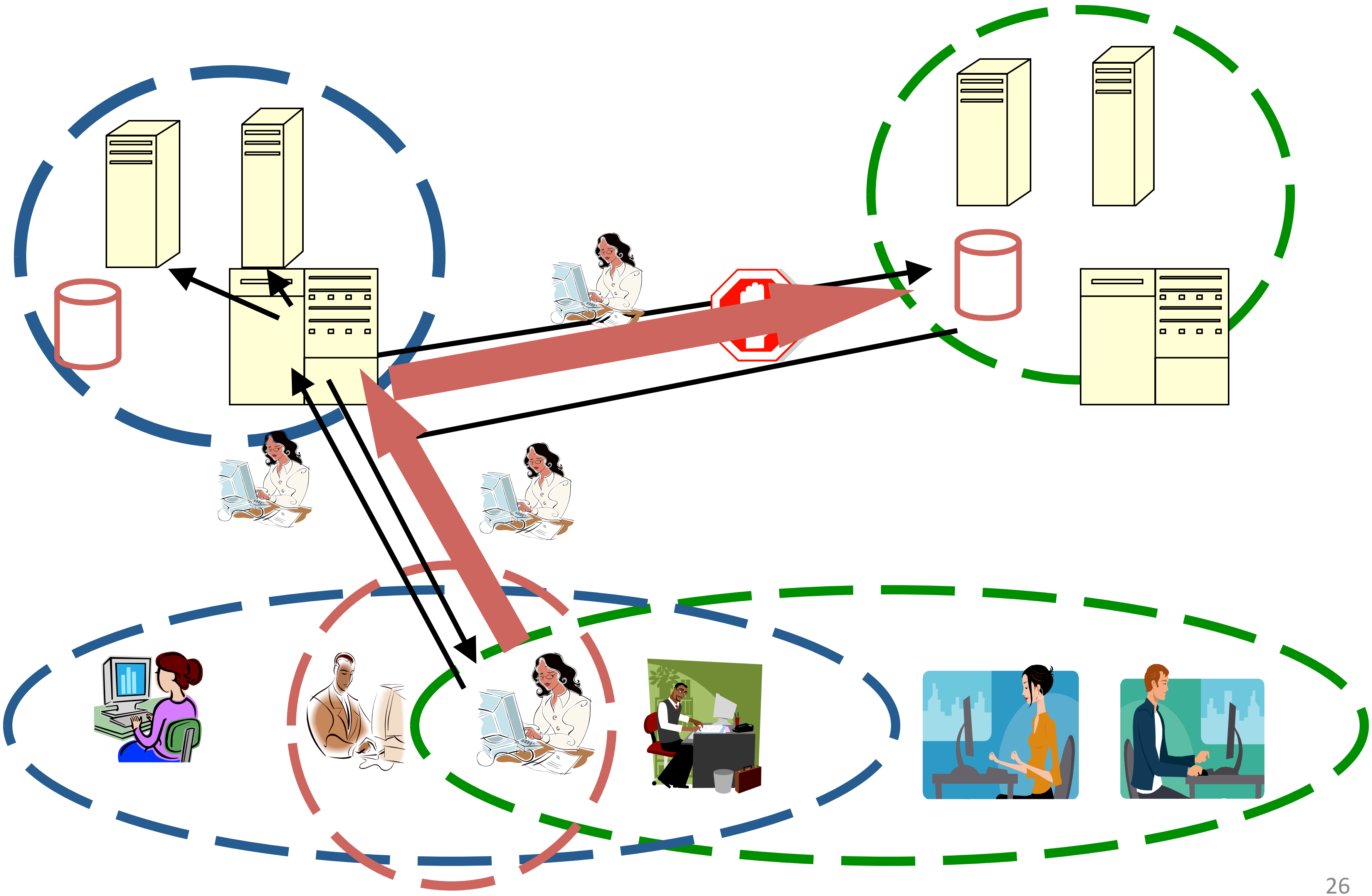
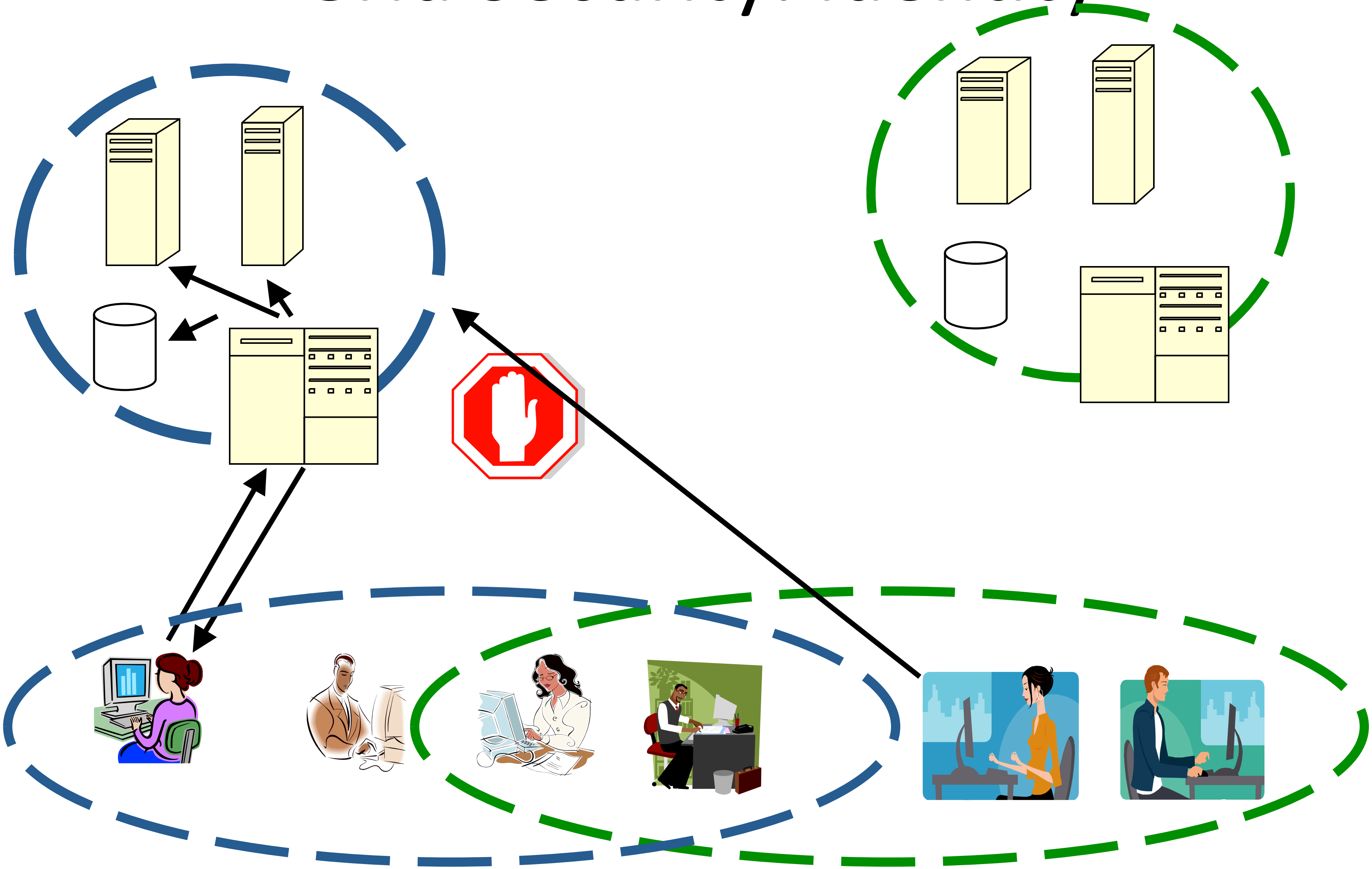


