European Master on

Theoretical Chemistry and Computational Modelling

Grid platforms and Services

The Grid architecture

Dept. of Chemistry, Univ. P. Sabatier, Toulouse, France

September 4-5, 2008

Osvaldo Gervasi

Dept of Mathematics and Computer Science

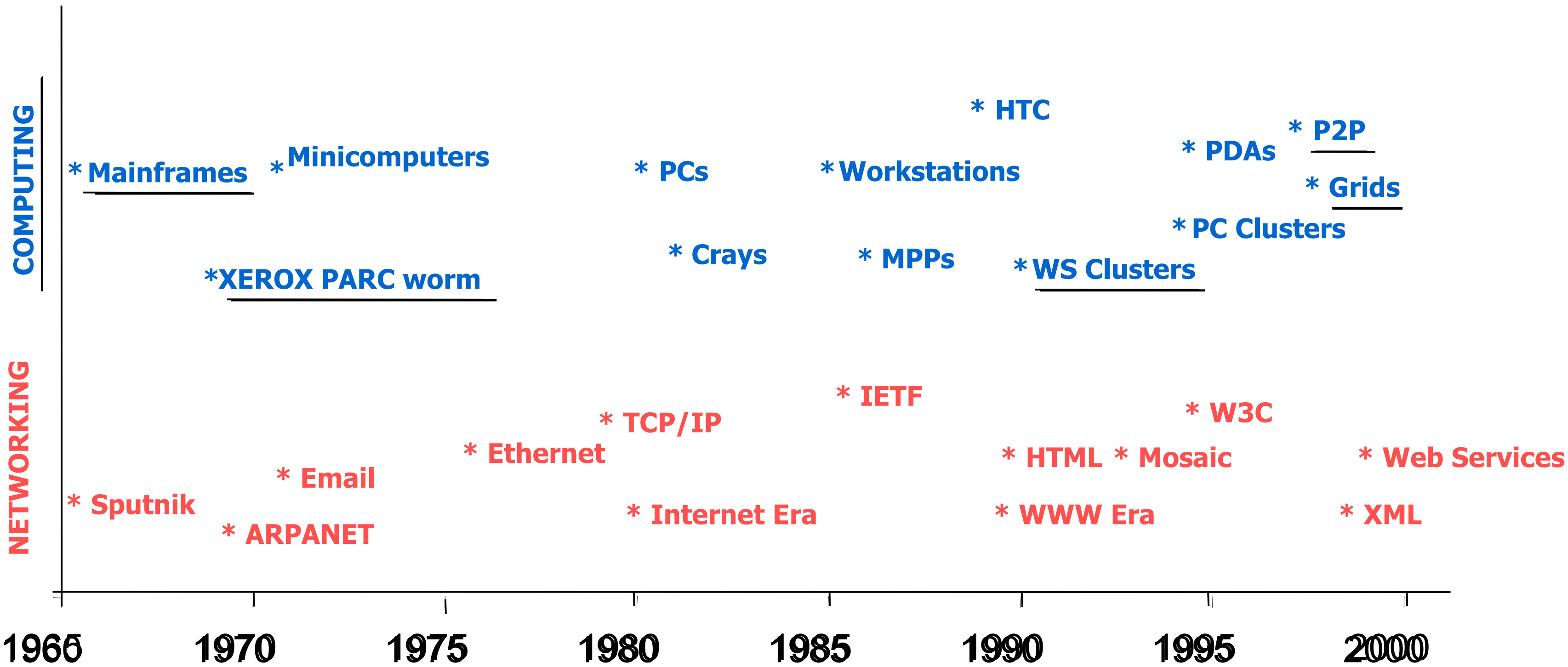
University of Perugia

osvaldo@unipg.it

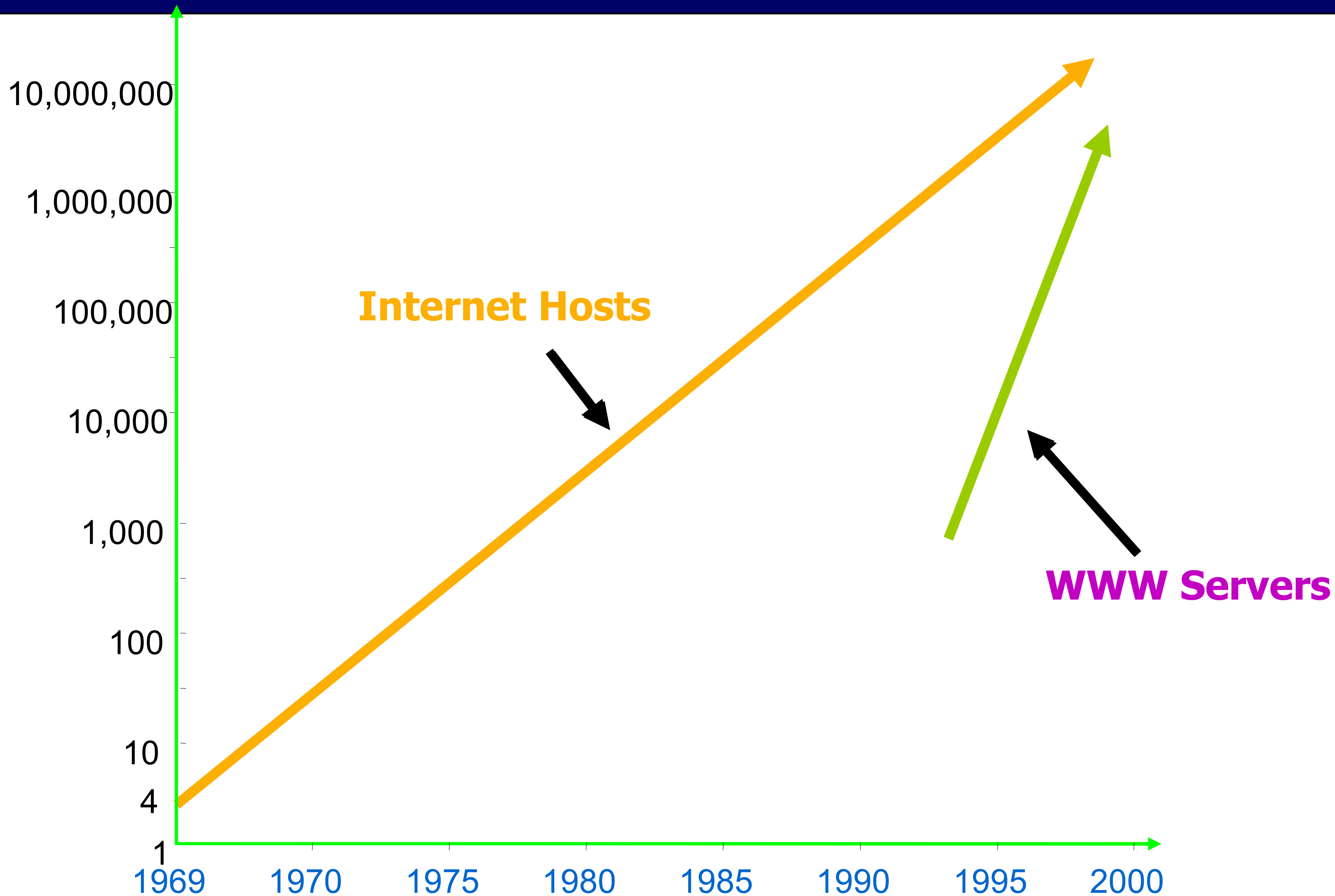
Outline

- Introduction to Grid Computing
  - Scientific environments and projects, big challanges
  - The EGEE Grid
- Grid Architecture
  - Some Definitions
  - The Programming Problem
- The Globus Toolkit™
  - Security
  - Resource Management
  - Information Services
  - Data Management

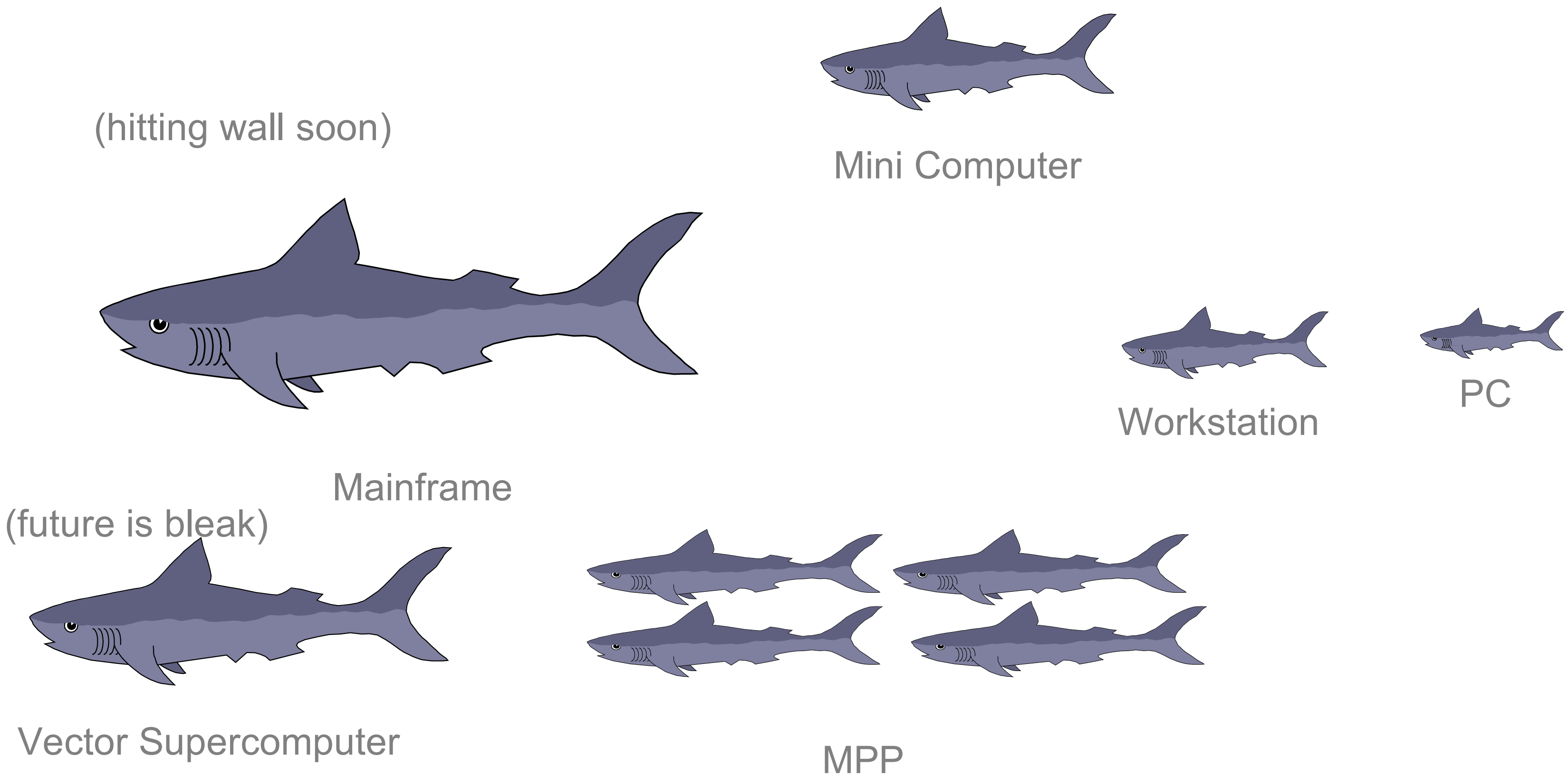
Computing Evolution Time Line



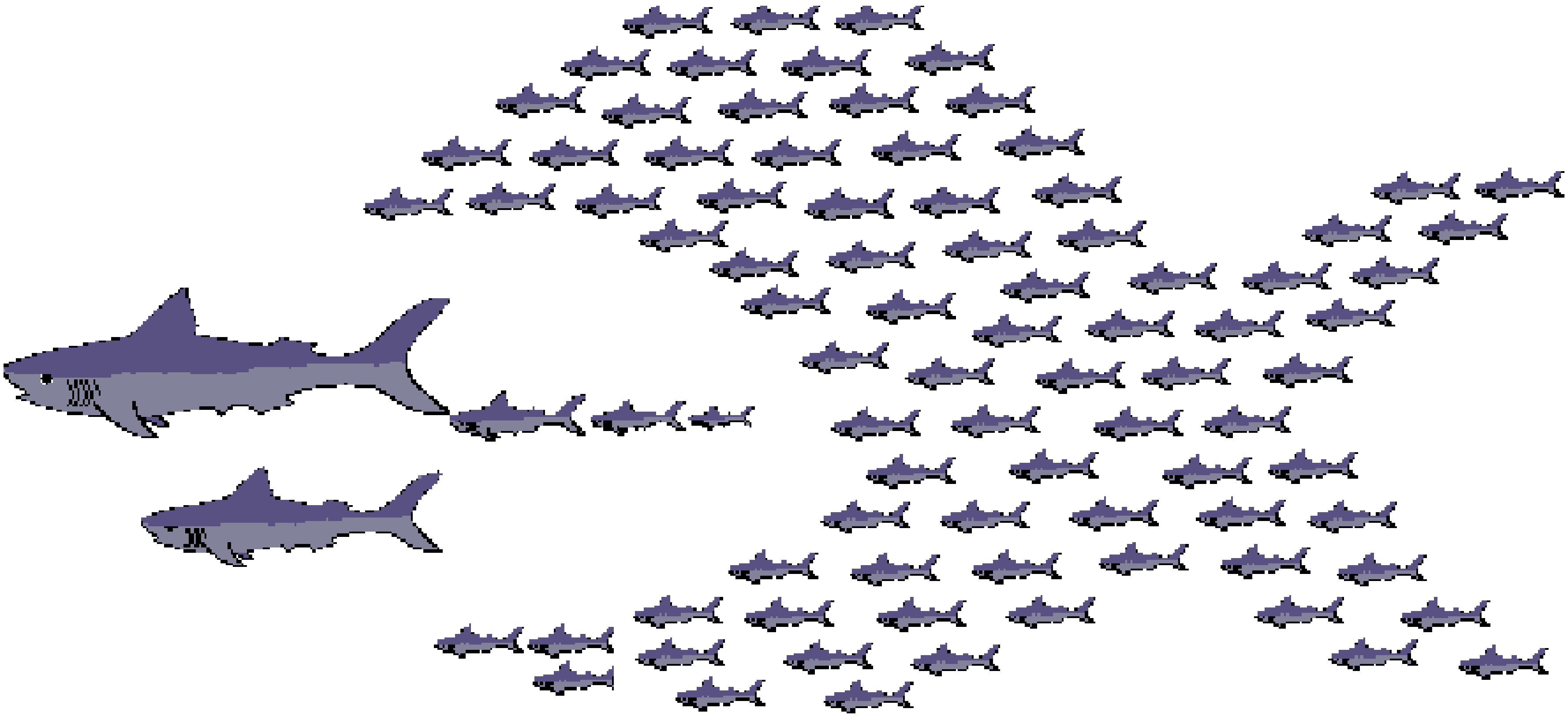
Crescita di Internet e del WWW



Computers of early '90s



The clusters era



Cluster's applications

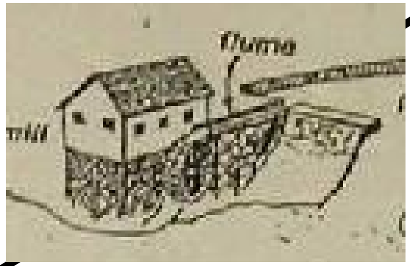
- Scientific and Engineering Applications
- Business Applications:
  - E-commerce (Amazon, eBay ….);
  - Database (Oracle on cluster).
- Internet Applications:
  - ASPs (Application Service Provider);
  - Computing Portal;
  - E-commerce and E-business.
- Critical Applications:
  - command control system, bank, control of risky processes

Grid Computing

- Unification of resources geographically distributed

The (Power) Grid:
On-Demand Access to Electricity

Quality, economies of scale



Decouple production & consumption, enabling

- On-demand access
- Economies of scale
- Consumer flexibility
- New devices

Time

(by Ian Foster)

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

But Computing Isn't Really Like Electricity!

- How about "access computing resources like we access Web content"?
  - We have no idea where a website is, or on what computer or operating system it runs