Figure 1-1 Grid computing key areas

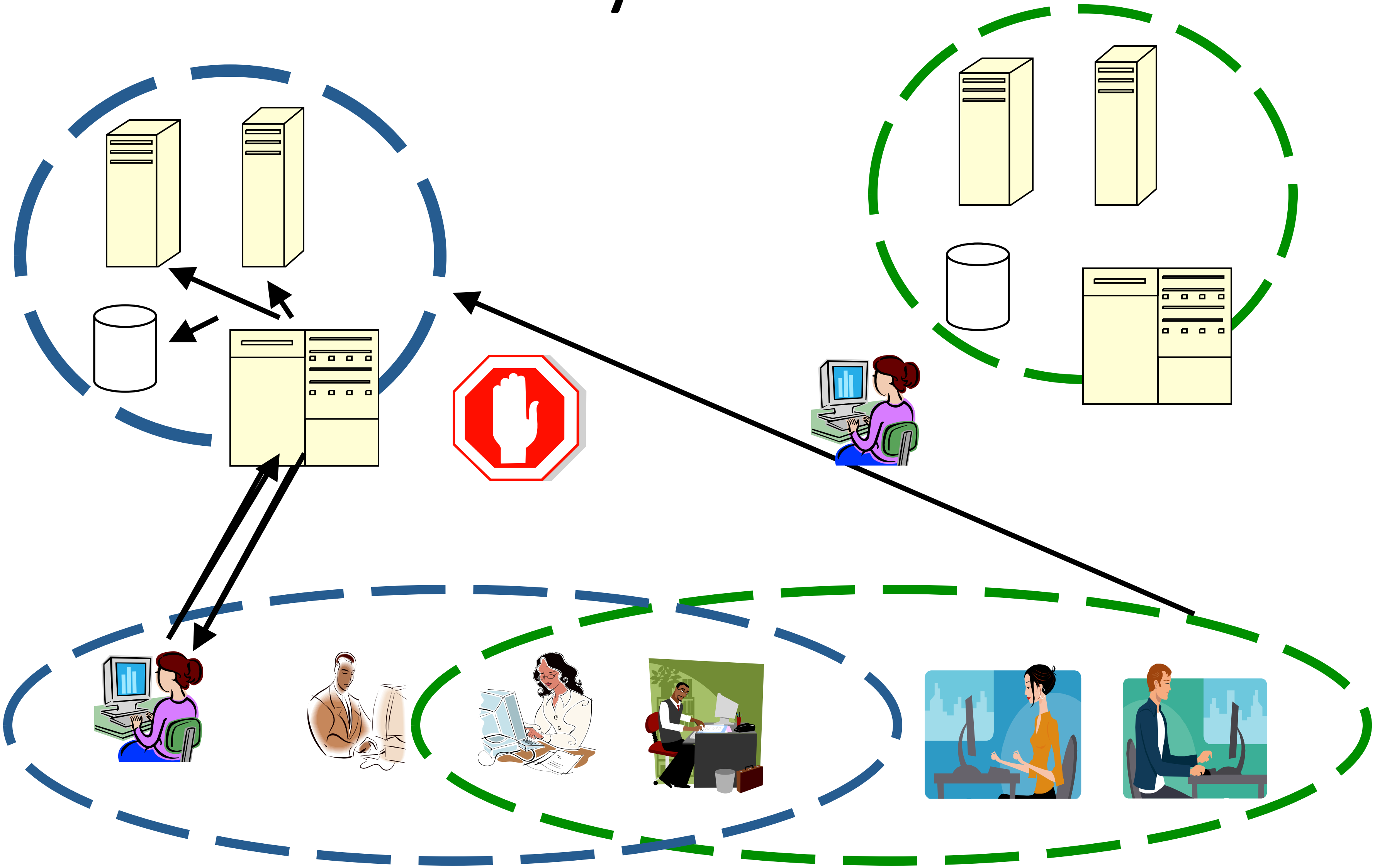
Grid Security: Single Sign On, Delegation



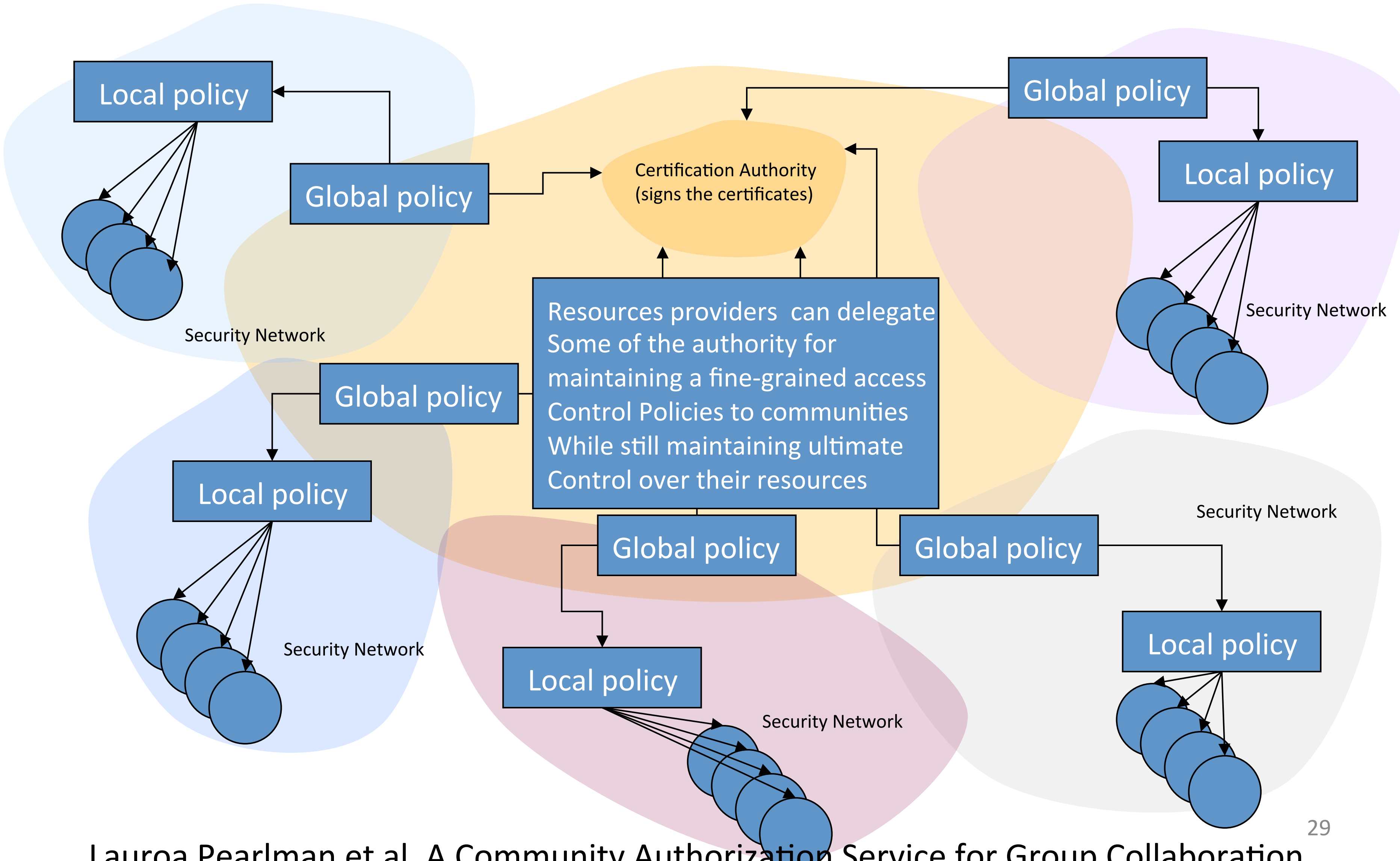
Grid Security: Identity



Grid Security: Authentication



Security cross Grid (V.O.)

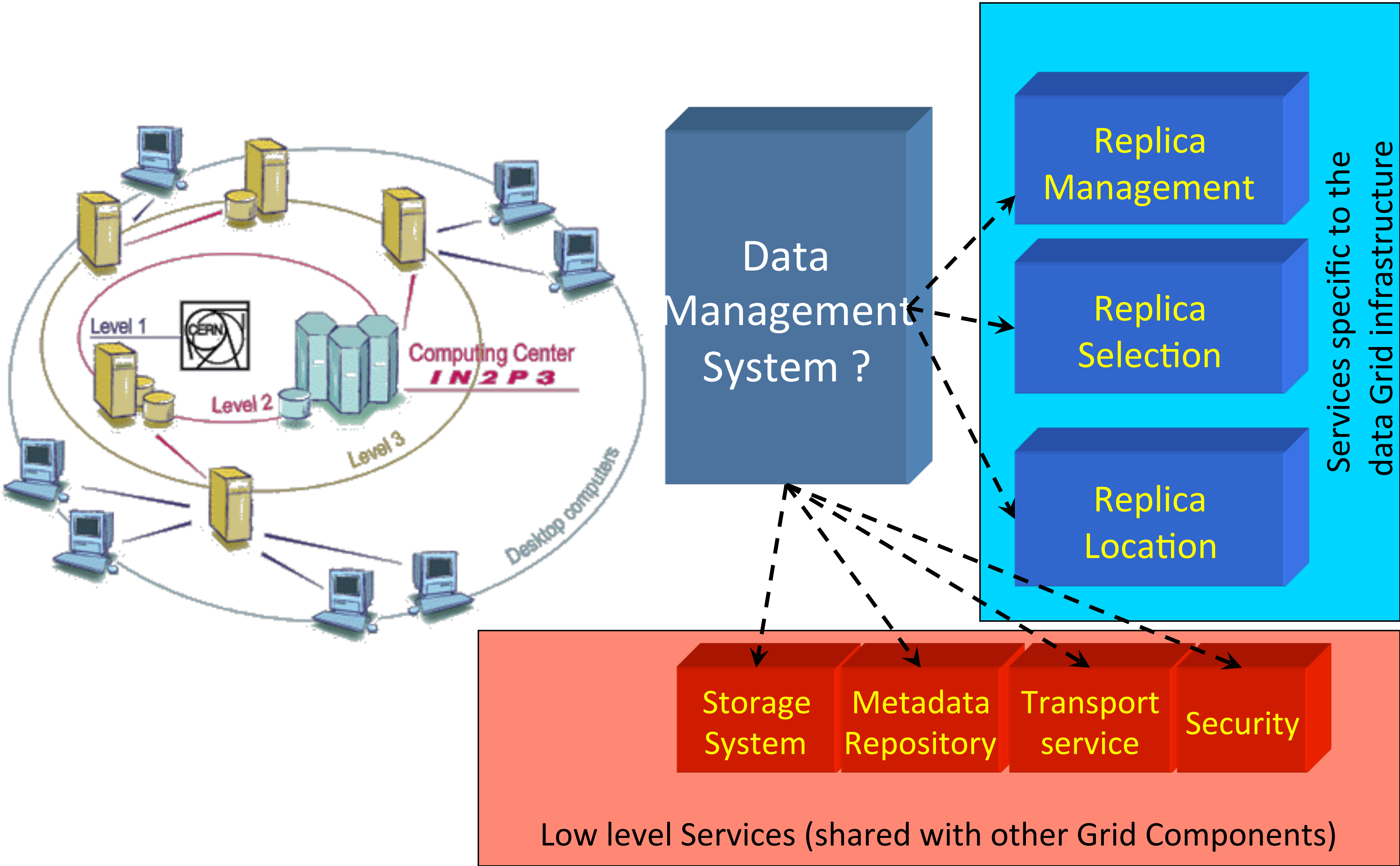


Lauroa Pearlman et al. A Community Authorization Service for Group Collaboration

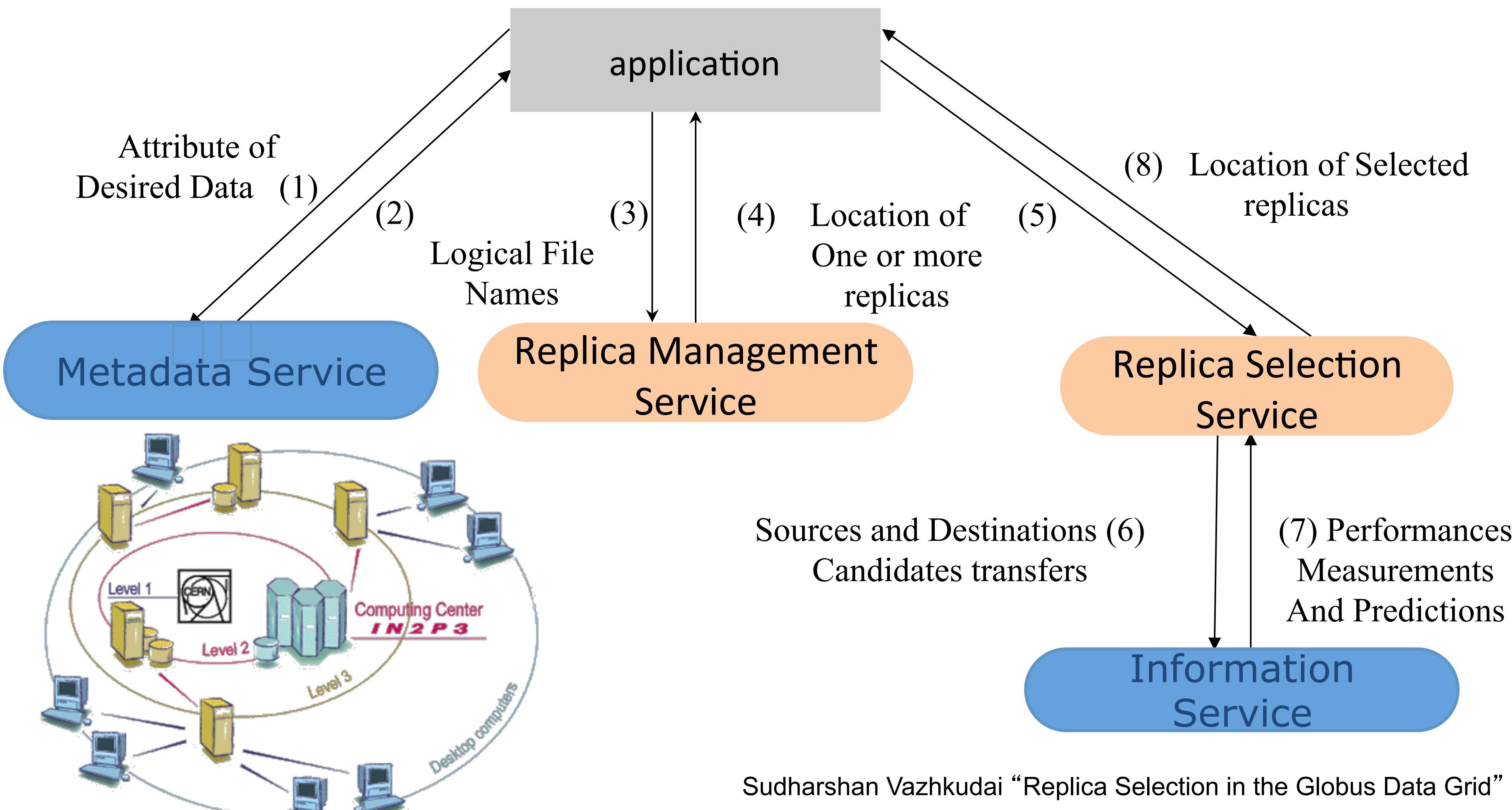
outline

- e-Science
- Grid approach
- Grid computing
- Programming models for the Grid
- Grid-middleware
- Web Services
- Open Grid Service Architecture (OGSA)