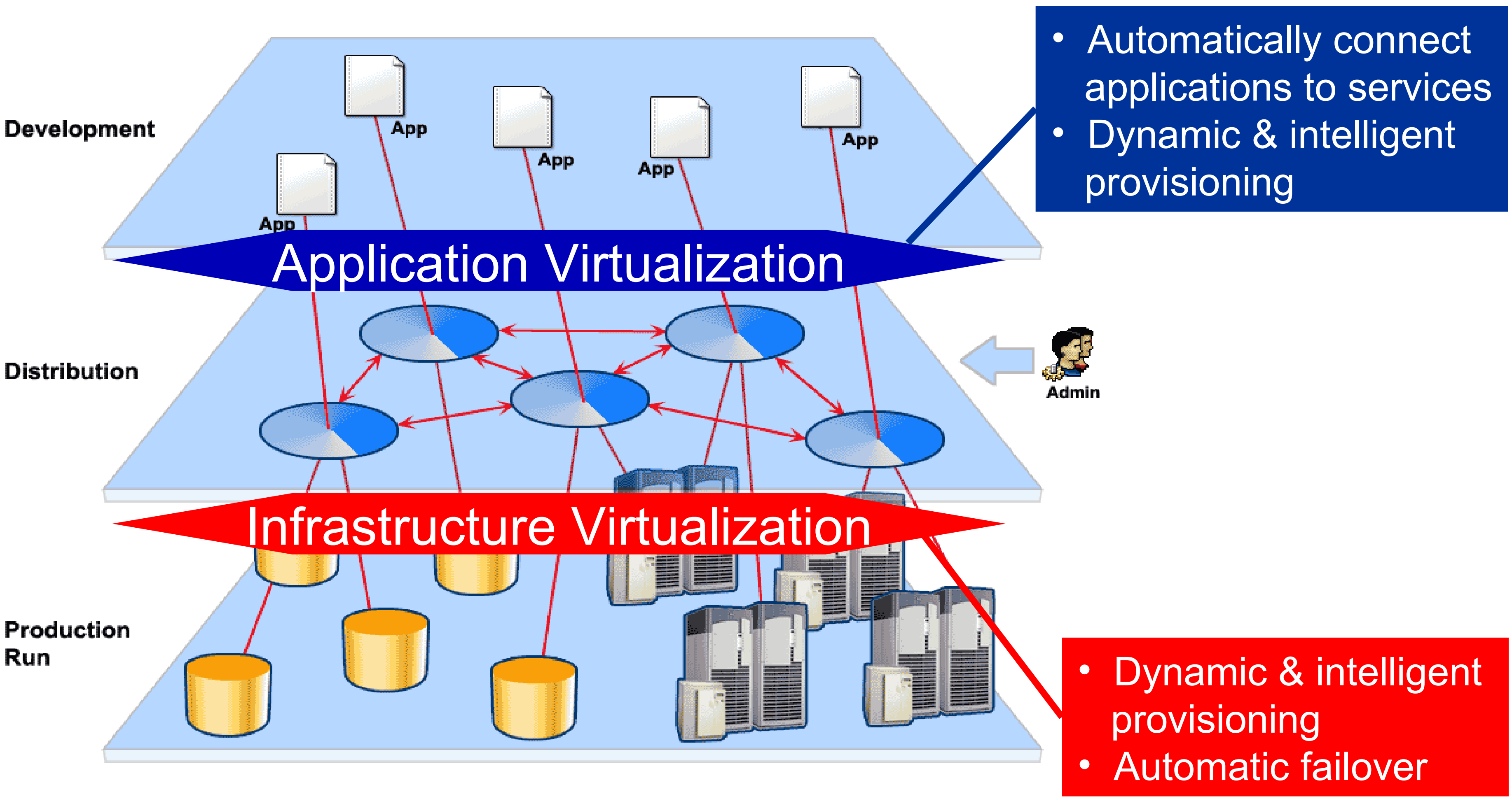
⇒ Two interrelated opportunities

1) Enhance economy, flexibility, access by **virtualizing** computing resources

2) Deliver entirely new capabilities by **integrating** distributed resources

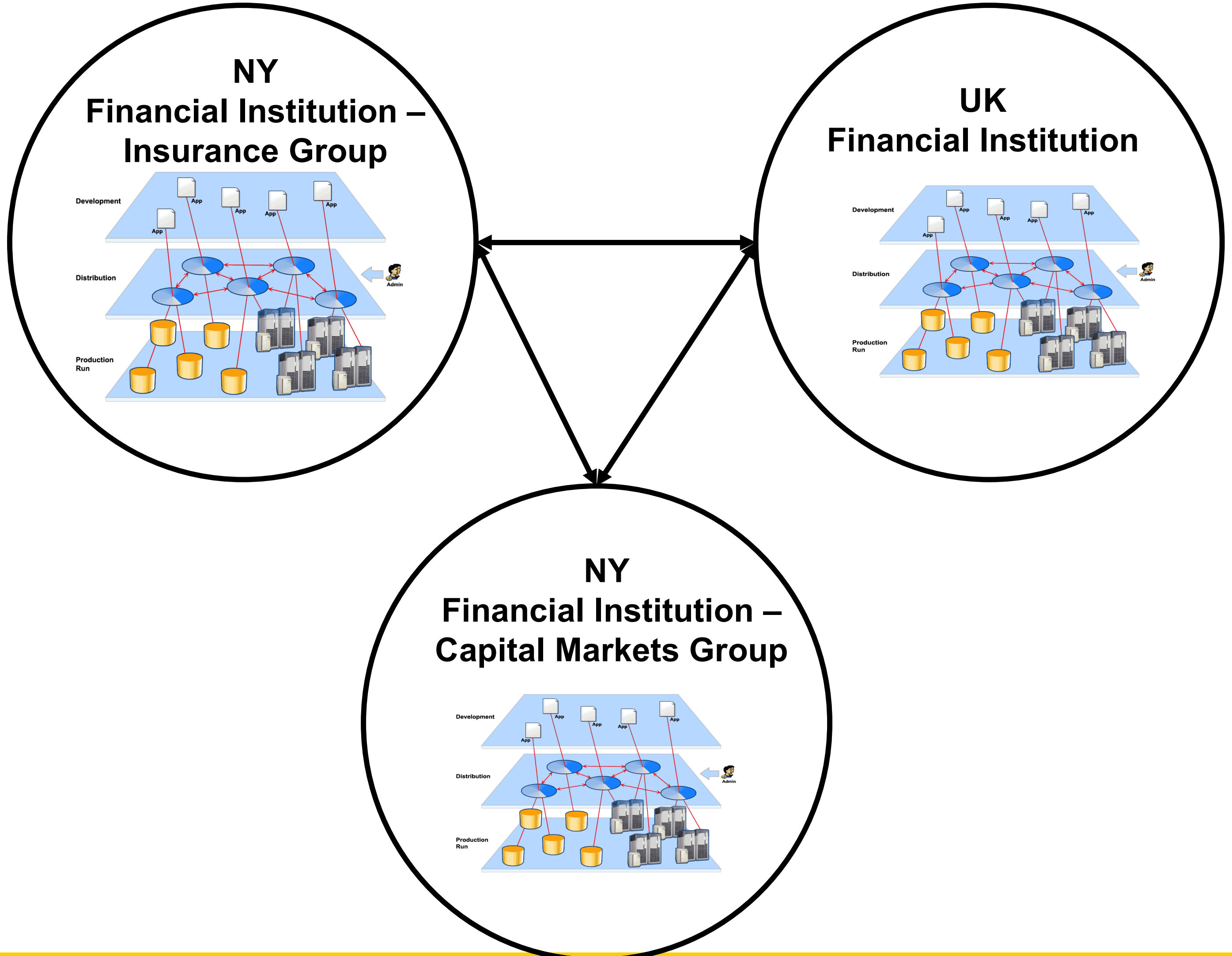
(by Ian Foster)

Virtualization

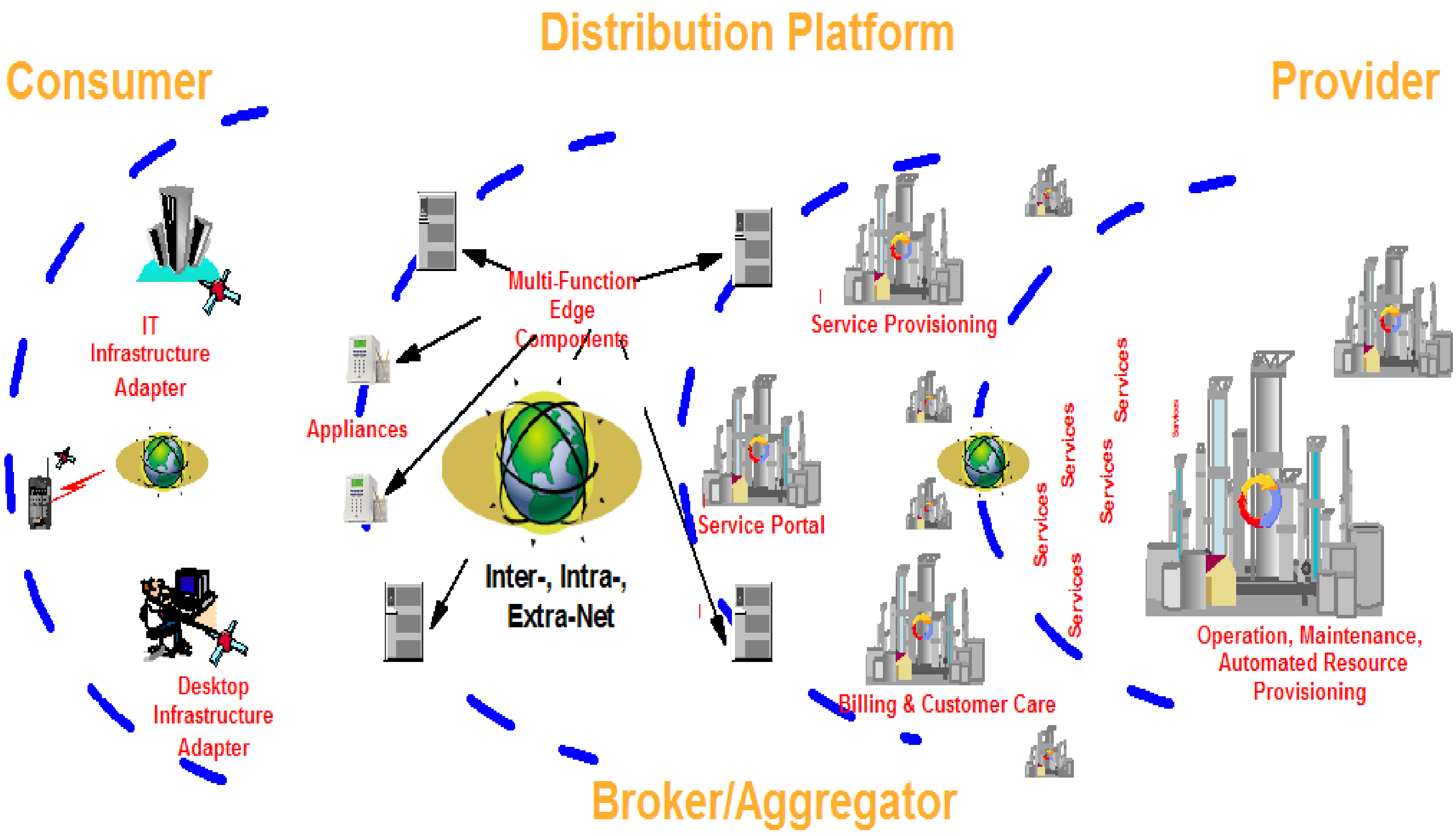


Source: *The Grid: Blueprint for a New Computing Infrastructure (2nd Edition)*, 2004

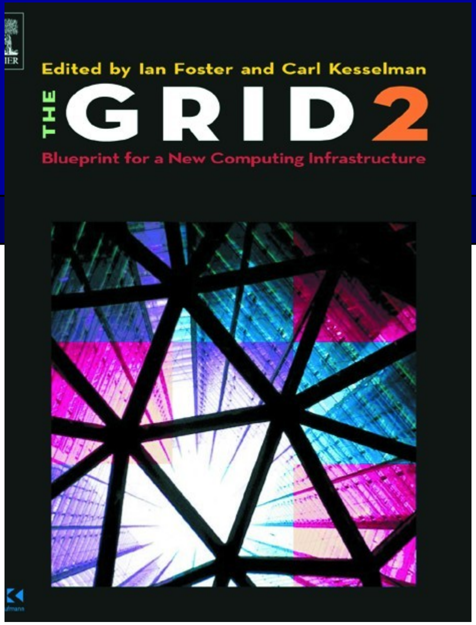
Distributed System Integration



And ultimately …

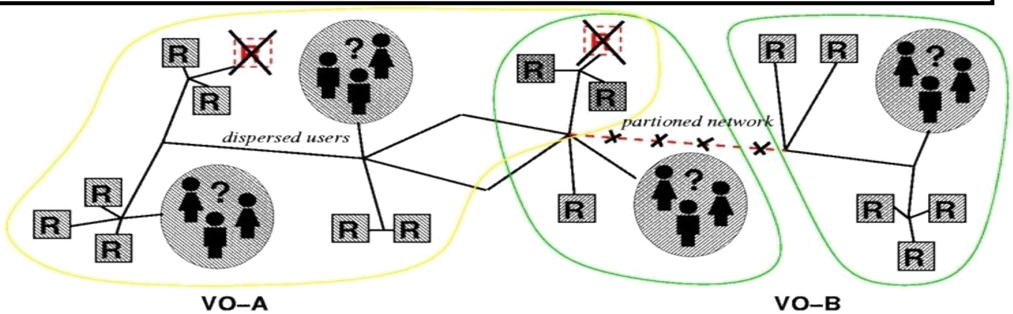


The Grid



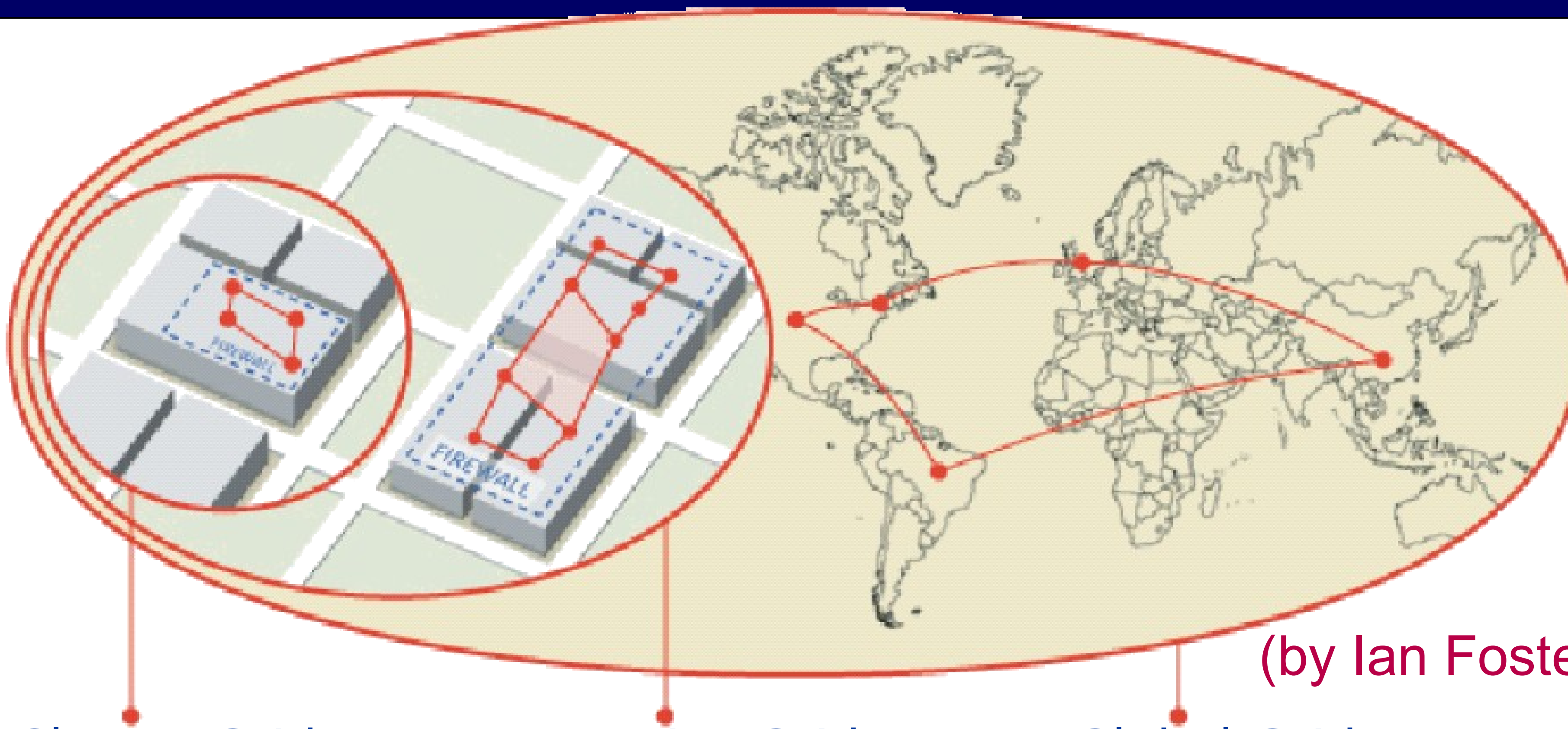
Enable *"coordinated resource sharing & problem solving in dynamic, multi-institutional virtual organizations."*

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



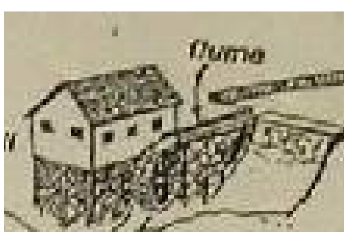
- Access to shared resources
  - → Virtualization, allocation, management
- With predictable behaviors
  - → Provisioning, quality of service
- In dynamic, heterogeneous environments
  - → Standards-based interfaces and protocols

Terminoplogy

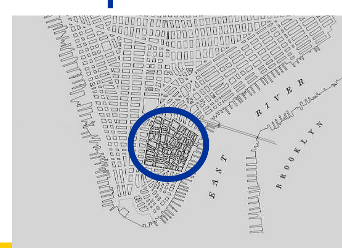


(by Ian Foster)

Cluster Grid



Enterprise Grid

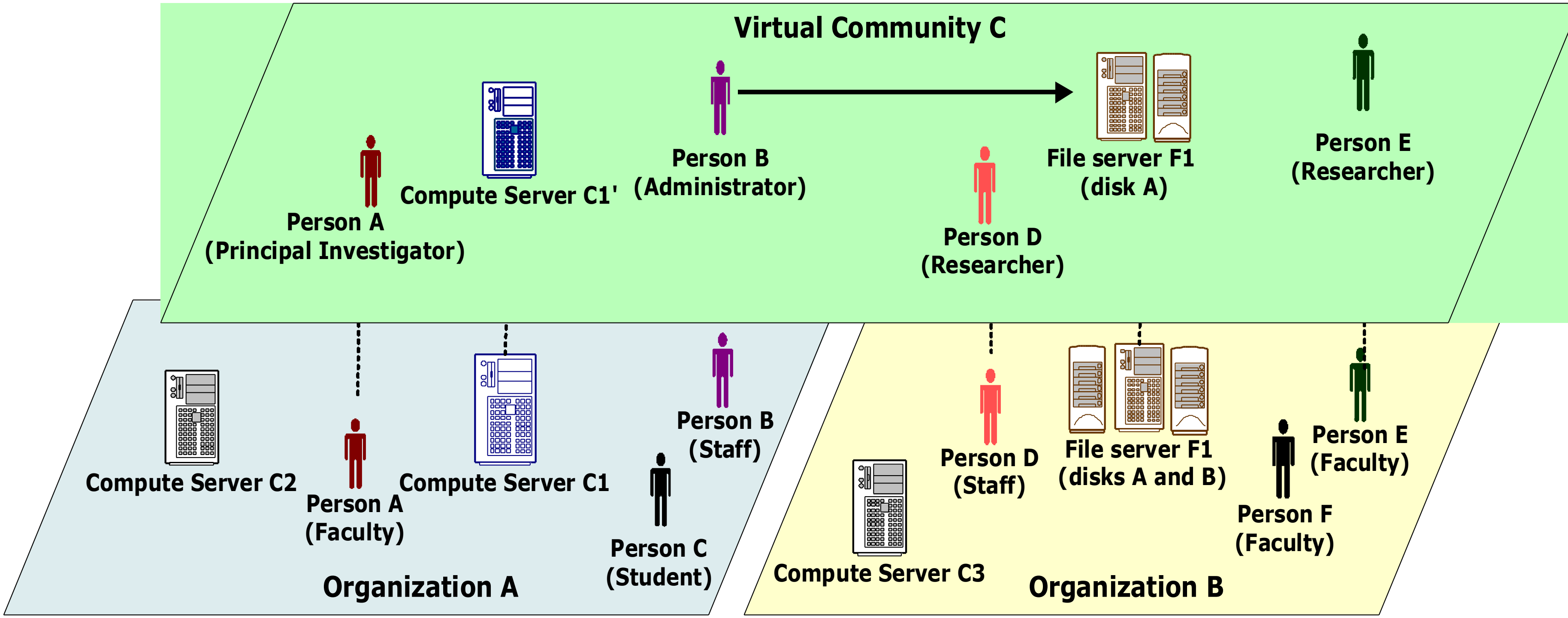


Global Grid

Why should you care?

- 1) Grid is a disruptive technology [**Vision**]
  - It ushers in a virtualized, collaborative, distributed world
- 2) Grid addresses pain points now [**Reality**]
  - Grids are built not bought, but are delivering real benefits in commercial settings
- 3) An open Grid is to your advantage [**Future**]
  - Standards are being defined now that will determine the future of this technology

The VO concept



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

What is the Grid

- Flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions and resources – what we refer to as Virtual Organization (VO).
- The sharing process is controlled using techniques that control what can be shared, how it must be shared, the operating conditions of the sharing process.
- Grid technologies enable the resource sharing among Internet by interconnecting resources and services of different organizations, belonging to the same VO.

Grid issues

- Usage of open interfaces and standard protocols
- Usage of smart tecniques to handle security issues
- Guatantee an adequate Quality of Service
- Management of distributed heterogeneous resources

Outline

- Introduction to Grid Computing
  - Scientific environments and projects, big challanges
  - The EGEE Grid
- Grid Architecture
  - Some Definitions
  - The Programming Problem
- The Globus Toolkit™
  - Security
  - Resource Management
  - Information Services
  - Data Management

Grid Computing is in the News …

RED HERRING

TOP TEN TRENDS 2001

DISTRIBUTE THE WEALTH

Distributed computing initiatives.

GLOBUS PROJECT

Globus is a research and development project focused on enabling the application of grid computing concepts to scientific and engineering computing. The grid is an emerging infrastructure protocol that enables the integrated use of remote high-end computers, databases, scientific instruments, networks, and other resources.

I.B.M. Making a Commitment to Next Phase of the Internet

The New York Times

By STEVE LOHR

I.B.M. is announcing today a new initiative to support and exploit a technology known as grid computing, which the company and much of the computer research community say is the

Computing power on tap

Economist.com

Jun 21st 2001
From The Economist print edition

In the first of two articles, we look at the most ambitious attempt yet to combine millions of computers seamlessly around the world—to make processing power available on demand anywhere, rather like electrical power.