Grid data management

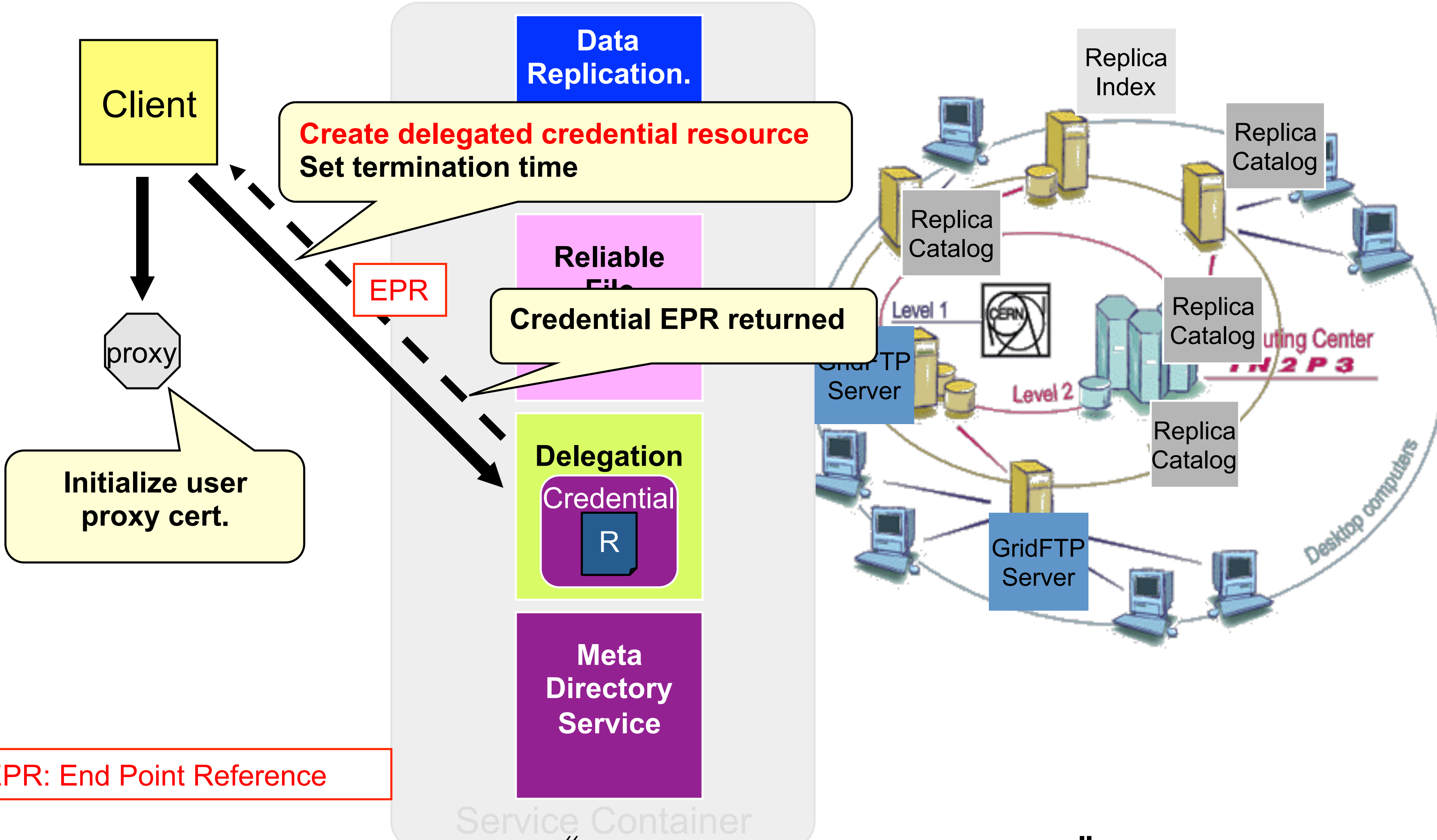


A Data selection scenario



Sudharshan Vazhkudai “Replica Selection in the Globus Data Grid”

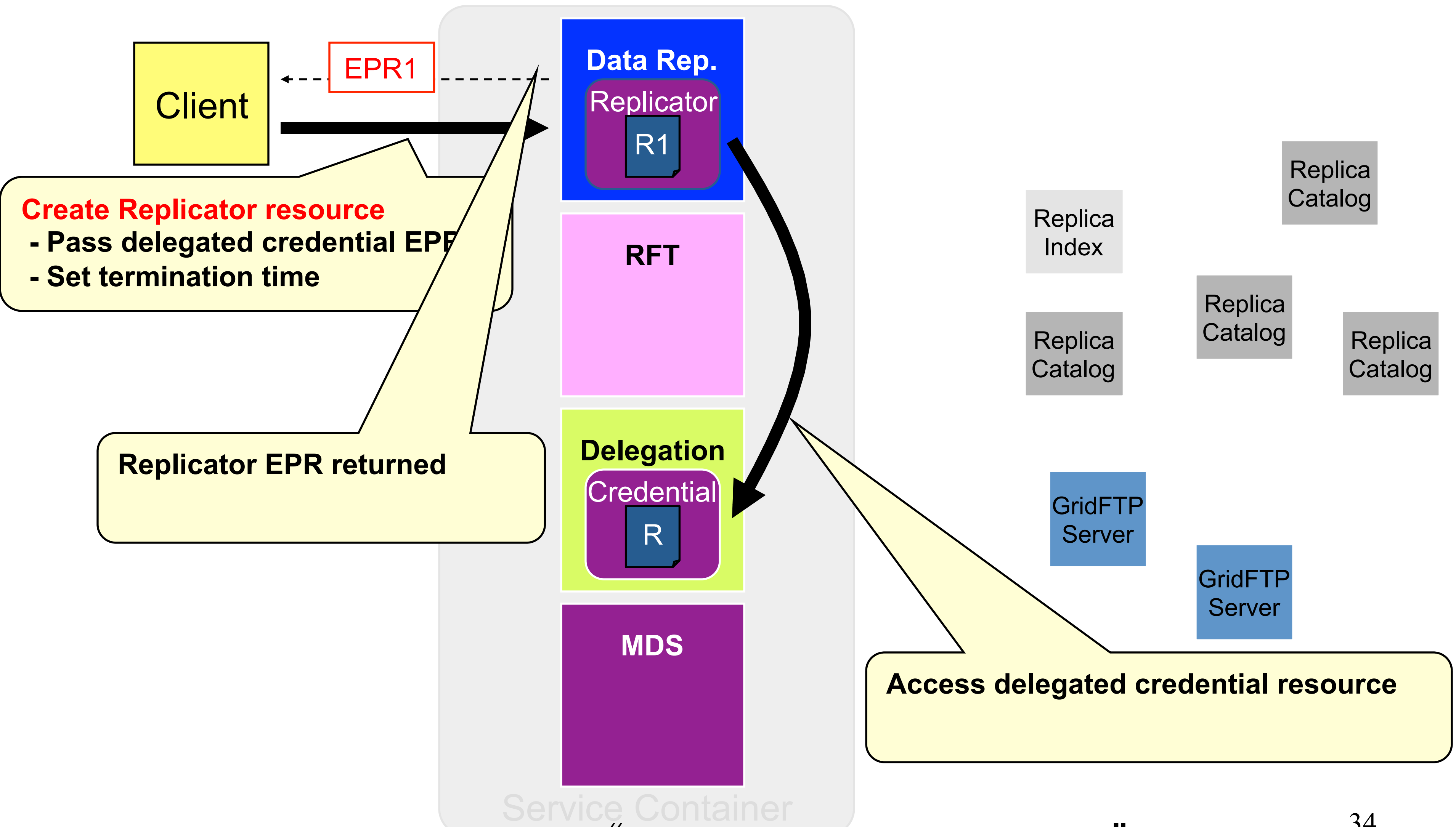
Create a data replica (step 1)



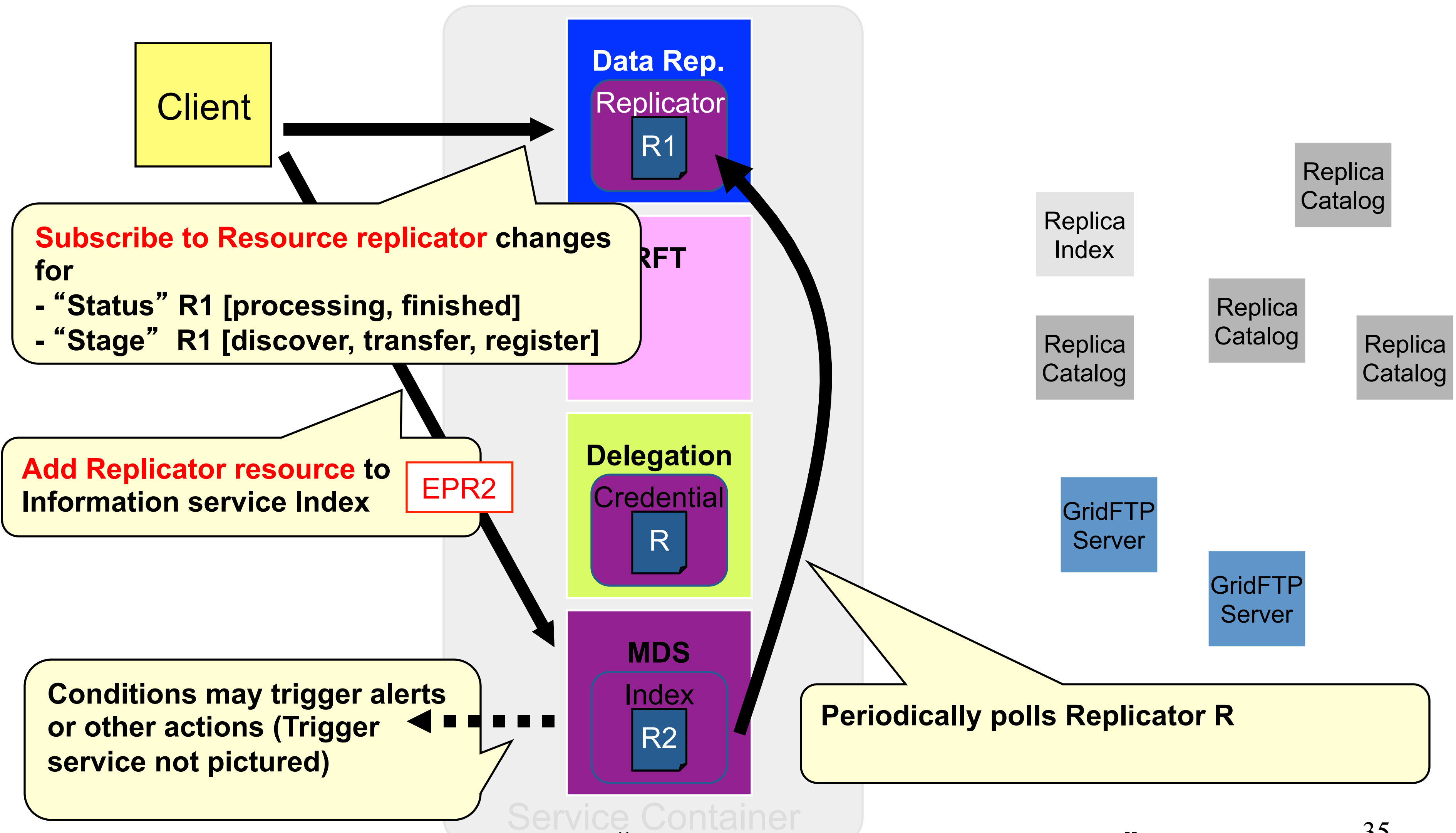
EPR: End Point Reference

Ann Chervenak, Robert Schuler "Globus Data Replication Services" USC ISI

Create a data replica (step 2)

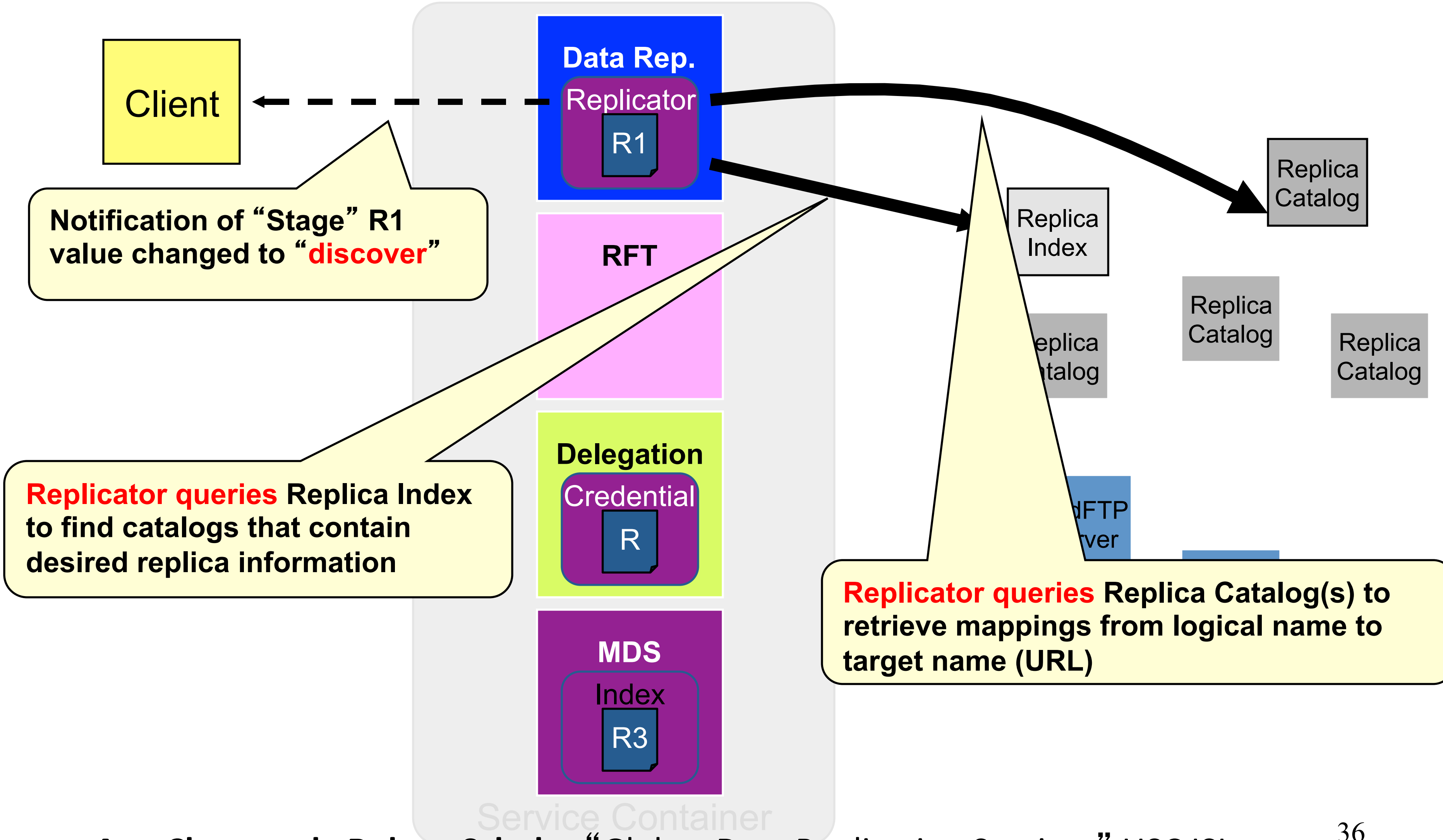


Create a data replica (step 3)