COVER STORY

ORACLE DATABASE 10g
THE WORLD'S FIRST
SELF-MANAGING, GRID–READY DATABASE ARRIVES

Oracle's new self-managing database increases performance and availability while enabling commercial grid computing.

There are trends. And there are trendsetters. In the early to mid-1990s, Oracle foretold the internet computing paradigm that organizations of every stripe have now woven into the fabric of their businesses. In the process, IT infrastructure has become extremely critical to the enterprise. "Businesses have become more dependent than ever on their IT systems for everything from day-to-day operations to providing service to their customers and clients," says Sushil Kumar, director of product management for Database Manageability at Oracle. "And many new-generation businesses, such as eBay and Amazon, rely completely on their IT infrastructure's being available—if the system goes away, their entire business is in jeopardy."

In short, says Kumar, IT systems have truly become strategic to the enterprise. And that has had a profound impact on the need for availability, scalability, and high performance of IT systems for organizations of all kinds. Downtime, even for much-needed maintenance, is not an option when a global business must run 24/7.

At the same time, says Kumar, there's growing pressure to maintain profitability amid ever-growing competition in a global economy that continues to tighten its belt. The result, is that "organizations must minimize operating expenses across the board—and IT is no exception," says Kumar.

But as IT systems have become more strategic and integral to the core business, they have also become more complex, more difficult to manage, and more costly. Complexity adds to costs across the board, in terms of time, labor, potential failures, and inability to recover from failure effectively. According to Kumar, these are all reasons why "one of the biggest challenges facing most organizations today is managing a strategic part of the business, its IT systems, more effectively than ever—ensuring the highest performance, scalability, and availability—but at a significantly lower cost than before." These are also some of the reasons that commercial grid computing, enabled in part by cost-effective blade servers, is getting so much attention today. For small incremental costs, organizations can gain more processing power to be used by all data center resources, delivering faster performance and high availability and scaling as needed—but only if the software can effectively take advantage of that architecture.

Clearly, the time is right for software that monitors and manages itself: software that eases management complexity in a cost-effective manner.

BY KELLI WISETH

28 SEPTEMBER/OCTOBER 2003 OTN.ORACLE.COM/ORACLEMAGAZINE

ORACLE MAGAZINE SEPTEMBER/OCTOBER 2003 29

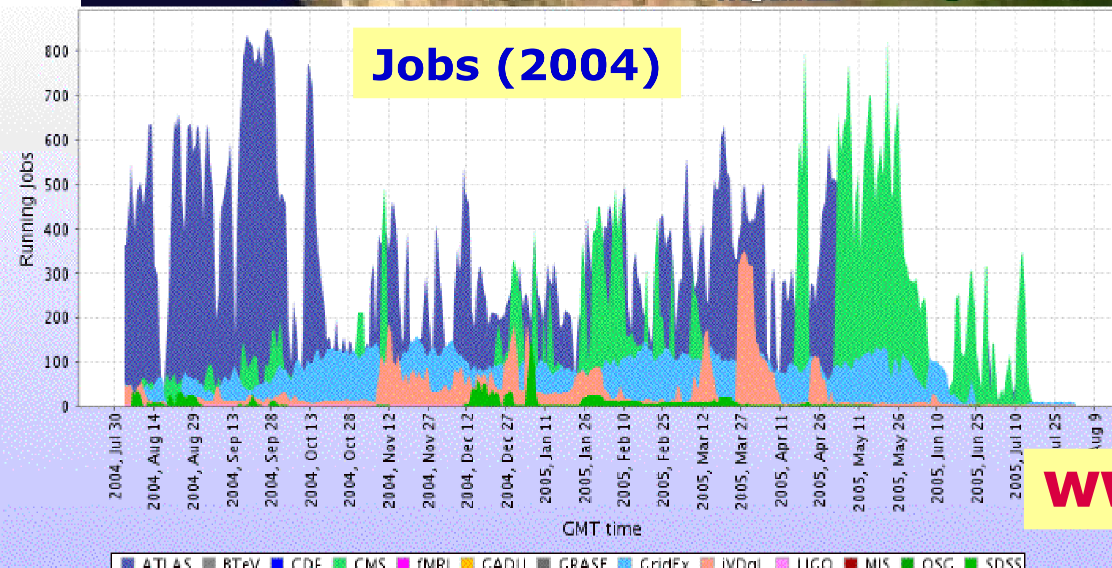
Merrill Lynch

Globus Grid Computing—the Next Internet
by John Roy/Steve Milunovich

The Internet was first a network and is now a communications platform. The next evolutionary step could be to a platform for distributed computing. This ability to manage applications and share data over the network is called "grid computing."

Open Science Grid

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



www.opensciencegrid.org

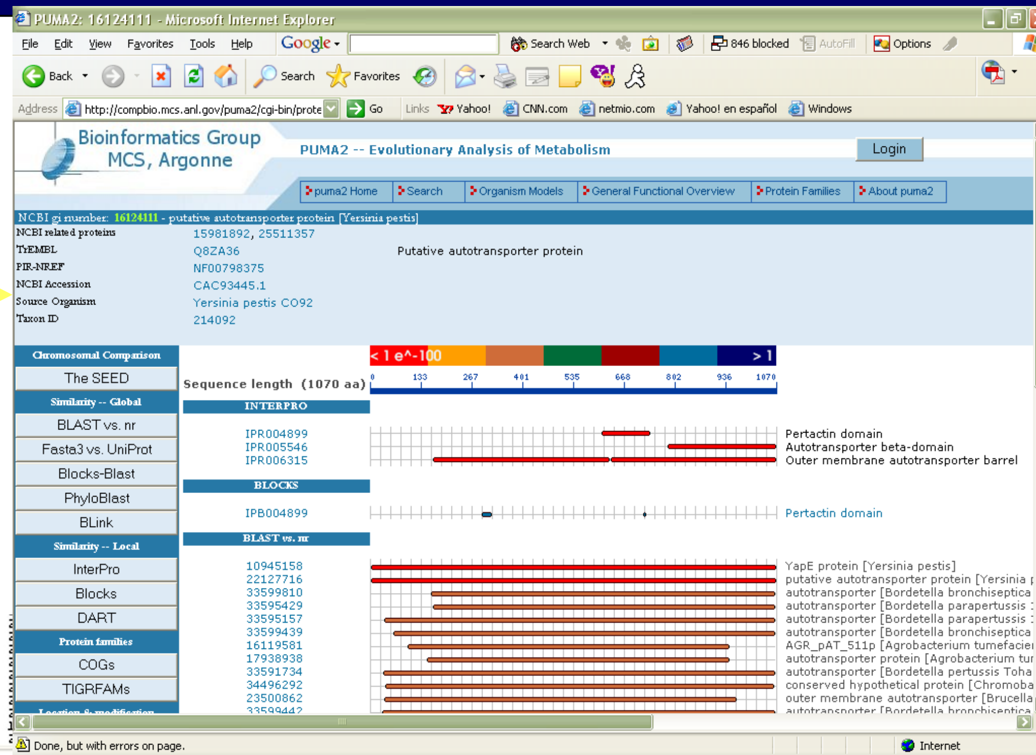
TeraGrid

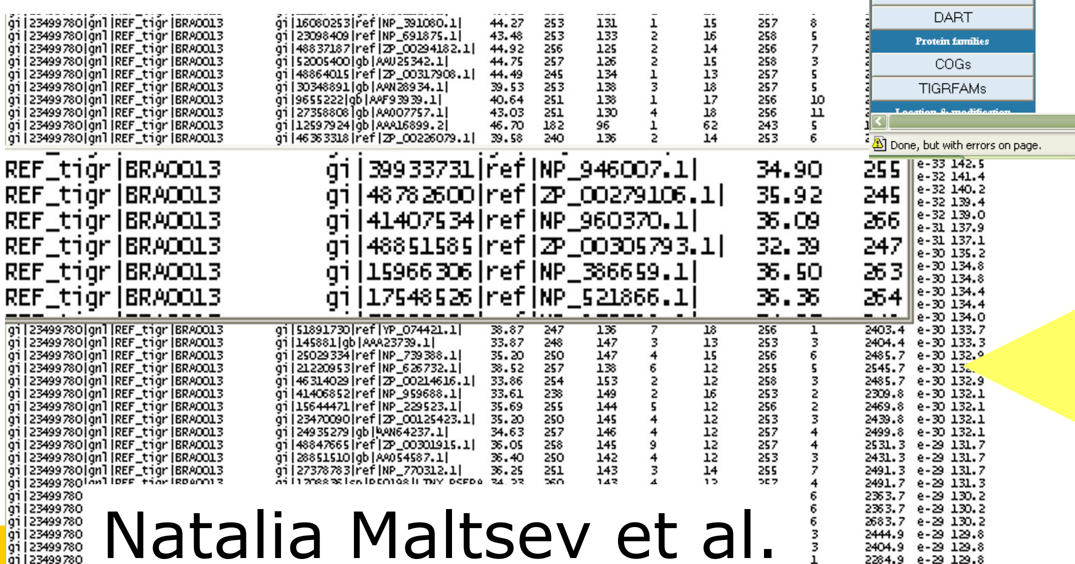
- Access to high-end **capability**
  - 60 Teraflop/s
  - 2 Petabyte online
  - 30 Gb/s net
- High reliability & performance **service hosting**
  - "Science Gateways"
- On-demand access to extra **capacity**

New Knowledge Services

PUMA Knowledge Base

Information about proteins analyzed against ~2 million gene sequences





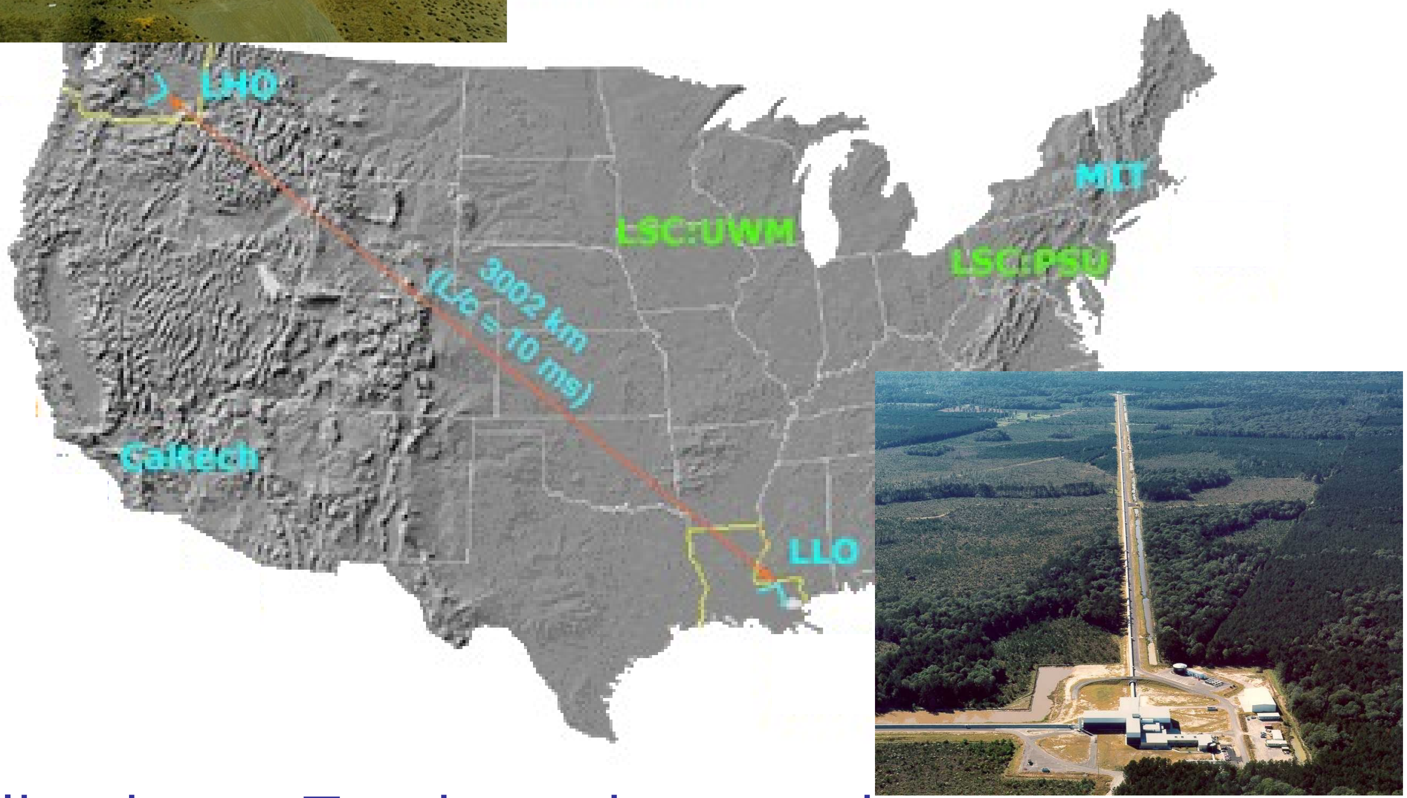
Analysis: TeraGrid

Involves millions of BLAST, BLOCKS, and other processes

Natalia Maltsev et al.

Distributing Astronomy Data

LIGO Gravitational Wave Observatory



Replicating >1 Terabyte/day to 8 sites

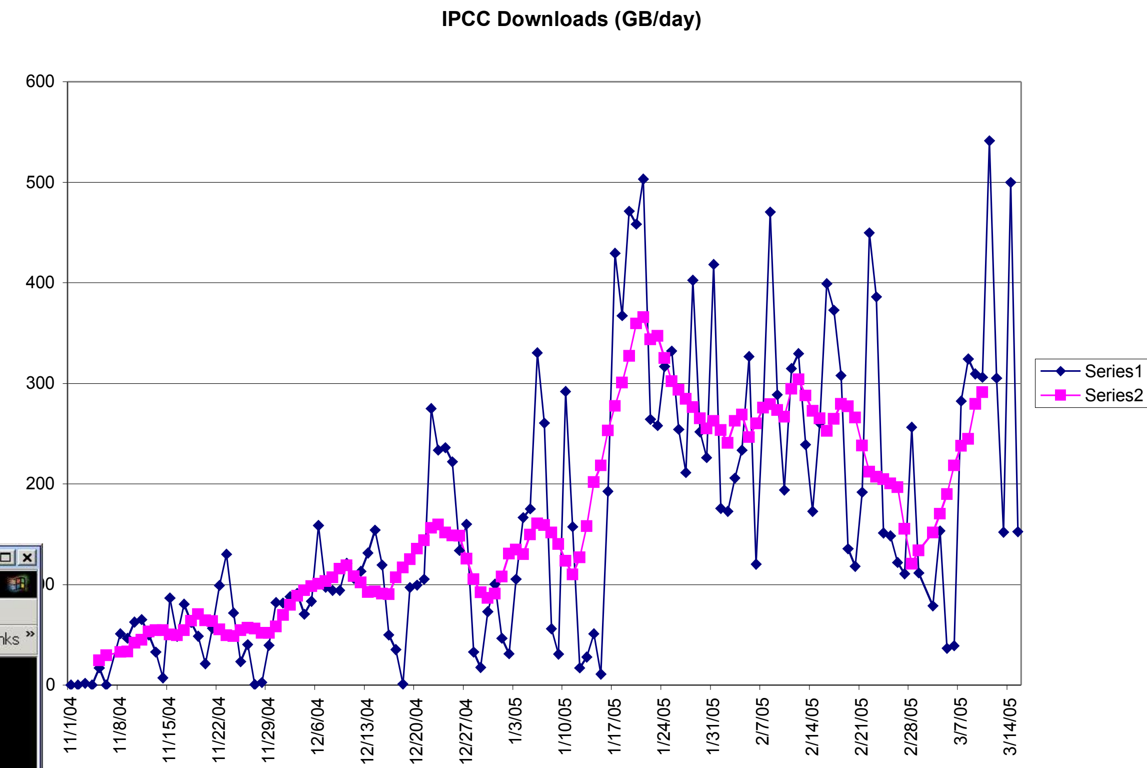
>30 million replicas so far

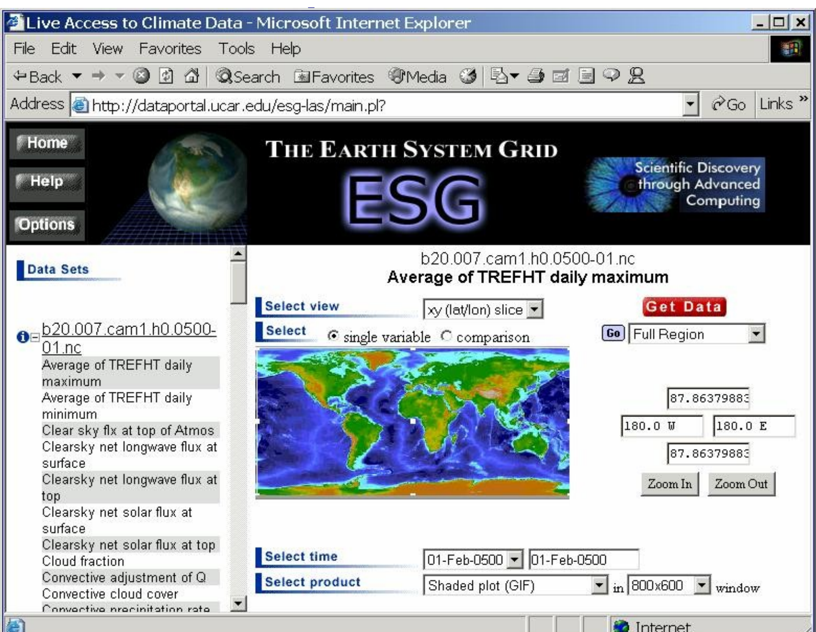
MTBF = 1 month

www.globus.org/solutions

Distributing Climate Model Data

- >100 TB online (18 TB IPCC)
- >1000 users
- >22 TB IPCC served in ~2





250 papers developed, to date, based on IPCC data

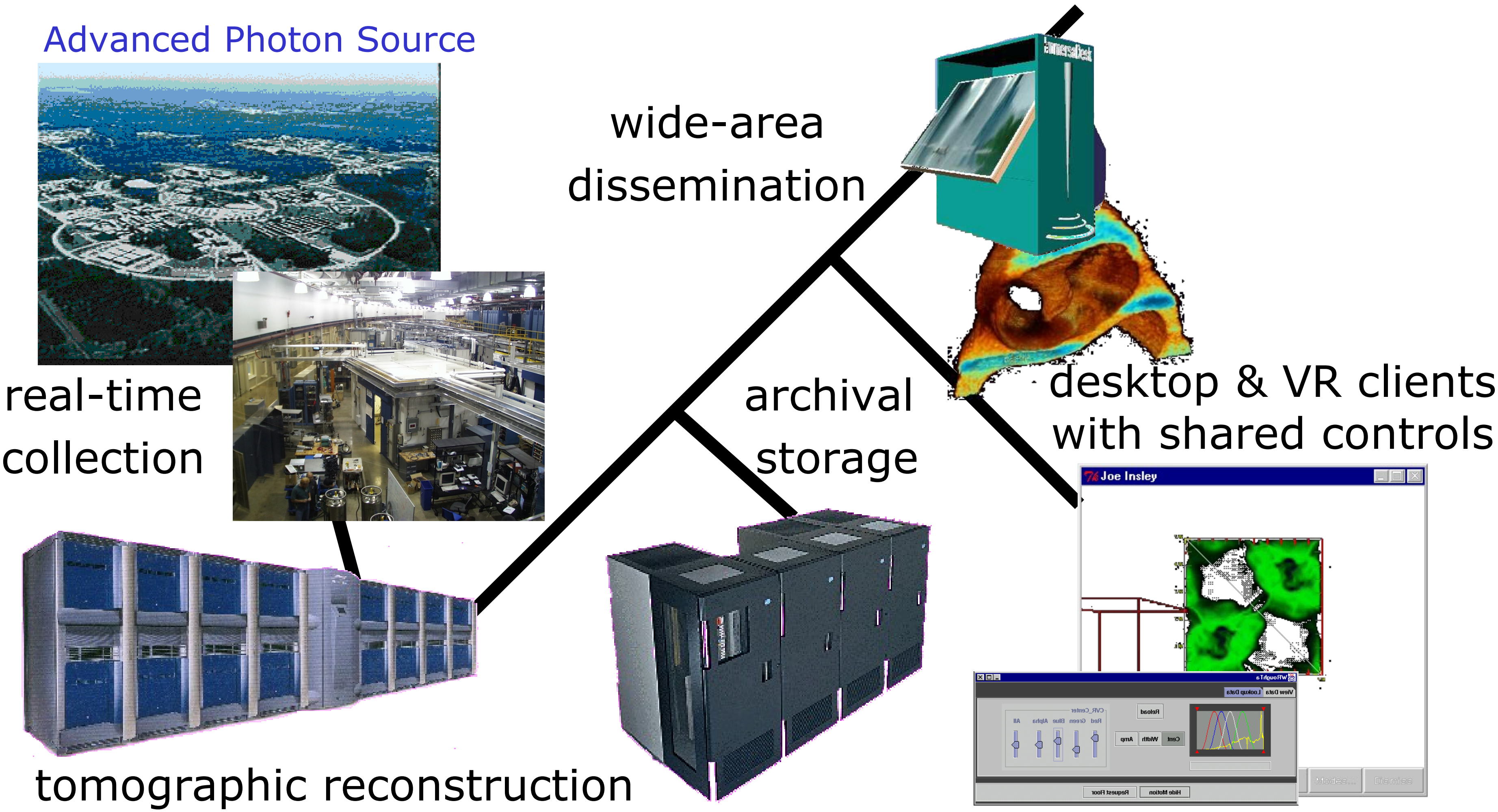
Access Grid
& Collaboration Technologies