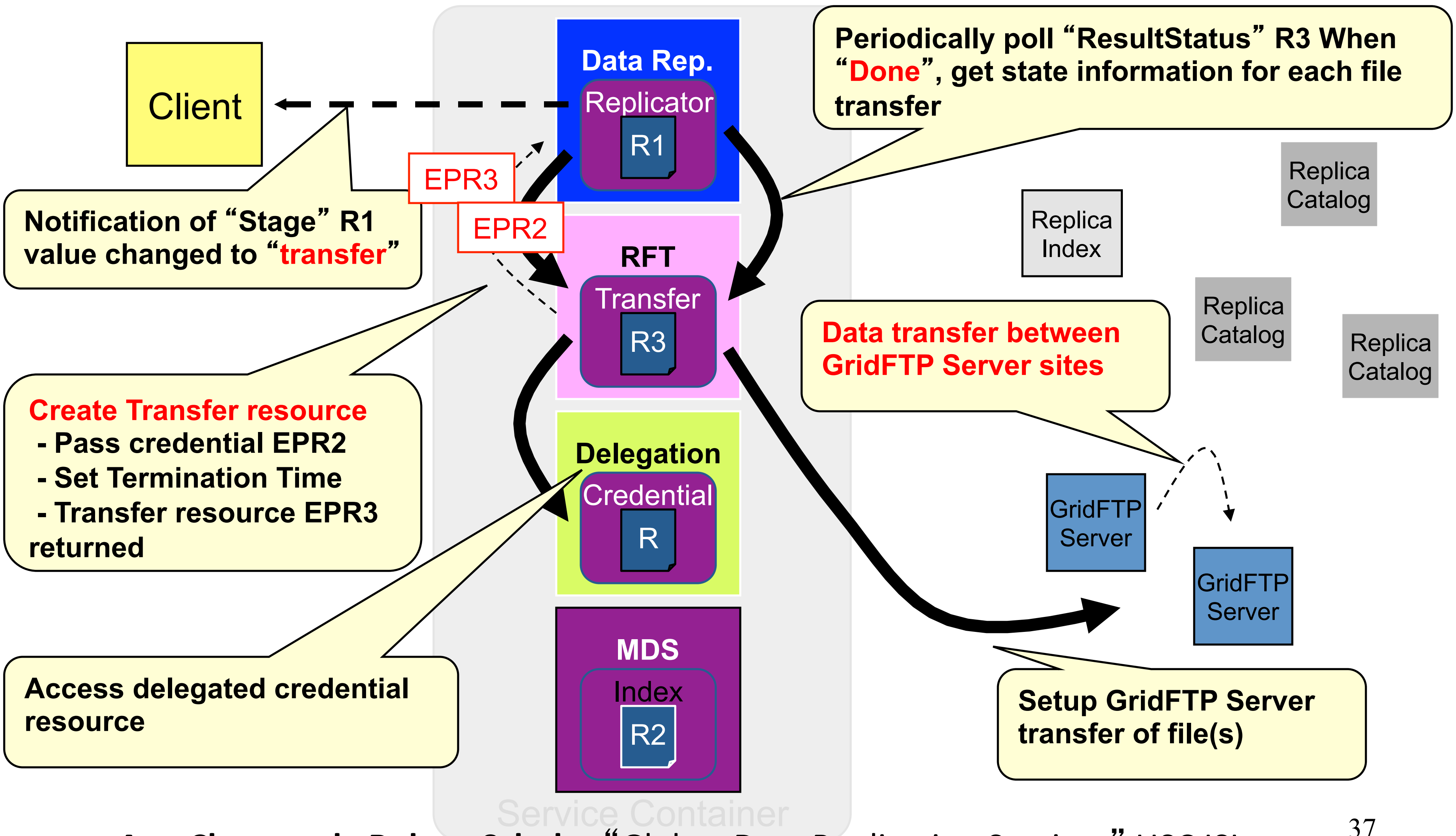


Ann Chervenak, Robert Schuler "Globus Data Replication Services" USC ISI

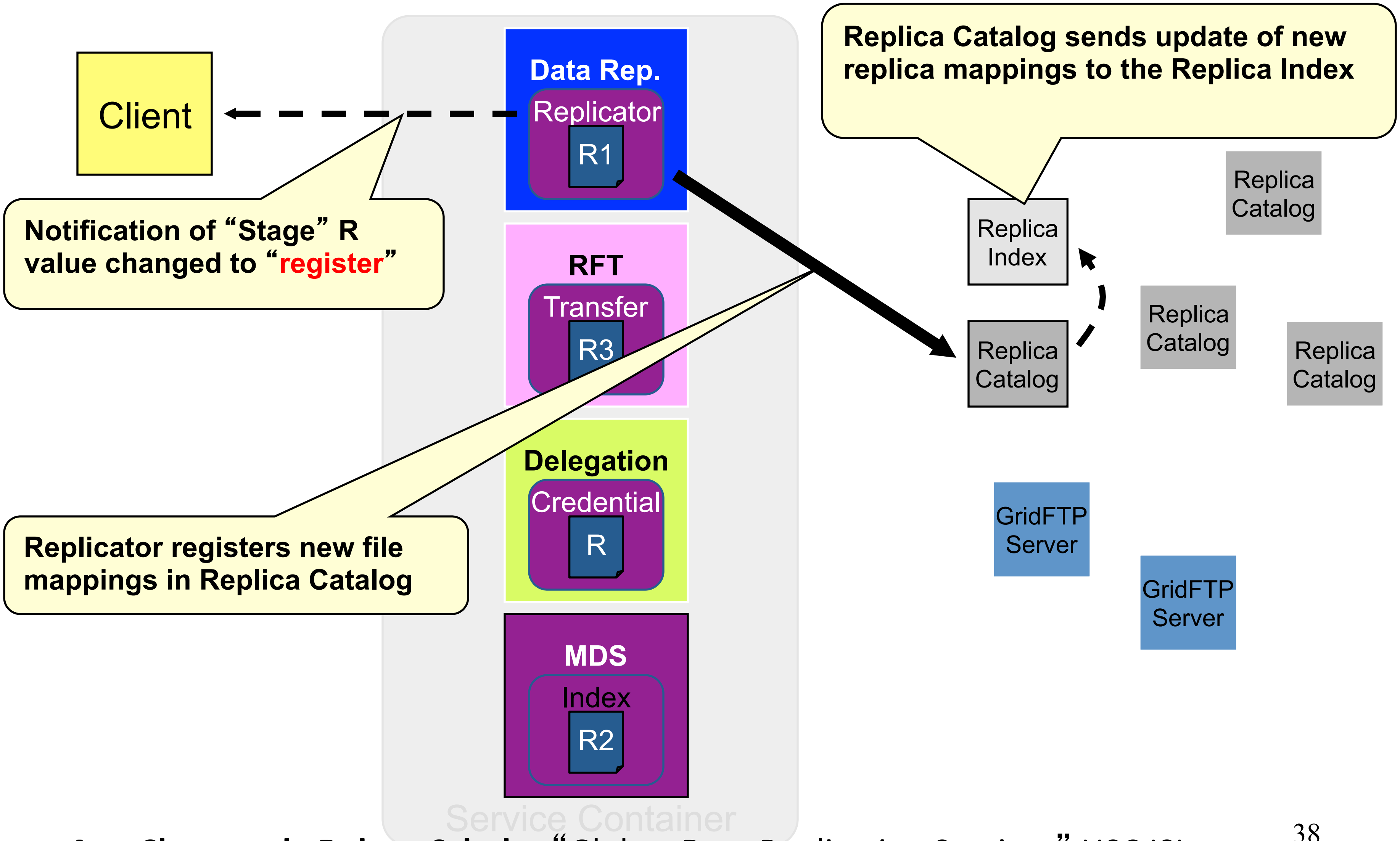
Create a data replica (step 4)



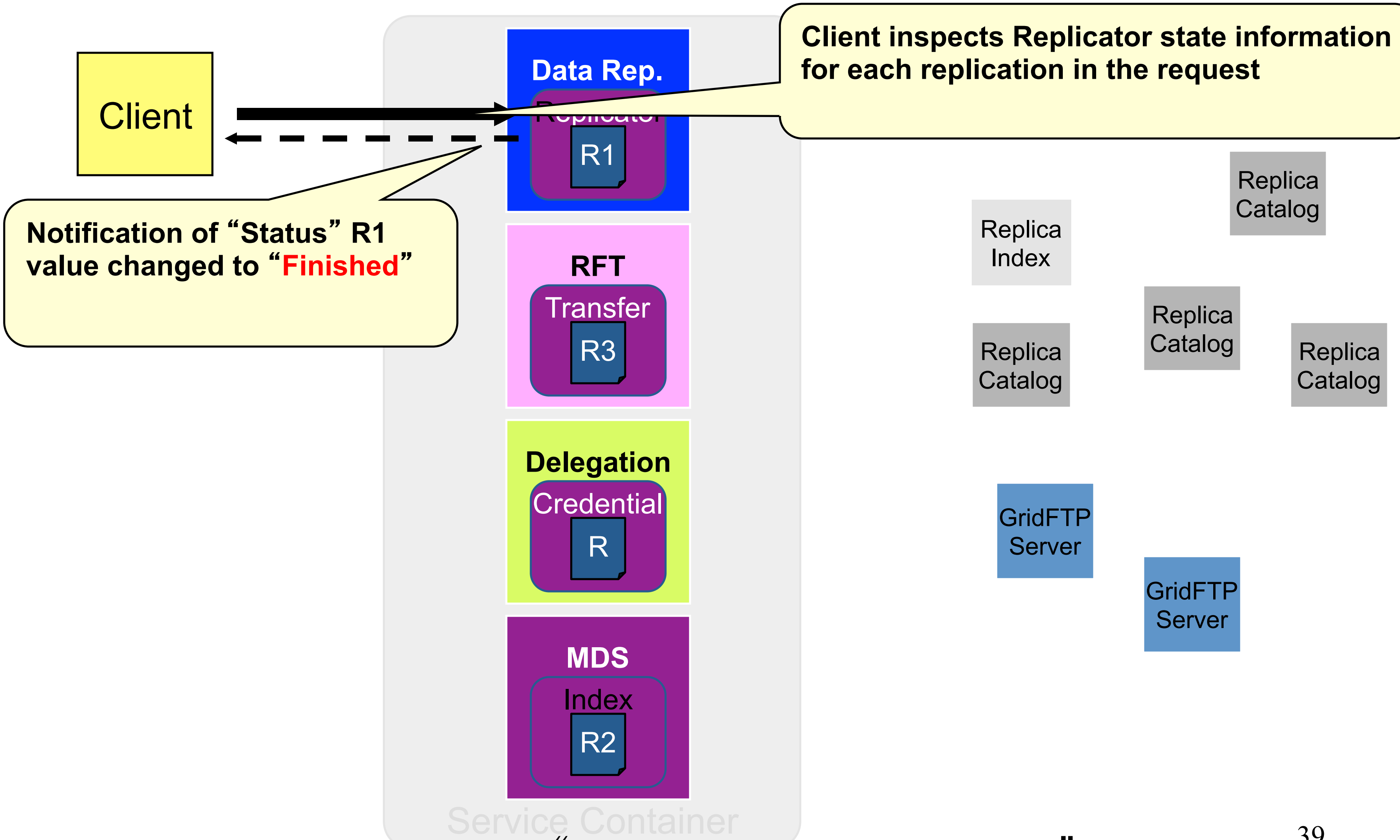
Create a data replica (step 5)



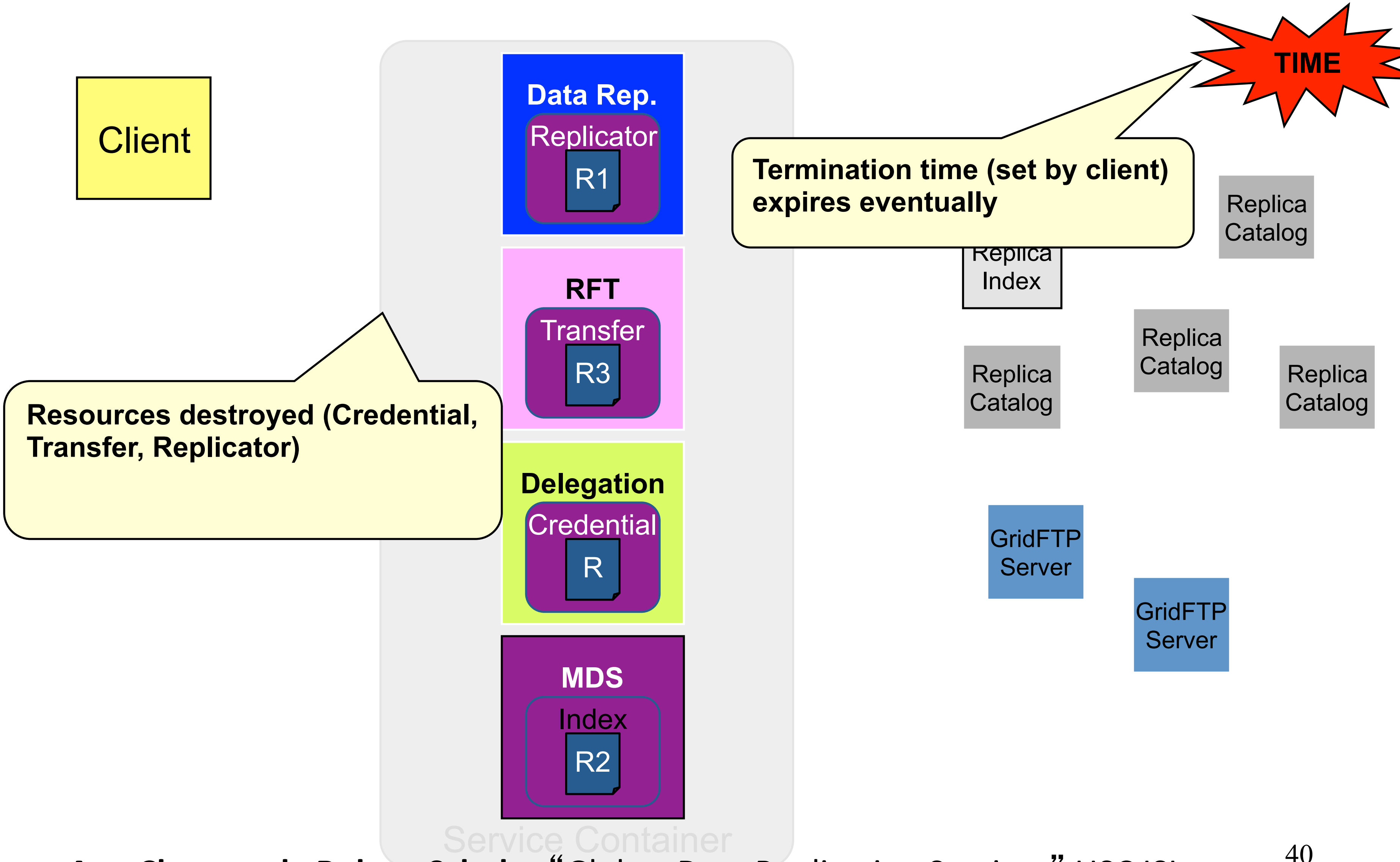
Create a data replica (step 6)



Create a data replica (step 7)

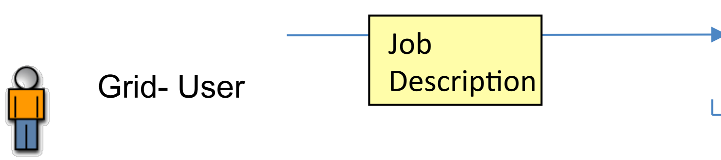


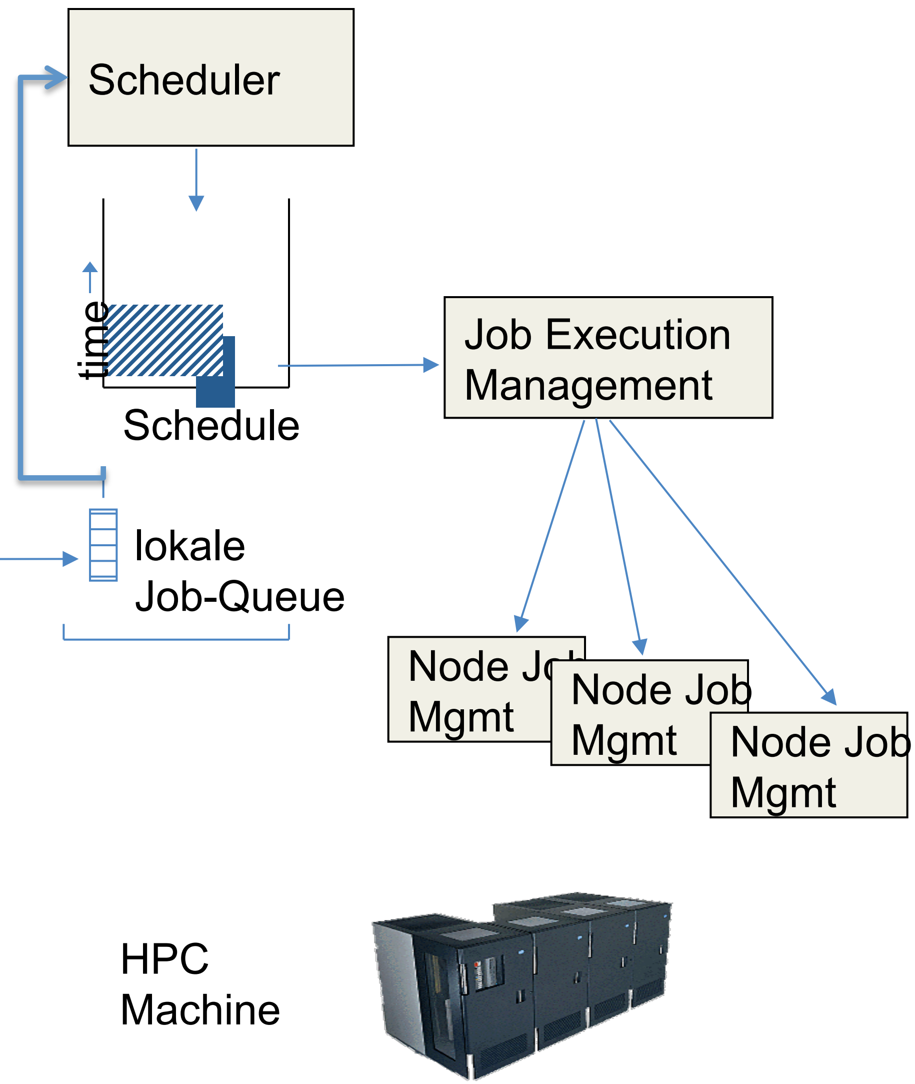
Create a data replica (step 8)