Connecting People and Applications via the Grid

inSORS®

ACCESS GRID

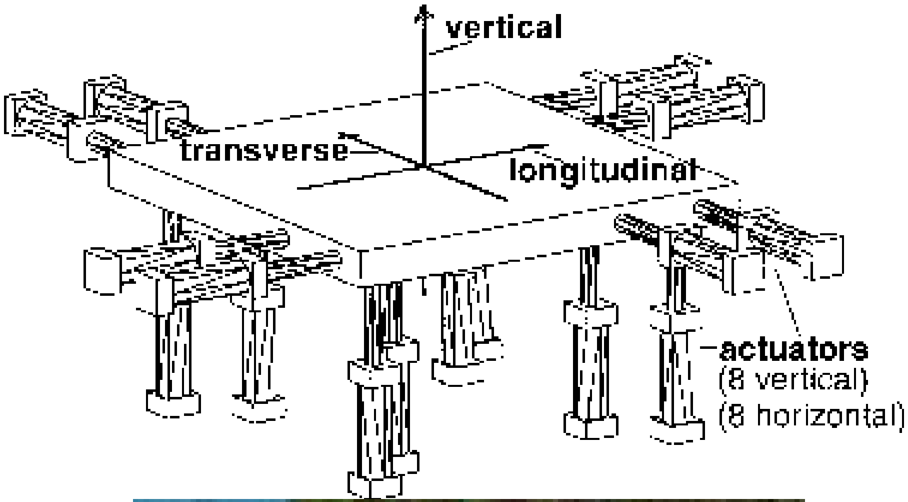
Online Access to Scientific Instruments

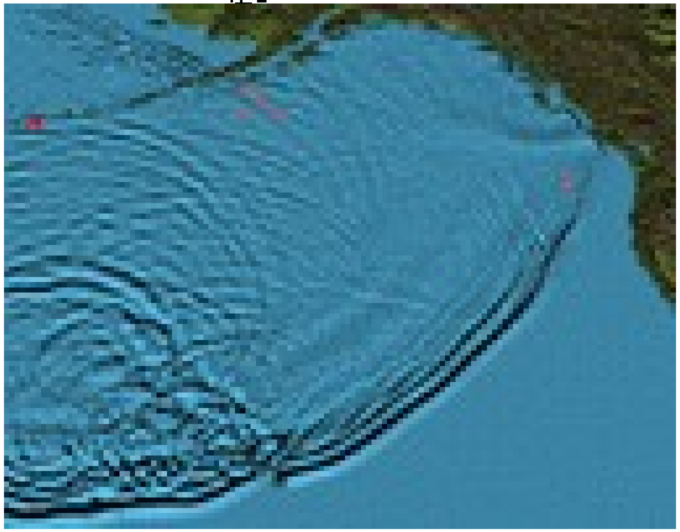


DOE X-ray grand challenge: ANL, USC/ISI, NIST, U.Chicago

Network for Earthquake Engineering Simulation

- NEESgrid: US national infrastructure to couple earthquake engineers with experimental facilities, databases, computers, & each other
- On-demand access to experiments, data streams, computing, archives, collaboration





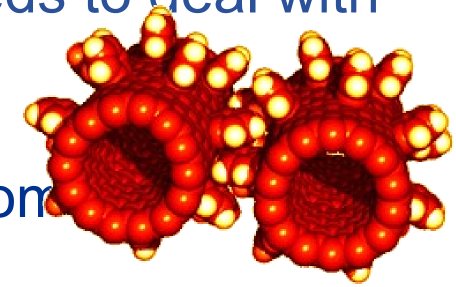
NEESgrid: Argonne, Michigan, NCSA, UIUC, USC

Home Computers Evaluate AIDS Drugs

- Community =
  - 1000s of home computer users
  - Philanthropic computing vendor (Entropia)
  - Research group (Scripps)
- Common goal= advance AIDS research

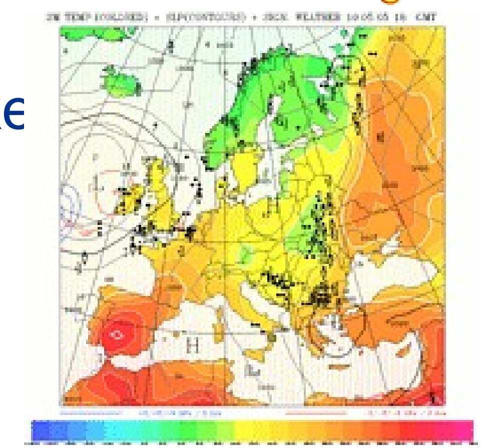
Computing intensive science

- Science is becoming increasingly digital and needs to deal with increasing amounts of data



Simulations get ever more detailed

  - Nanotechnology – design of new materials from the molecular scale
  - Modelling and predicting complex systems (weather forecasting, river floods, earthquake



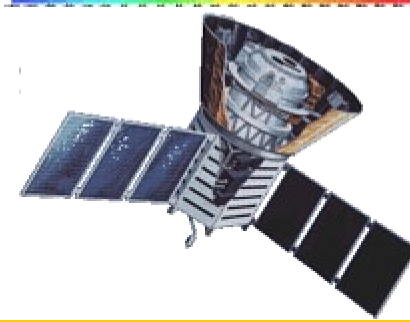
  - Decoding the human genome

- Experimental Science uses ever more sophisticated sensors to make precise measurements

  →Need high statistics

  →Huge amounts of data

  →Serves user communities around the world

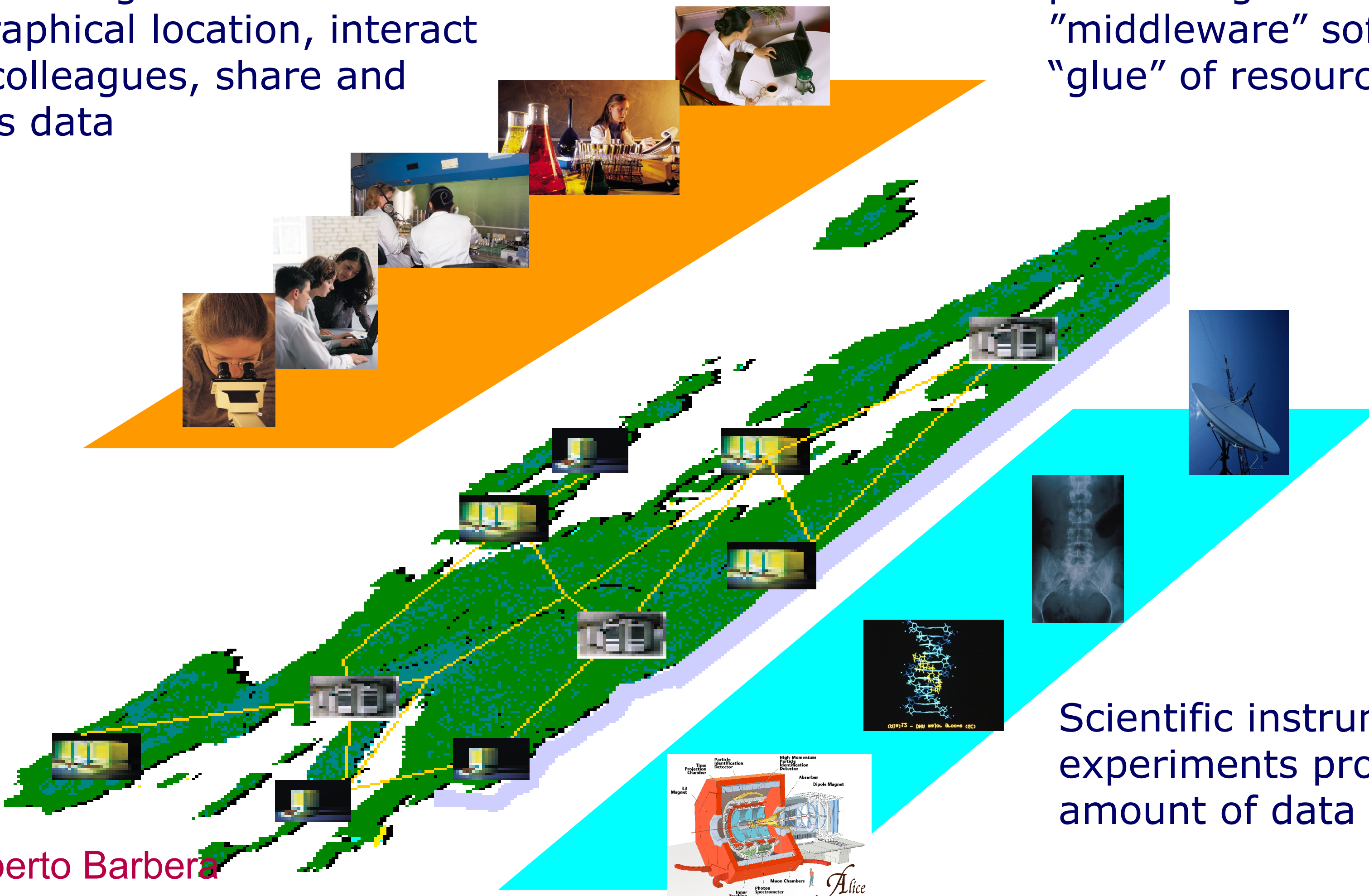


by Roberto Barbera

The Grid Vision

Researchers perform their activities regardless geographical location, interact with colleagues, share and access data

The Grid: networked data processing centres and ”middleware” software as the “glue” of resources.

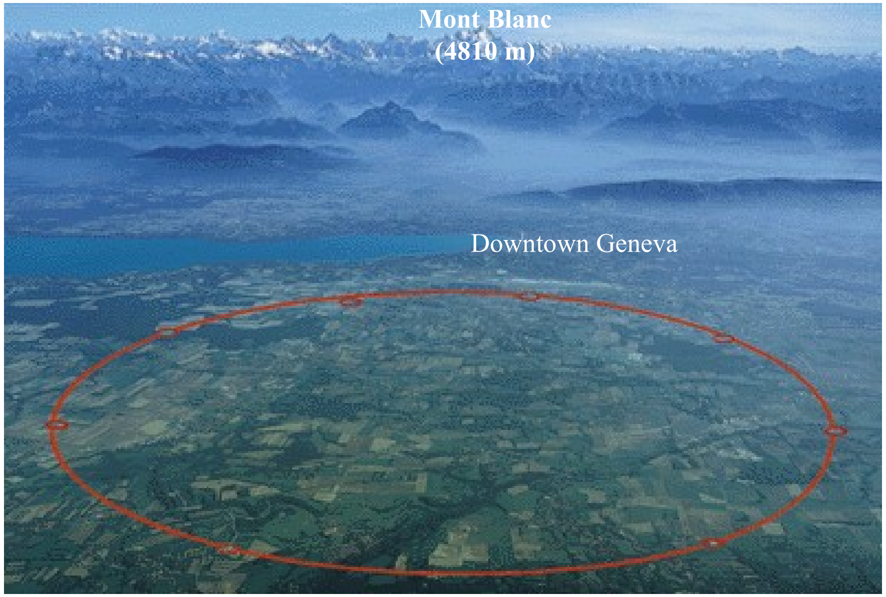


Scientific instruments and experiments provide huge amount of data

by Roberto Barbera

A good example: Particle Physics

- Large amount of data produced in a few places: CERN, FNAL, KEK…
- Large worldwide organized collaborations (i.e. LHC CERN experiments) of computer-savvy scientists
- Computing and data management resources distributed world-wide owned and managed by many different entities
- Large Hadron Collider (LHC) at CERN in Geneva Switzerland:
  - One of the most powerful instruments ever built to investigate matter



Data Grids for High Energy Physics

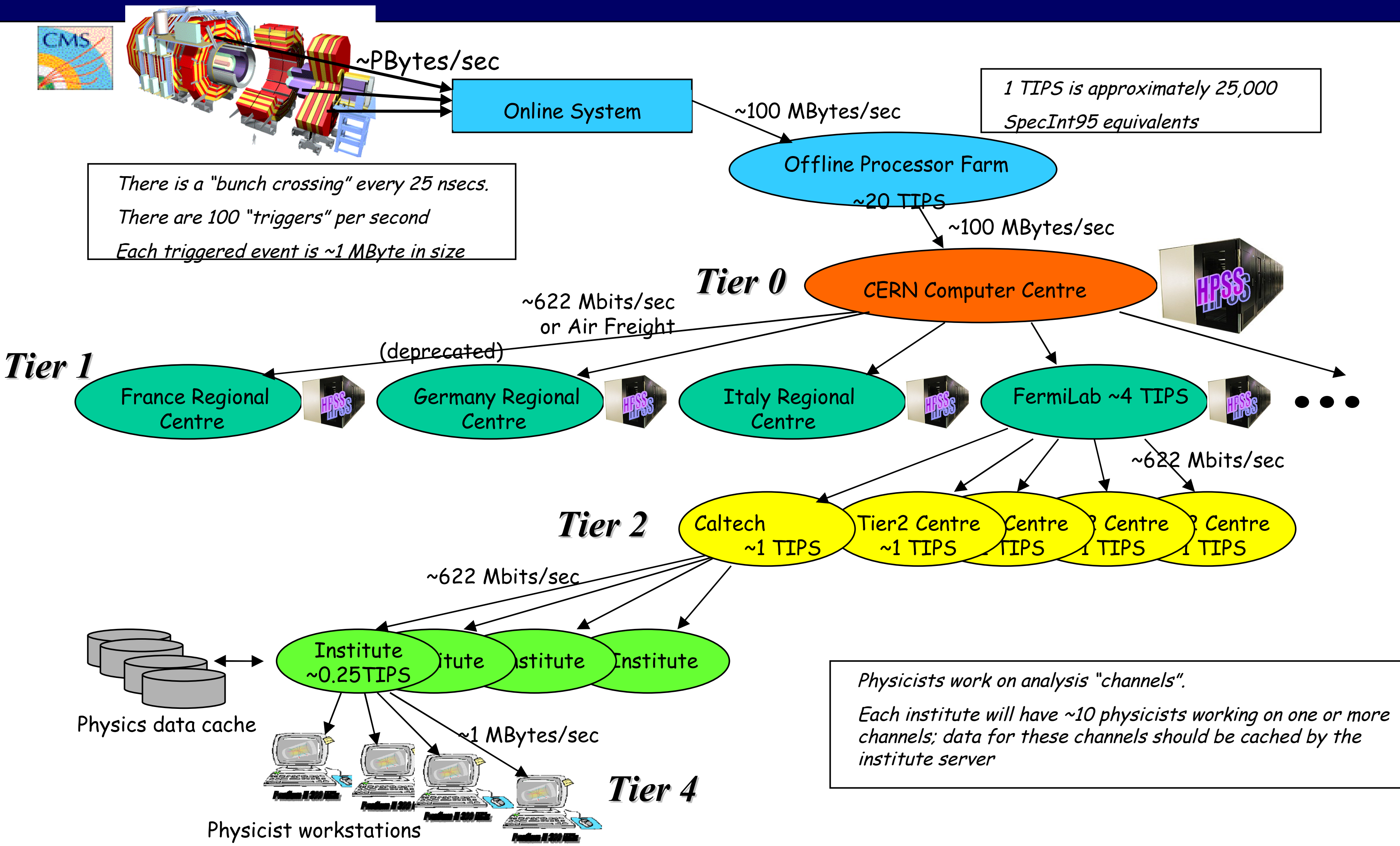


Image courtesy Harvey Newman, Caltech

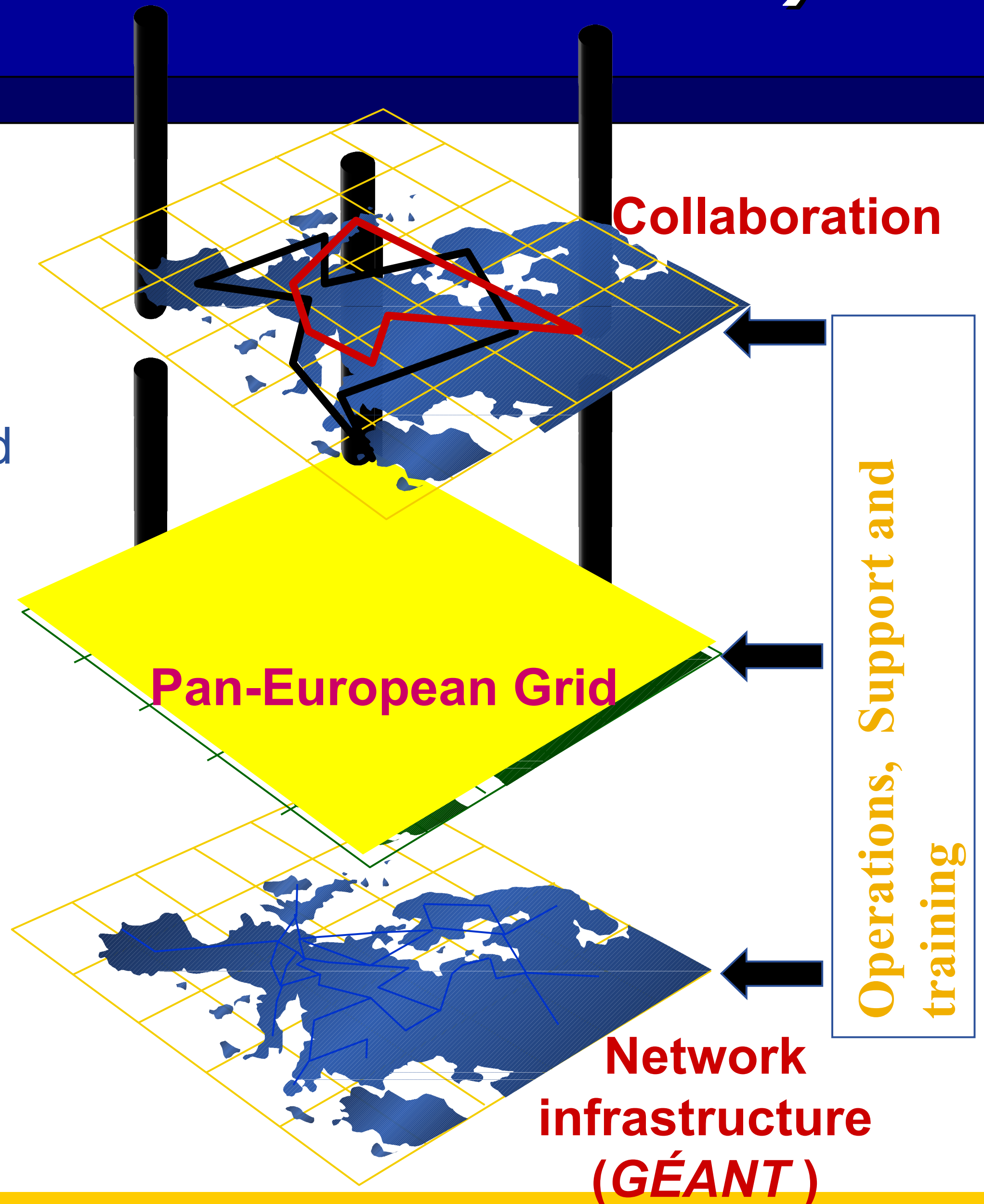
Outline

- Introduction to Grid Computing
  - Scientific environments and projects, big challanges
  - The EGEE Grid
- Grid Architecture
  - Some Definitions
  - The Programming Problem
- The Globus Toolkit™
  - Security
  - Resource Management
  - Information Services
  - Data Management

EGEE (Enabling Grids for E-sciencE)

Build a large-scale production grid service to:

- Underpin European science and technology
- Link with and build on national, regional and international initiatives
- Foster international cooperation both in the creation and the use of the e-infrastructure



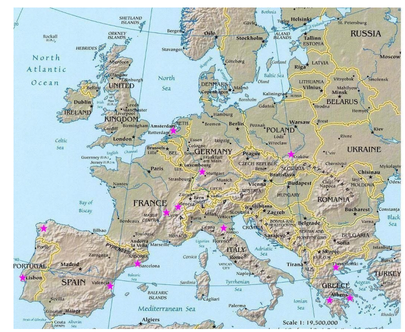
The largest e-Infrastructure: EGEE

- Objectives
  - consistent, robust and secure service grid infrastructure
  - improving and maintaining the middleware
  - attracting new resources and users from industry as well as science

- Structure
  - 71 leading institutions in 27 countries, federated in regional Grids
  - leveraging national and regional grid activities worldwide
  - funded by the EU with ~32 M Euros for first 2 years starting 1st April 2004

EGEE services

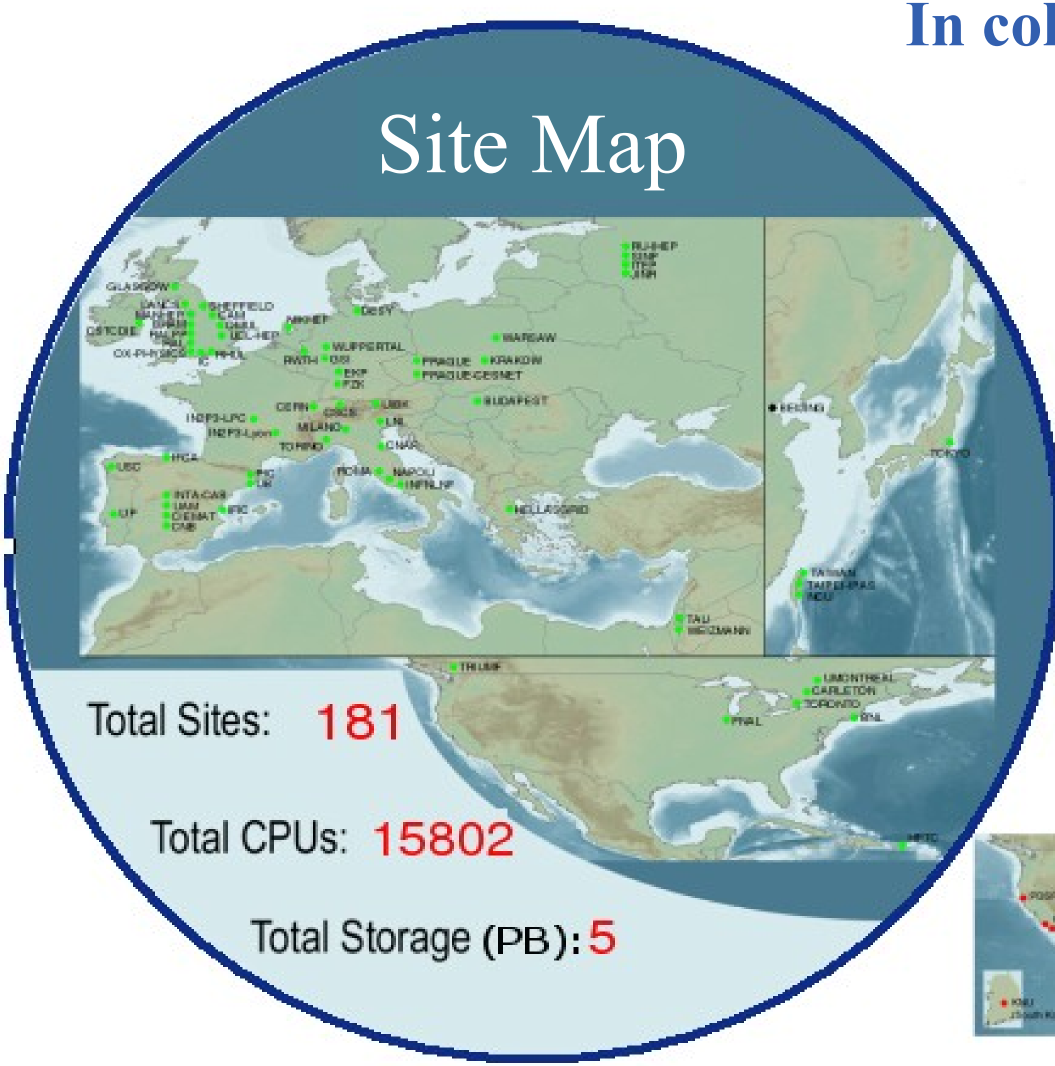
- Production service
  - Based on the LCG-2 service
  - With new resource centres and new applications encouraged to participate
  - Stable, well-supported infrastructure, running only well-tested and reliable middleware
- Pre-production service (14 sites)
  - Run in parallel with the production service
  - Access to new versions of the middleware
  - Applications test-bed



- GILDA testbed
  - https://gilda.ct.infn.it/testbed.html
  - Complete suite of Grid elements and applications
    - Testbed, CA, VO, monitoring
  - Everyone can register and use GILDA for training and testing

EGEE Infrastructure

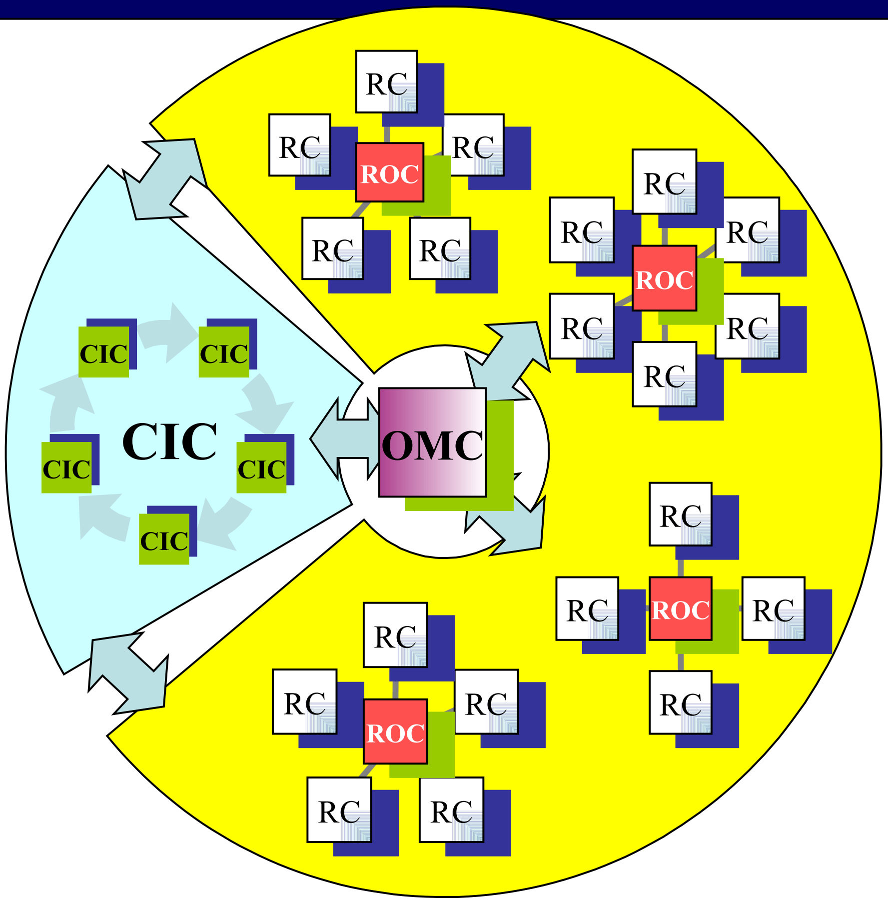
In collaboration with LCG



NorduGrid

Grid3/OSG

Grid Operations

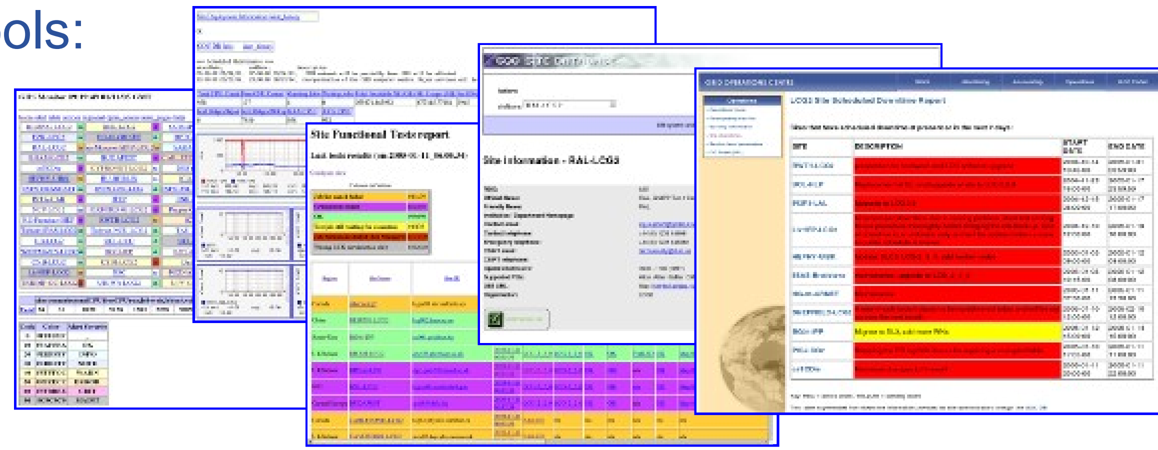


RC = Resource Centre
ROC = Regional Operations Centre
CIC = Core Infrastructure Centre
OMC = Operations Management Centre

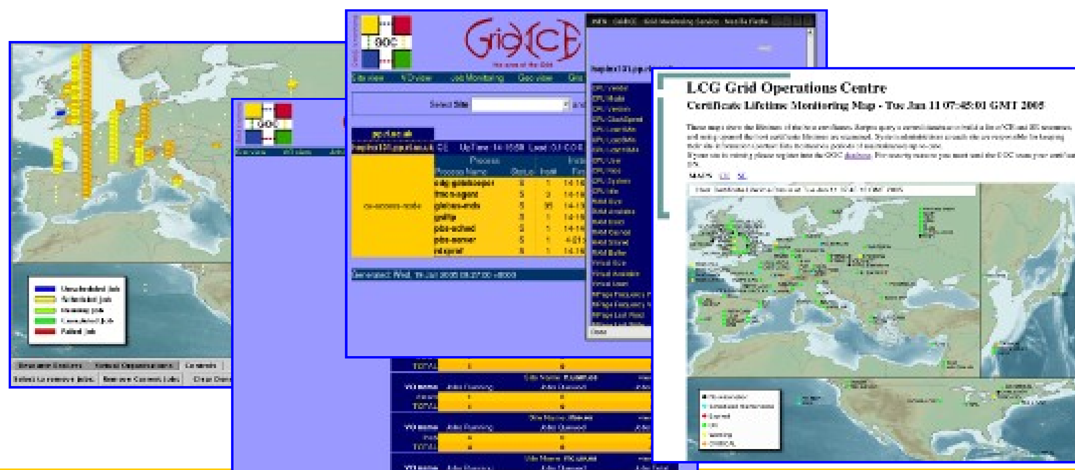
- The *grid* is flat, but
- *Hierarchy* of responsibility
  - Essential to scale the operation
- CICs act as a single Operations Centre
  - Operational oversight ***(grid operator)*** responsibility
  - rotates weekly between CICs
  - Report problems to ROC/RC
  - ROC is *responsible* for ensuring problem is resolved
  - ROC oversees regional RCs
- ROCs responsible for organising the operations in a region
  - Coordinate deployment of middleware, etc
- CERN coordinates sites not associated with a ROC

Grid monitoring (http://www.ukiroc.eu/content/view/115/235)

- Operation of Production Service: real-time display of grid operations
- Accounting Information
- Selection of Monitoring tools:
  - GIIS Monitor + Monitor Graphs
  - Sites Functional Tests
  - GOC Data Base
  - Scheduled Downtimes

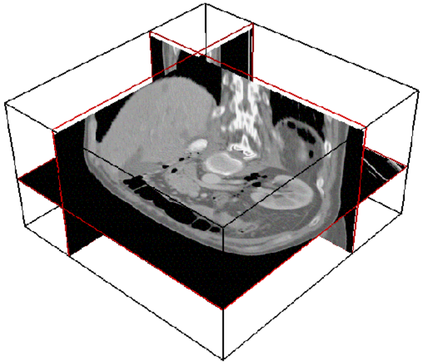


  - Live Job Monitor
  - GridIce – VO + Fabric View
  - Certificate Lifetime Monitor



EGEE pilot applications

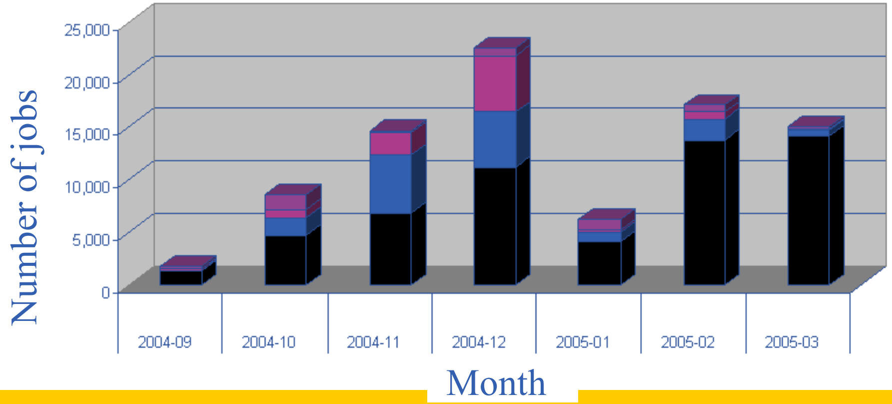
- High-Energy Physics (HEP)
  - Provides computing infrastructure (LCG)
  - Challenging:
    - thousands of processors world-wide
    - generating petabytes of data
    - ‘chaotic’ use of grid with individual user analysis (thousands of users interactively operating within experiment VOs)



- Biomedical Applications
  - Similar computing and data storage requirements
  - Major additional challenge: security & privacy

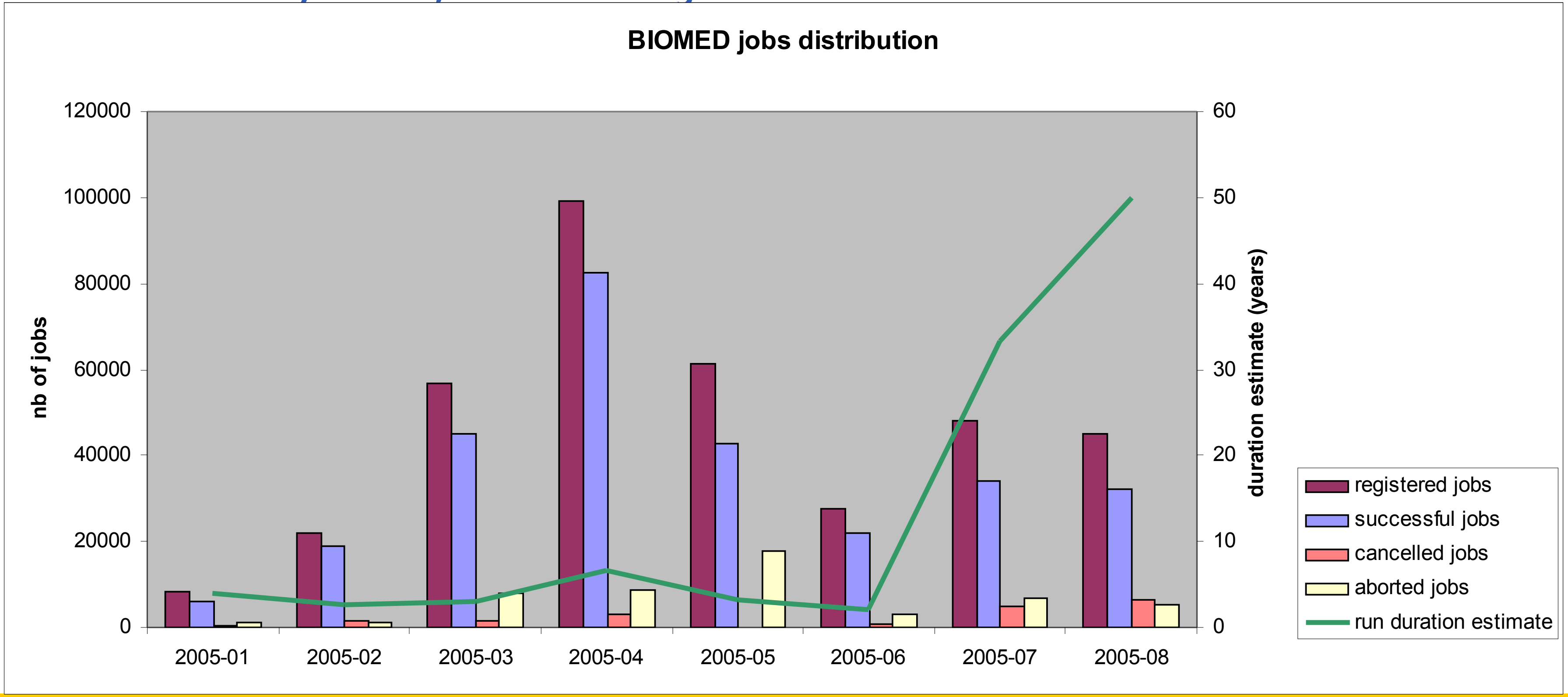
BioMed Overview

- Infrastructure
  - ~3.000 CPUs
  - ~12 TB of disk
  - in 9 countries
- >50 users in 7 countries working with 12 applications
- 18 research labs



Biomed Virtual Organisation

- ~ 70 users, 9 countries
- > 12 Applications (medical image processing, bioinformatics)
- ~3000 CPUs, ~12 TB disk space
- ~100 CPU years, ~ 500K jobs last 6 months