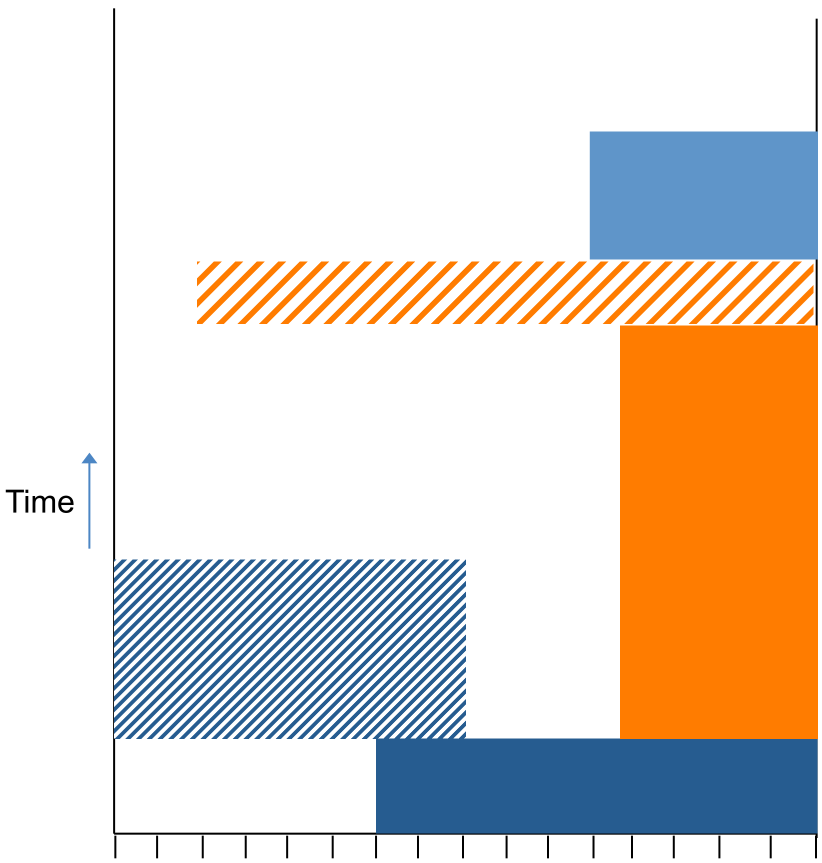
Grid Resource Management

- A user job enters a job queue,
- Scheduler decides when to start the job and resource allocation of the job.

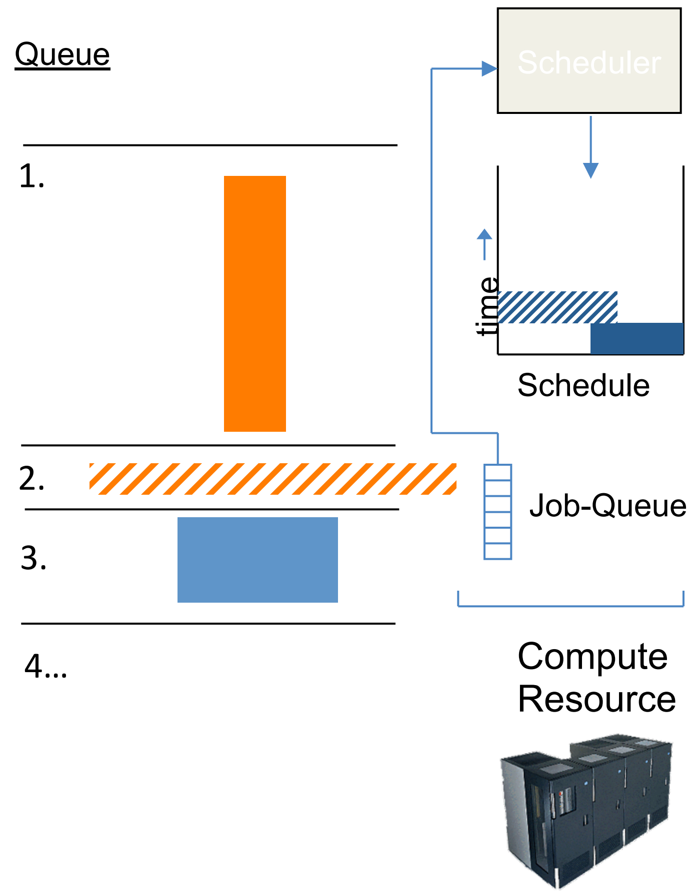




Grid Job Scheduling Scenario



Resources
Procssing Nodes



What is Grid Computing

- Grid computing is the use of hundreds, thousands, or millions of geographically and organizationally disperse and diverse resources to solve:

  ➔ problems that require more computing power than is available from a single machine or from a local area distributed system

Potential Grid Application

- An application which requires the grid solution is likely distributed (Distributed Computing) and fit in one of the following paradigms:
  - High throughput Computing
  - High performance Computing

Grid computing will be mainly needed for large-scale, high-performance computing.

Distributed Computing

- Distributed computing is a programming model in which processing occurs in many geographically distributed places.
  - Processing can occur wherever it makes the most sense, whether that is on a server, Web site, personal computer, etc.

- Distributed computing and grid computing either
  - overlap or distributed computing is a subset of grid computing

From "The Anatomy of the Grid: Enabling Scalable Virtual Organizations" Foster et al

High Throughput Computing

- HTC employs large amounts of computing power for very lengthy periods
  - HTC is needed for doing sensitivity analyses, parametric studies or simulations to establish statistical confidence.
- The features of HTC are
  - Availability of computing power for a long period of time
  - Efficient fault tolerance mechanism
- The key to HTC in grids
  - Efficiently harness the use of all available resources across organizations

High Performance Computing

- HPC brings enormous amounts of computing power to bear over relatively short periods of time.
  - HPC is needed for decision-support or applications under sharp time-constraint, such as weather modeling
- HPC applications are:
  - Large in scale and complex in structure.
  - Real time requirements.
  - Ultimately must run on more than one type of HPC system.

HPC/HTC requirements

- HPC/HTC requires a balance of computation and communication among all resources involved.
  - Managing computation,
  - communication,
  - data locality

Programming Model for the grid

- To achieve petaflop rates on tightly/loosely coupled grid clusters, applications will have to allow:
  - extremely large granularity or produce massive parallelism such that high latencies can be tolerated.

- This type of parallelism, and the performance delivered in a heterogeneous environment, is
  - currently manageable by hand-coded applications

Programming Model for the grid

- A programming model can be presented in different forms: a language, a library API, or a tool with extensible functionality.

- The successful programming model will
  - enable both high-performance and the flexible composition and management of resources.
  - influence the entire software lifecycle: design, implementation, debugging, operation, maintenance, etc.
  - facilitate the effective use of all manner of development tools, e.g., compilers, debuggers, performance monitors, etc

Grid Programming Issues

- Portability, Interoperability, and Adaptability
- Discovery
- Performance
- Fault Tolerance
- Security

Programming models

- Shared-state models
- Message passing models
- RPC and RMI models
- Peer to Peer Models
- Web Service Models
- …

References

- Ian Foster, Carl Kesselman, Steven Tuecke The ***Anatomy of the Grid: Enabling Scalable Virtual Organizations*** International Journal of High Performance Computing Applications Fall 2001 15: 200-222.

- Ian Foster, **The physiology of the grid: An open grid services architecture for distributed systems integration,** (2002)