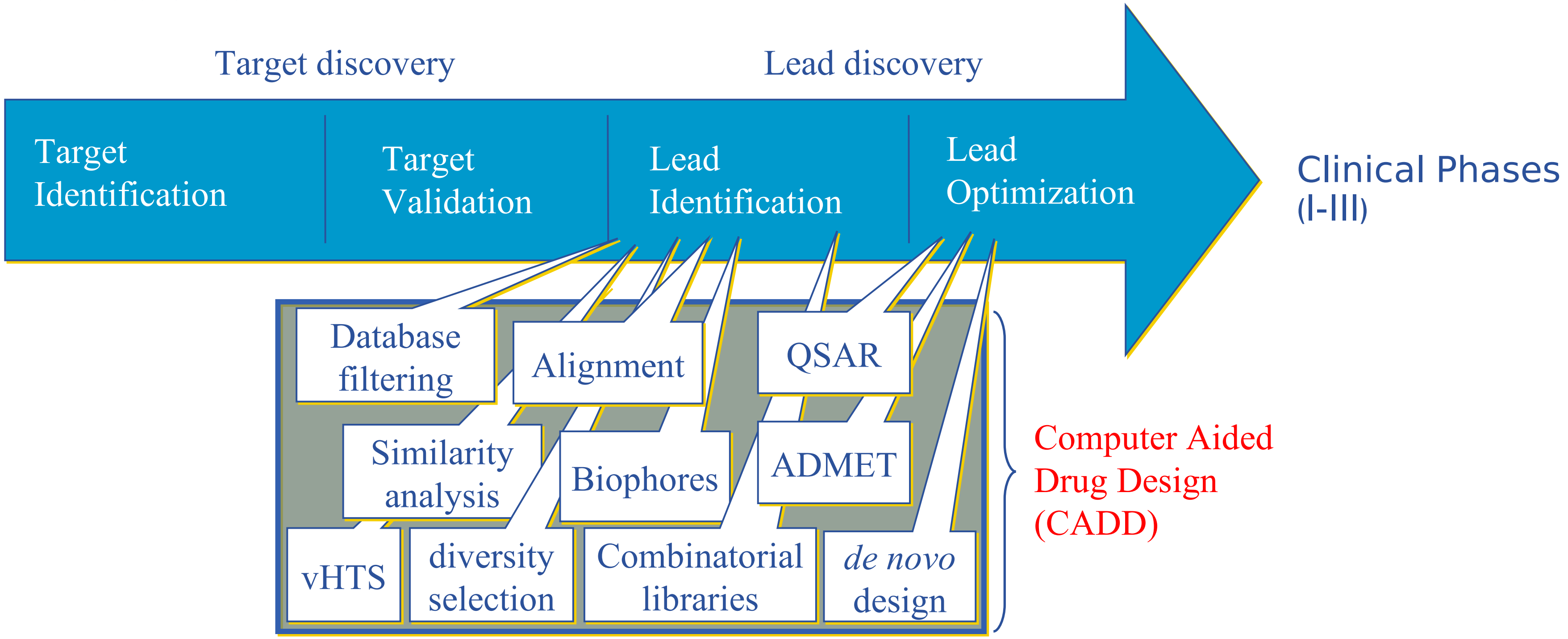


Bioinformatics

- GPS@: Grid Protein Sequence Analysis
  - **Gridified version of NPSA web portal**
    - Offering proteins databases and sequence analysis algorithms to the bioinformaticians (3000 hits per day)
    - Need for large databases and big number of short jobs
  - **Objective**: increased computing power
  - **Status**: 9 bioinformatic softwares gridified
  - **Grid added value**: open to a wider community with larger bioinformatic computations
- xmipp_MLrefine
  - **3D structure analysis of macromolecules**
    - From (very noisy) electron microscopy images
    - Maximum likelihood approach to find the optimal model
  - **Objective**: study molecule interaction and chem. properties
  - **Status**: algorithm being optimised and ported to 3D
  - **Grid added value**: parallel computation on different resources of independent jobs

Drug Discovery

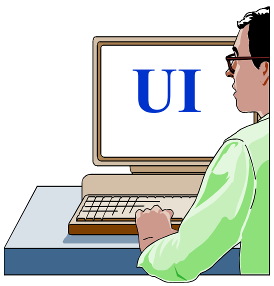
- Demonstrate the relevance and the impact of the grid approach to address Drug Discovery for neglected diseases



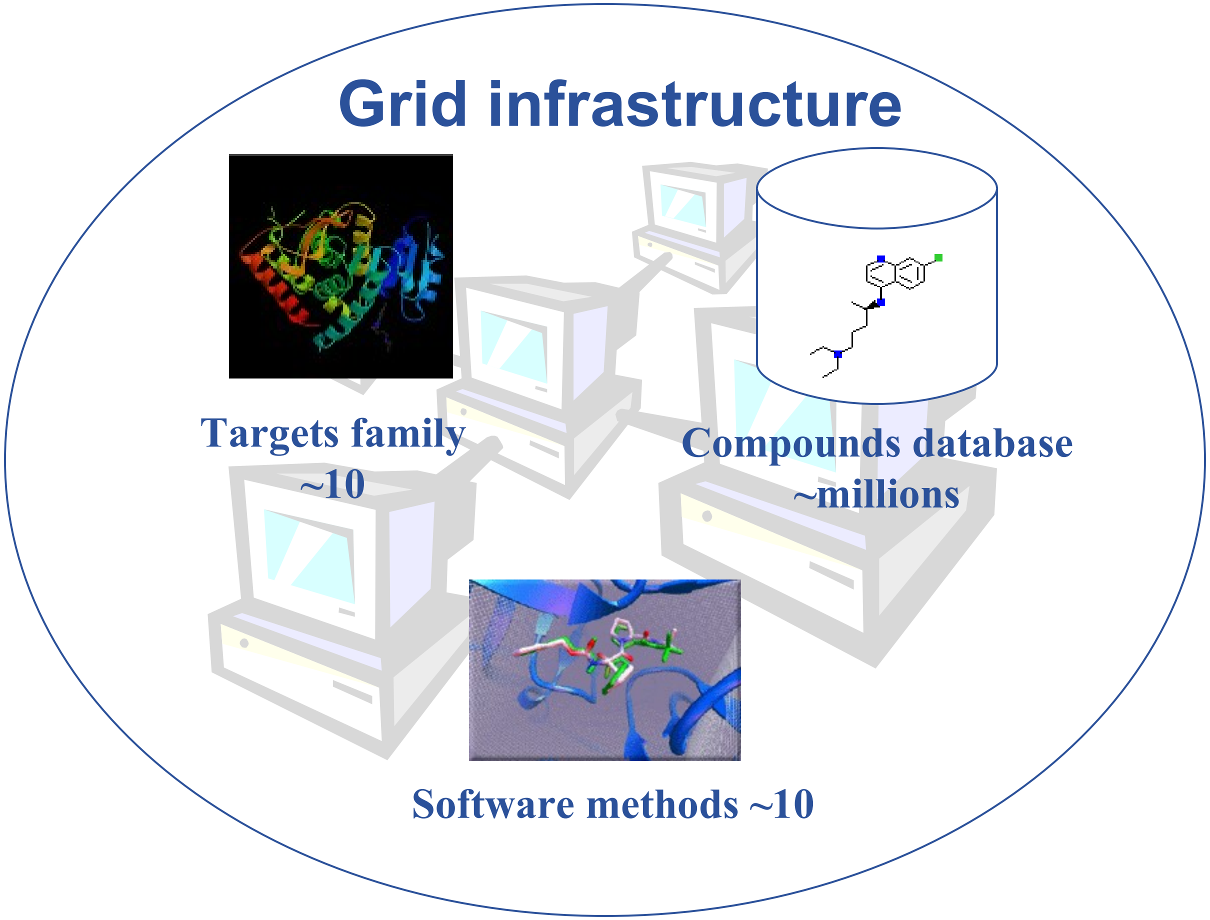
Duration: 12 - 15 years, Costs: 500 - 800 million US $

Docking platform components

- Predict how small molecules, such as substrates or drug candidates, bind to a receptor of known 3D structure

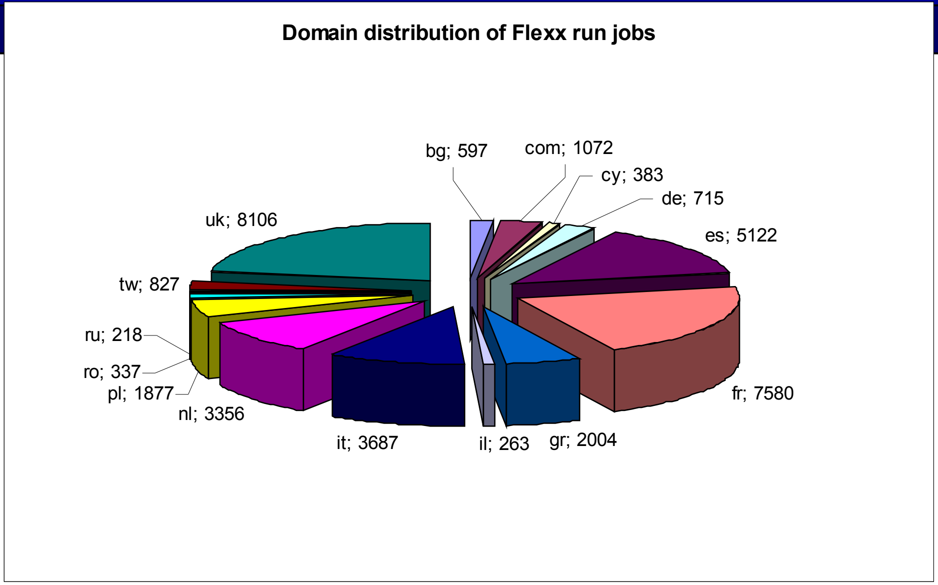


Parameter / scoring settings

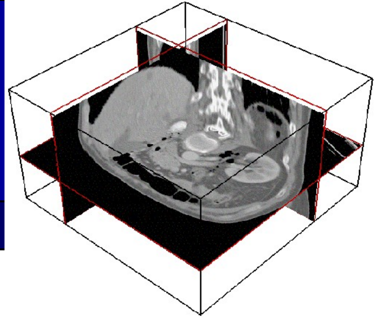


First biomedical data challenge: World-wide In Silico Docking On Malaria (WISDOM)

- Significant biological parameters
  - two different molecular docking applications (Autodock and FlexX)
  - about one million virtual ligands selected
  - target proteins from the parasite responsible for malaria
- Significant numbers
  - Total of about 46 million ligands docked in 6 weeks
  - 1TB of data produced
  - Up 1000 computers in 15 countries used simultaneously corresponding to about 80 CPU years



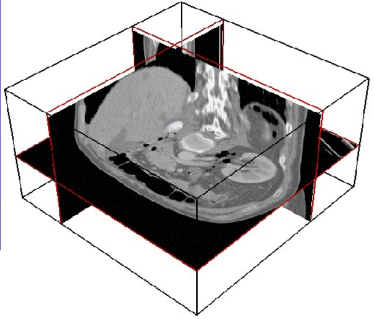
Medical imaging



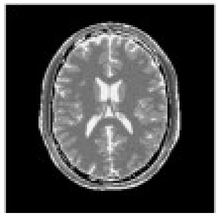
- GATE
  - **Radiotherapy planning**
    - Improvement of precision by Monte Carlo simulation
    - Processing of DICOM medical images
  - **Objective**: very short computation time compatible with clinical practice
  - **Status**: development and performance testing
  - **Grid Added Value**: parallelisation reduces computing time
- CDSS

  - **Clinical Decision Support System**
    - Assembling knowledge databases
    - Using image classification engines
  - **Objective:** access to knowledge databases from hospitals
  - **Status**: from development to deployment, some medical end users
  - **Grid Added Value**: ubiquitous, managed access to distributed databases and engines

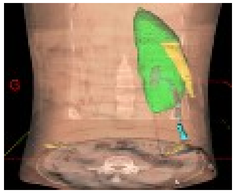
Medical imaging



- SiMRI3D
  - **3D Magnetic Resonance Image Simulator**
    - MRI physics simulation, parallel implementation
    - Very compute intensive
  - **Objective**: offering an image simulator service to the research community
  - **Status**: parallelised and now running on EGEE resources
  - **Grid Added Value**: enables simulation of high-res images



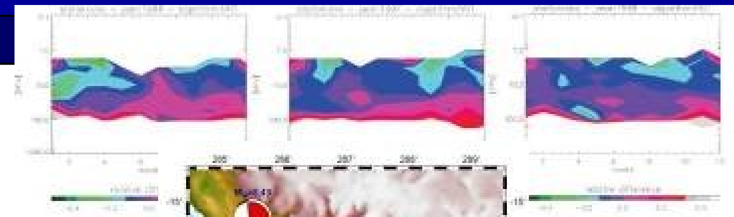
- gPTM3D
  - **Interactive tool to segment and analyse medical images**
    - A non gridified version is distributed in several hospitals
    - Need for very fast scheduling of interactive tasks
  - **Objectives**: shorten computation time using the grid
    - Interactive reconstruction time: < 2min and scalable
  - **Status**: development of the gridified version being finalized
  - **Grid Added Value**: permanent availability of resources

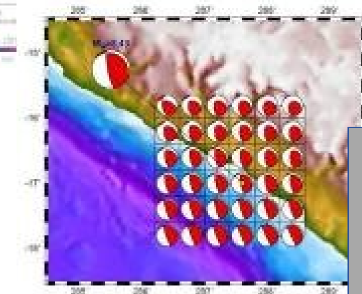


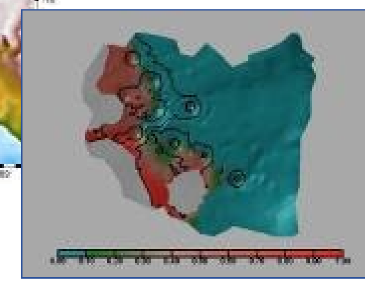
Generic Applications

- EGEE Generic Applications Advisory Panel (EGAAP)
  - UNIQUE entry point for “external” applications
  - Reviews proposals and make recommendations to EGEE management
    - Deals with “scientific” aspects, not with technical details
    - Generic Applications group in charge of introducing selected applications to the EGEE infrastructure
  - 8 applications selected so far:
    - Earth sciences (earth observation, geophysics, hydrology, seismology)
    - MAGIC (astrophysics)
    - Computational Chemistry
    - PLANCK (astrophysics and cosmology)
    - Drug Discovery
    - E-GRID (e-finance and e-business)
    - FUSION
    - ArchaeoGrid
    - GRACE (grid search engine)

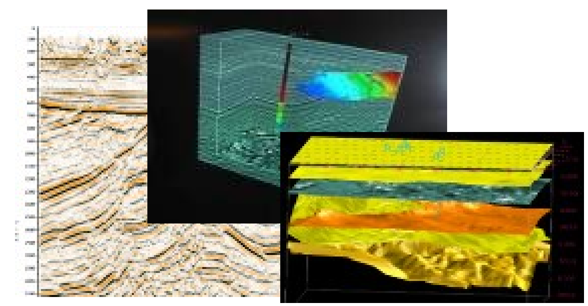
Earth sciences applications





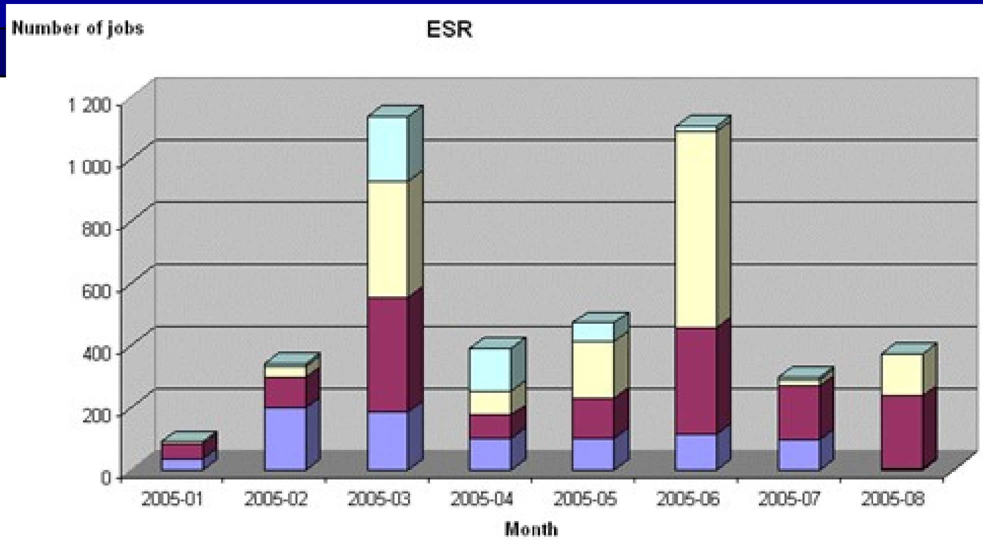


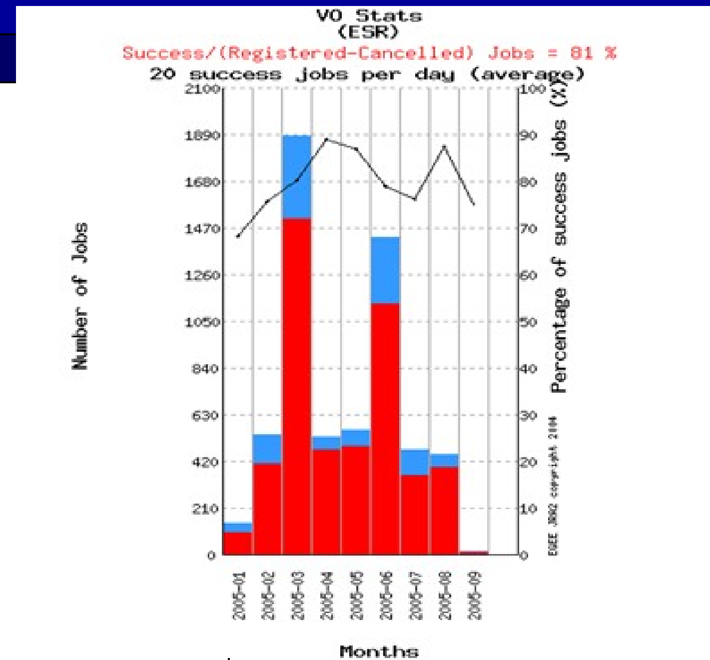
- Earth Observations by Satellite
  - Ozone profiles
- Solid Earth Physics
  - Fast Determination of mechanisms of important earthquakes
- Hydrology
  - Management of water resources in Mediterranean area (SWIMED)
- Geology
  - Geocluster: R&D initiative of the Compagnie Générale de Géophysique



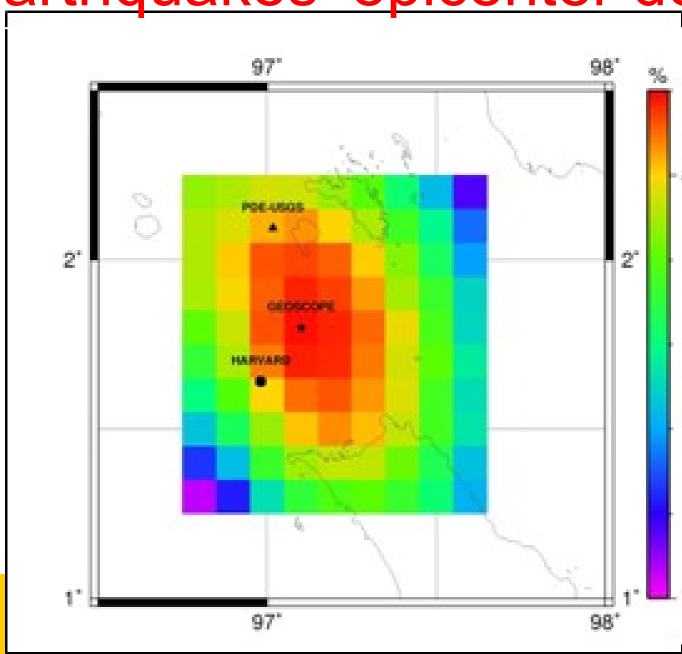
- ➢ A large variety of applications ported on EGEE which incites new users
- ➢ Interactive Collaboration of the teams around a project

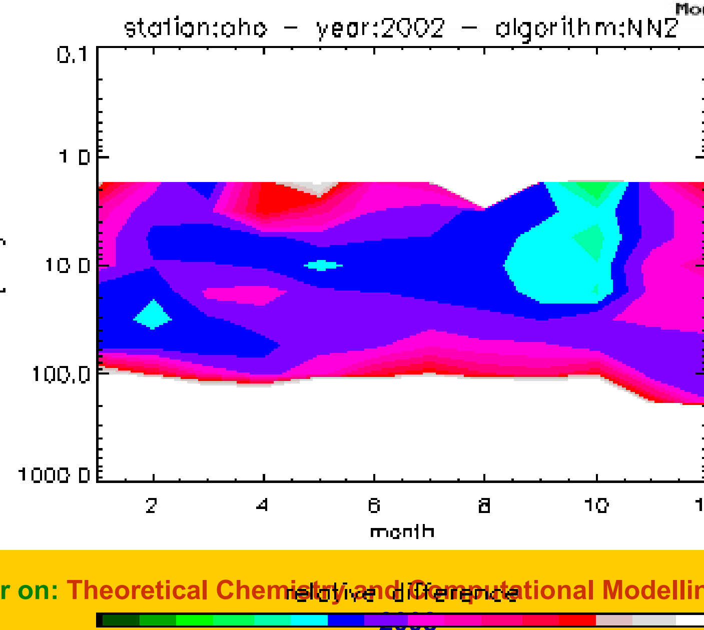
Generic Applications' use of EGEE





Earthquakes' epicenter determination

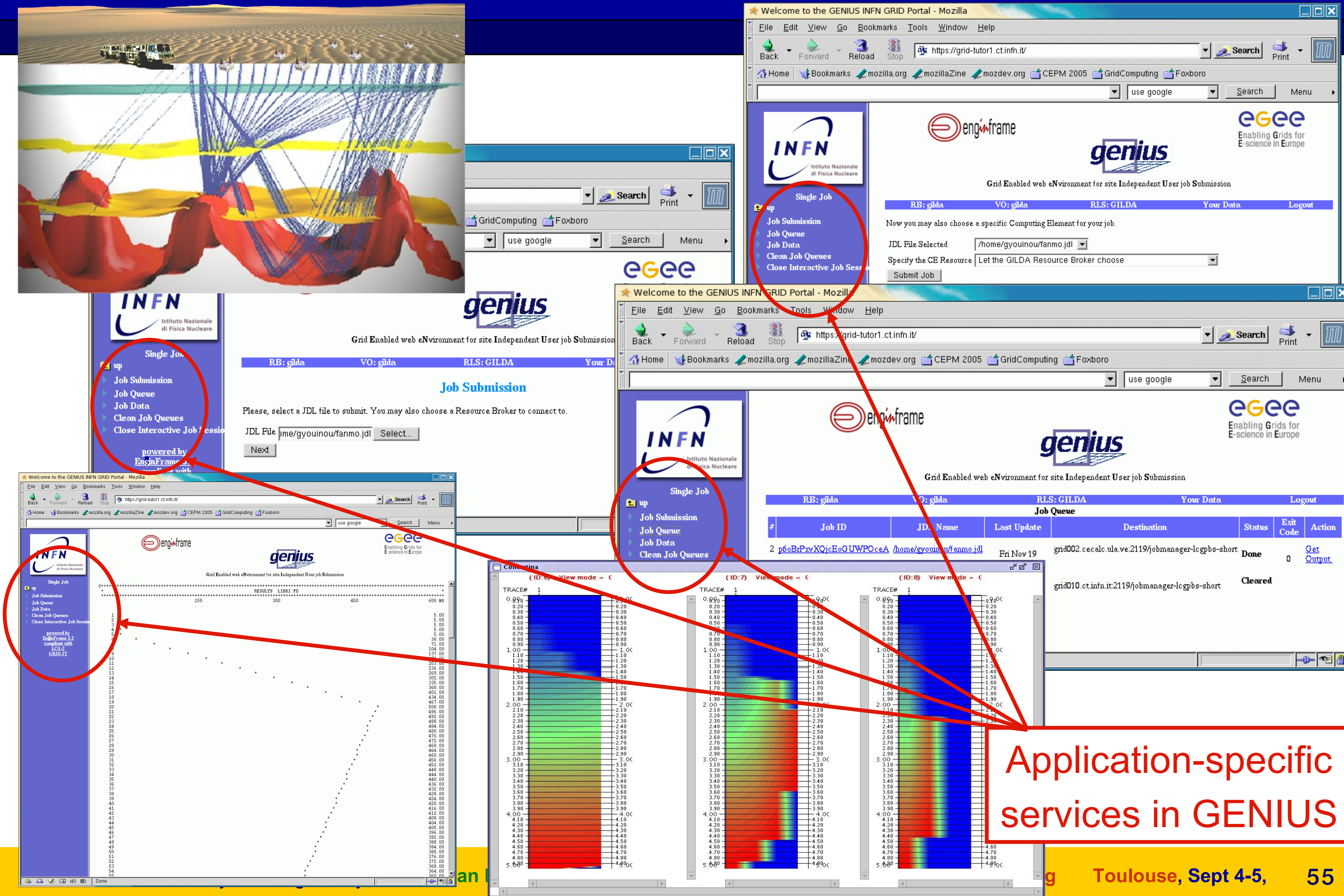




Ozone maps

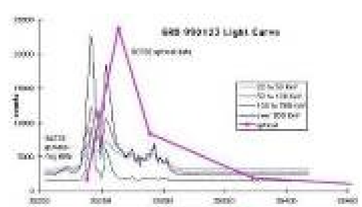
Climate

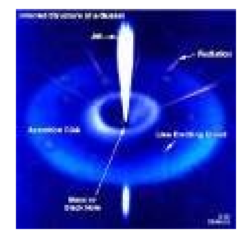
Earth Science (industrial): EGEODE example

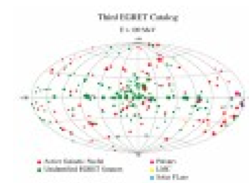


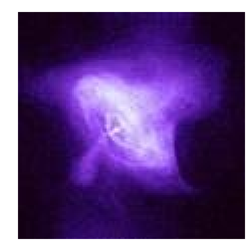
MAGIC

- Ground based Air Cerenkov Telescope 17 m diameter
- Physics Goals:
  - Origin of VHE Gamma rays
  - Active Galactic Nuclei
  - Supernova Remnants
  - Unidentified EGRET sources
  - Gamma Ray Burst
- MAGIC II will come 2007
- Grid added value
  - Enable “(e-)scientific” collaboration between partners
  - Enable the cooperation between different experiments
  - Enable the participation on Virtual Observatories

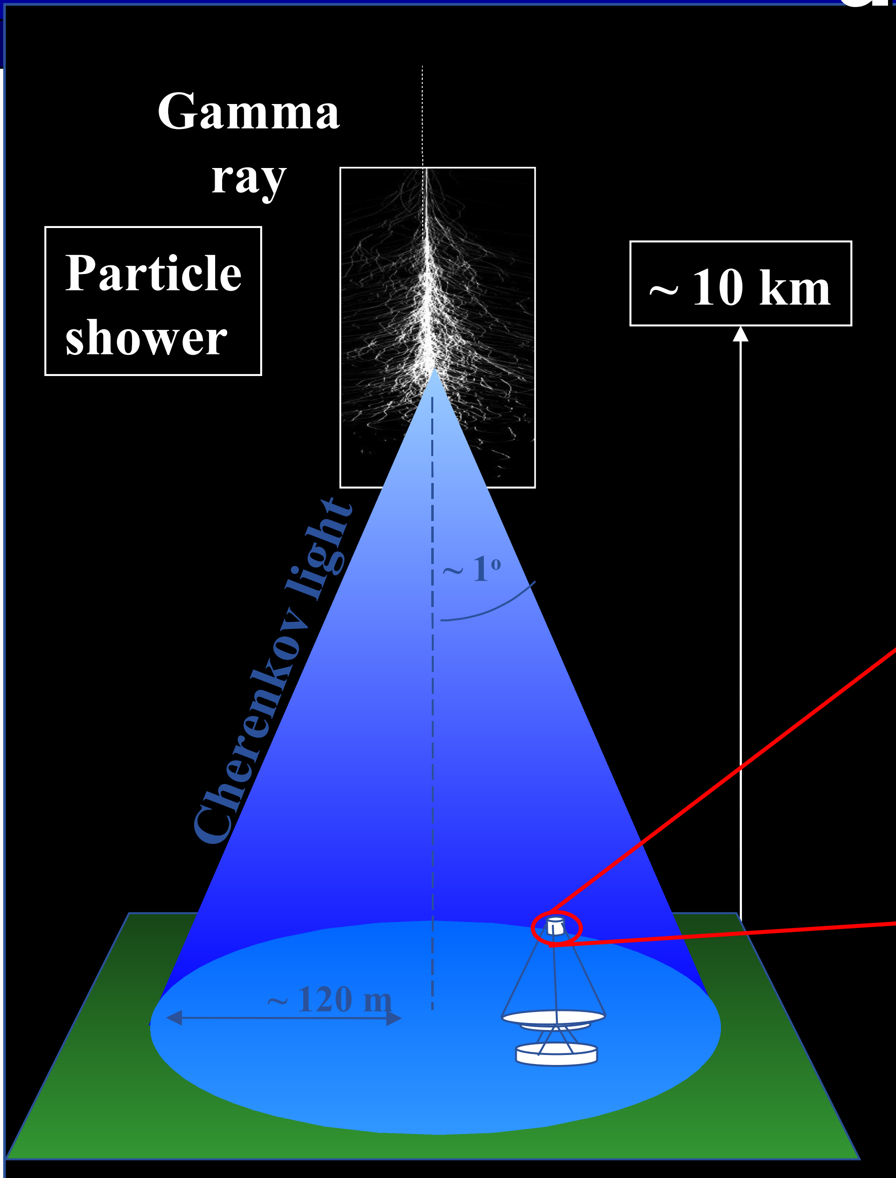




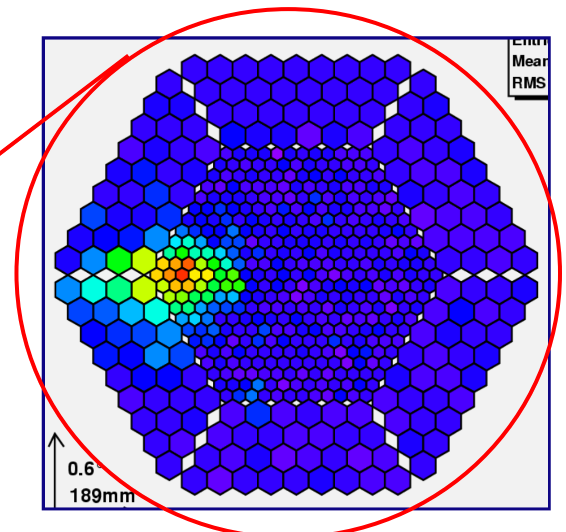




Ground based γ-ray astronomy



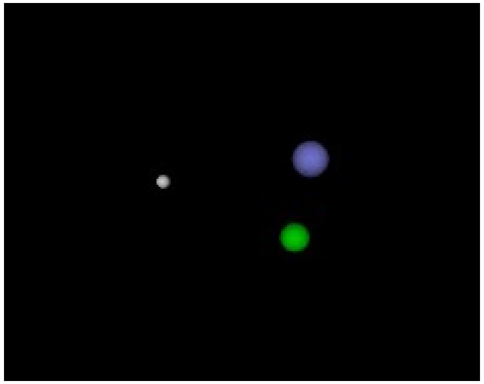
Cherenkov light Image of particle shower in telescope camera

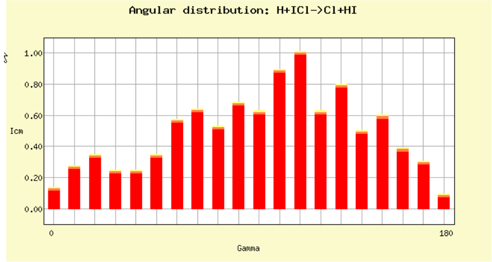


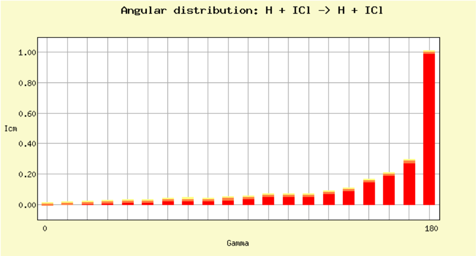
reconstruct:
  arrival direction, energy
reject hadron background

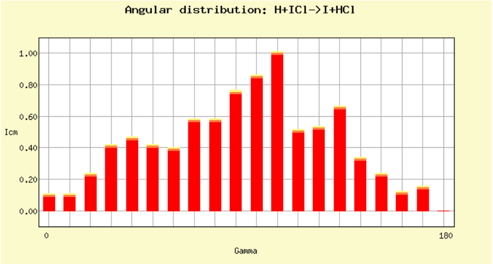
Computational Chemistry

- The Grid Enabled Molecular Simulator (GEMS)
  - Motivation:
    - Modern computer simulations of molecular systems produce an abundance of data, which could be reused several times by different researchers.
      - data must be catalogued and searchable
  - GEMS database and toolkit:
    - autonomous storage resources
    - metadata specification
    - automatic storage allocation and replication policies
    - interface for distributed computation







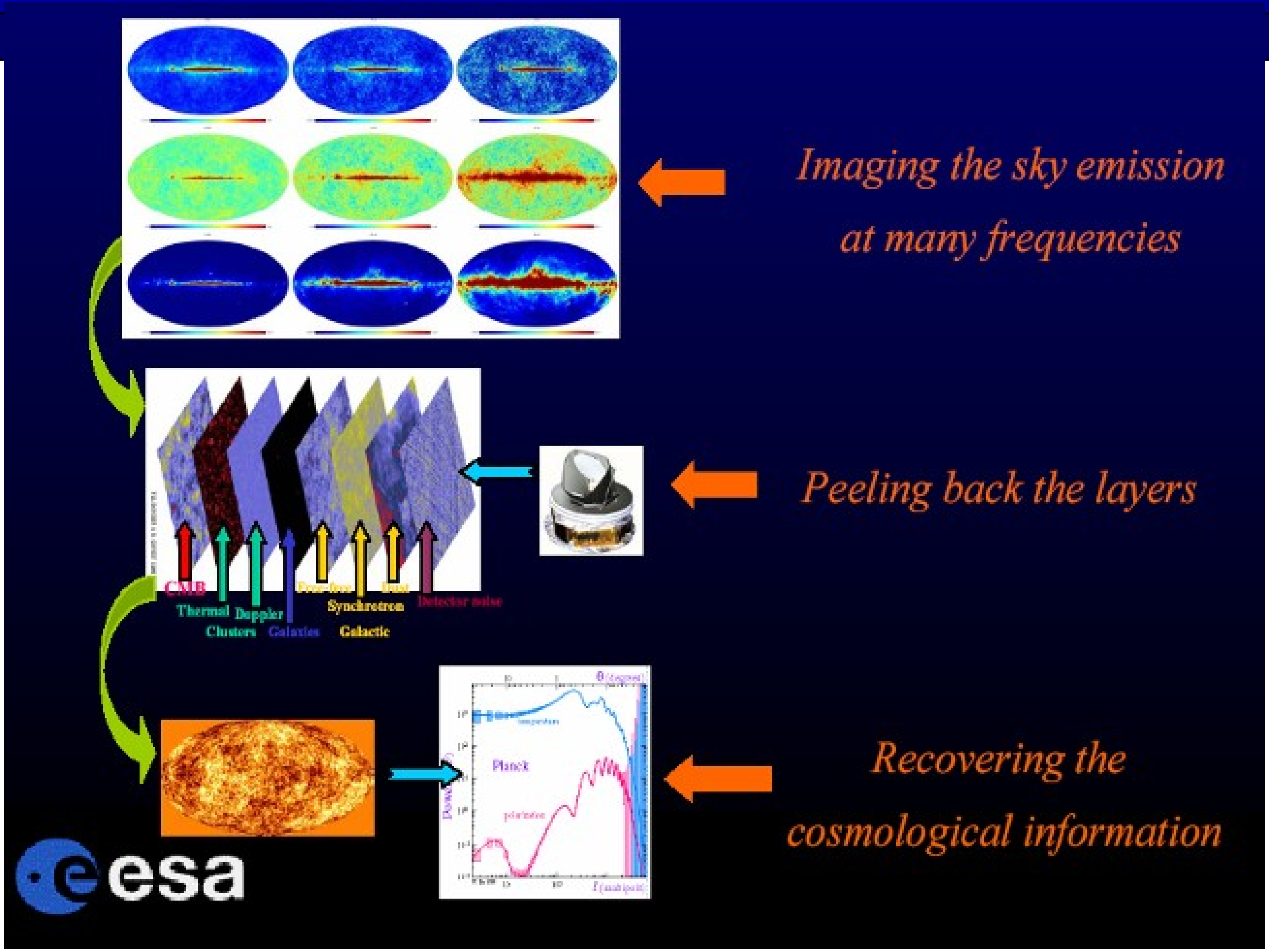


Planck

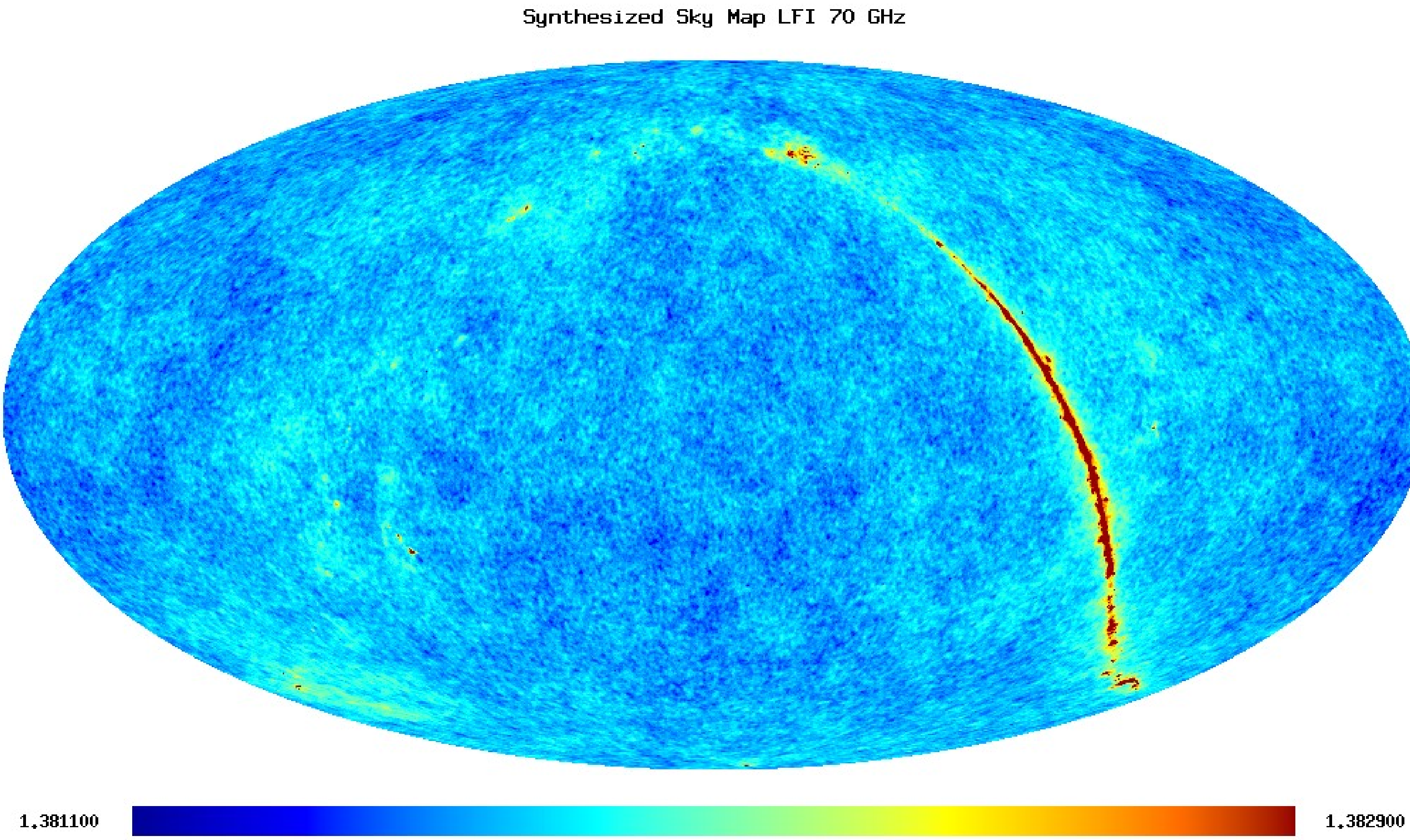
- On the Grid:
  > 12 time faster
  (only ~5% failures)

- Complex data structure
  - data handling important

- The Grid as
  – collaboration tool
  – common user-interface
  – flexible environment
  – new approach to data and S/W sharing



Planck first tests

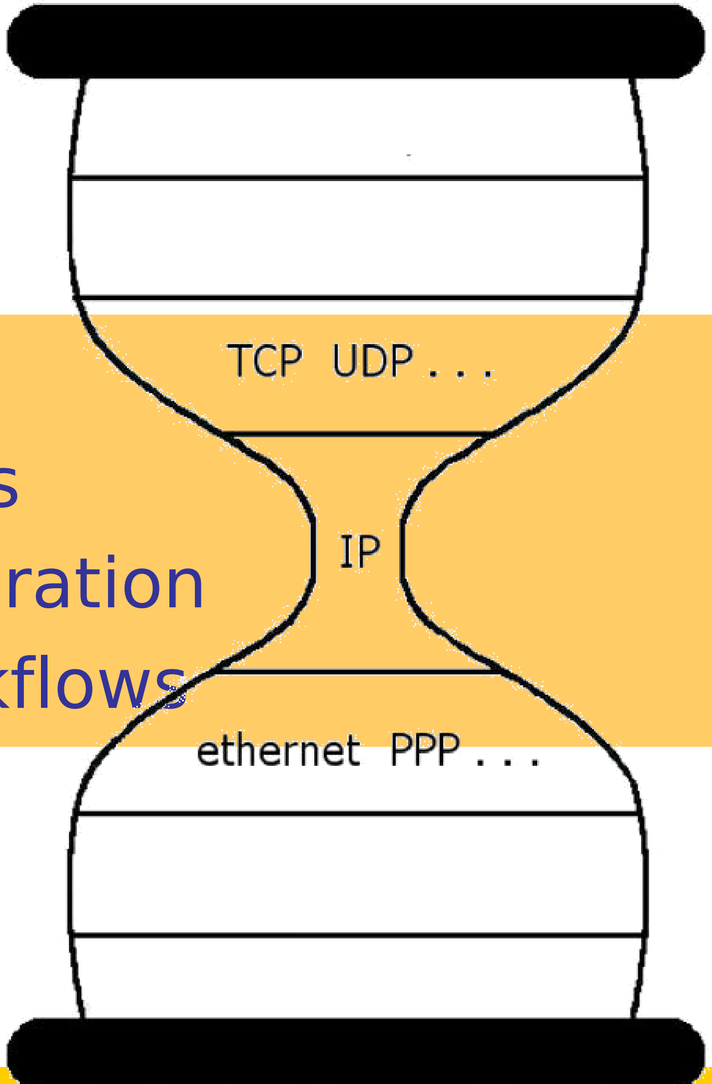


Outline

- Introduction to Grid Computing
  - Scientific environments and projects, big challenges
  - The EGEE Grid
- Grid Architecture
  - Some Definitions
  - The Programming Problem
- The Globus Toolkit™
  - Security
  - Resource Management
  - Information Services
  - Data Management

Grid Technology

- **Applications**:
  - Address complex problems
  - Provide community services
- **Software** enables:
  - On-demand access to services
  - Secure, reliable, dynamic federation
  - Definition & execution of workflows
  - Provide access to resources
  - Host robust services and content



Broader Context

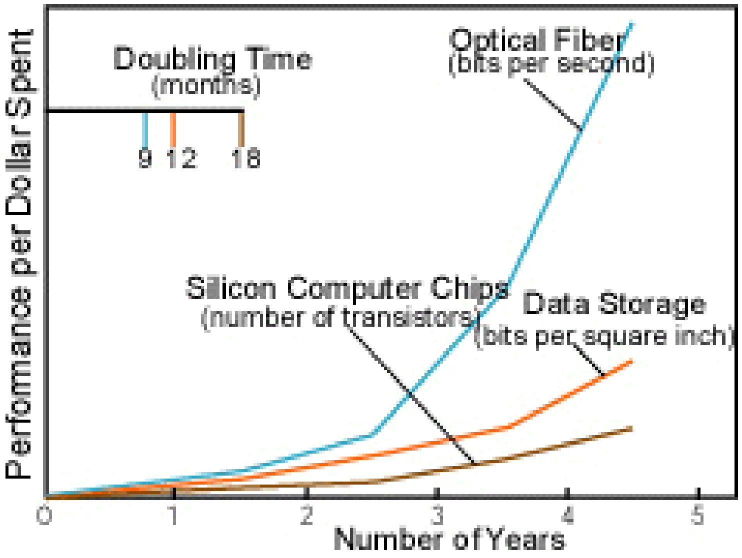
- "Grid Computing" has much in common with major industrial thrusts
  - Business-to-business, Peer-to-peer, Application Service Providers, Storage Service Providers, Distributed Computing, Internet Computing…
- Sharing issues not adequately addressed by existing technologies
  - Complicated requirements: "run program X at site Y subject to community policy P, providing access to data at Z according to policy Q"
  - High performance: unique demands of advanced & high-performance systems

Why Now?

- Moore's law improvements in computing produce highly functional endsystems
- The Internet and burgeoning wired and wireless provide universal connectivity
- Changing modes of working and problem solving emphasize teamwork, computation
- Network exponentials produce dramatic changes in geometry and geography

Network Exponentials

- Network vs. computer performance
  - Computer speed doubles every 18 months
  - Network speed doubles every 9 months
  - Difference = order of magnitude per 5 years
- 1986 to 2000
  - Computers: x 500
  - Networks: x 340,000
- 2001 to 2010
  - Computers: x 60



Moore's Law vs. storage improvements vs. optical improvements. Graph from **Scientific American** (Jan-2001) by Cleo Vilett, source Vined Khoslan, Kleiner, Caufield and Perkins.

The Globus Project™
Making Grid computing a reality

- Close collaboration with real Grid projects in science and industry
- Development and promotion of standard Grid protocols to enable interoperability and shared infrastructure
- Development and promotion of standard Grid software APIs and SDKs to enable portability and code sharing
- The Globus Toolkit™: Open source, reference software base for building grid infrastructure and applications
- Global Grid Forum: Development of standard protocols and APIs for Grid computing

Some Important Definitions

- Resource
- Network protocol
- Network enabled service
- Application Programmer Interface (API)
- Software Development Kit (SDK)
- Syntax
- Not discussed, but important: policies

Resource

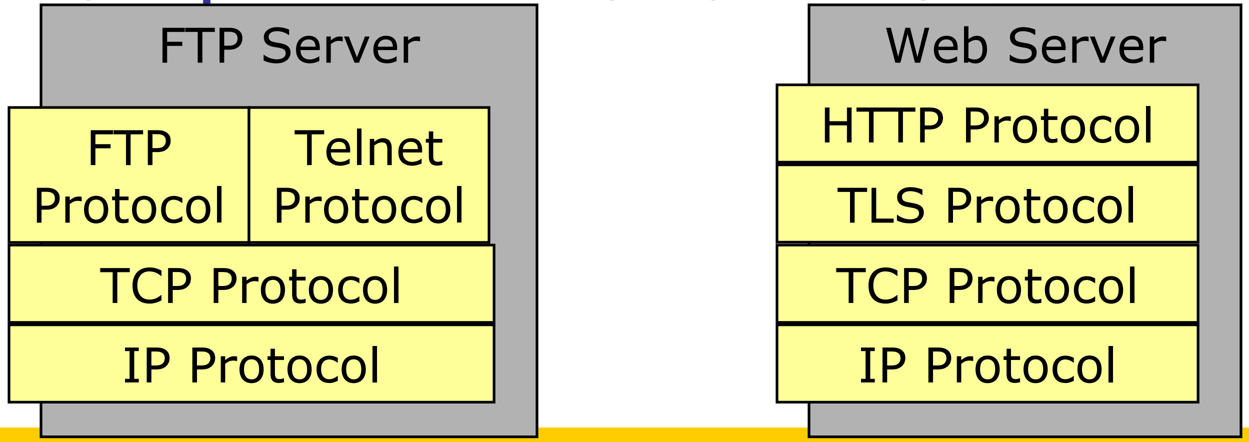
- An entity that is to be shared
  - E.g., computers, storage, data, software
- Does <u>not</u> have to be a physical entity
  - E.g., Condor pool, distributed file system, …
- Defined in terms of interfaces, not devices
  - E.g. scheduler such as LSF and PBS define a compute resource
  - Open/close/read/write define access to a distributed file system, e.g. NFS, AFS, DFS

Network Protocol

- A formal description of message formats and a set of rules for message exchange
  - Rules may define sequence of message exchanges
  - Protocol may define state-change in endpoint, e.g., file system state change
- Good protocols designed to do one thing
  - Protocols can be layered
- Examples of protocols
  - IP, TCP, TLS (was SSL), HTTP, Kerberos

Network Enabled Services

- Implementation of a protocol that defines a set of capabilities
  - Protocol defines interaction with service
  - All services require protocols
  - Not all protocols are used to provide services (e.g. IP, TLS)
- Examples: FTP and Web servers



Application Programming Interface (API)

- A specification for a set of routines to facilitate application development
  - Refers to definition, not implementation
  - E.g., there are many implementations of MPI
- Spec often language-specific (or IDL)
  - Routine name, number, order and type of arguments; mapping to language constructs
  - Behavior or function of routine
- Examples
  - GSS API (generic security services), MPI (message passing)

Software Development Kit

- A particular instantiation of an API
- SDK consists of libraries and tools
  - Provides implementation of API specification
- Can have multiple SDKs for an API
- Examples of SDKs
  - MPICH, Motif Widgets

Syntax

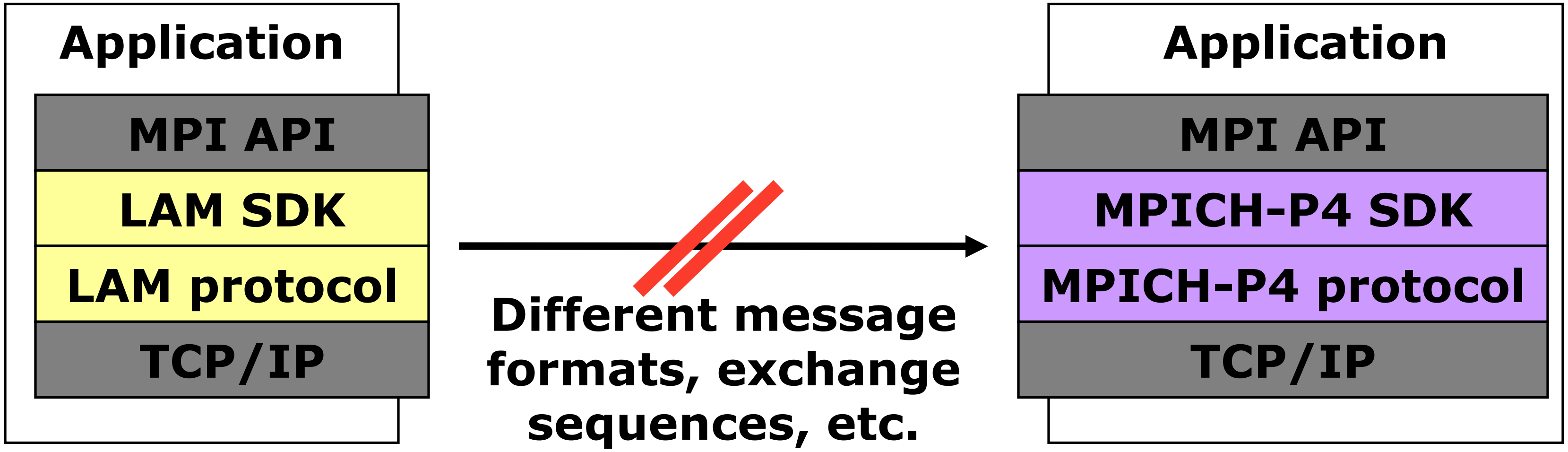
- Rules for encoding information, e.g.
  - XML, Condor ClassAds, Globus RSL
  - X.509 certificate format (RFC 2459)
  - Cryptographic Message Syntax (RFC 2630)
- Distinct from protocols
  - One syntax may be used by many protocols (e.g., XML); & useful for other purposes
- Syntaxes may be layered
  - E.g., Condor ClassAds -> XML -> ASCII
  - Important to understand layerings when comparing or evaluating syntaxes

A Protocol can have Multiple APIs

- TCP/IP APIs include BSD sockets, Winsock, System V streams, …
- The protocol provides interoperability: programs using different APIs can exchange information
- I don't need to know remote user's API

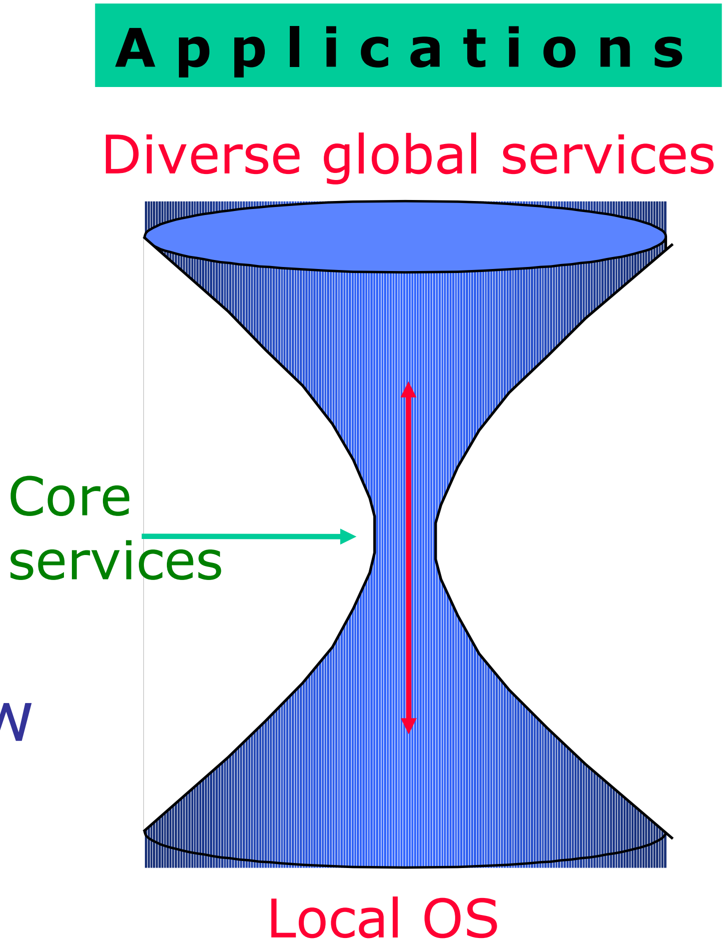
An API can have Multiple Protocols

- MPI provides portability: any correct program compiles & runs on a platform
- Does not provide interoperability: all processes must link against same SDK
  - E.g., MPICH and LAM versions of MPI

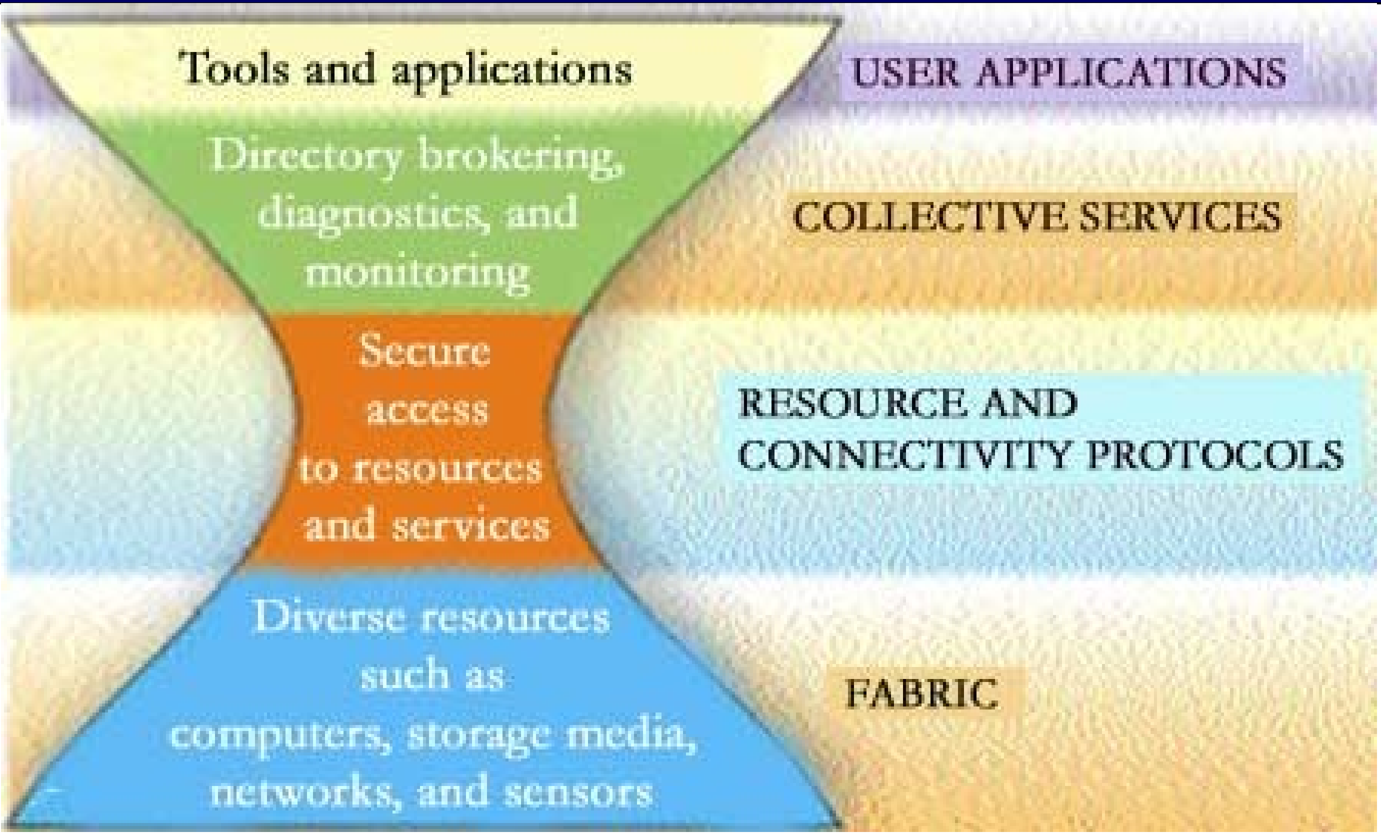


The Grid architecture

- Focus on architecture issues
  - Propose set of core services as basic infrastructure
  - Use to construct high-level, domain-specific solutions
- Design principles
  - Keep participation cost low
  - Enable local control
  - Support for adaptation
  - "IP hourglass" model



The hourglass model



- Grid architecture can be thought of a series of layers of different widths. At the center are the resource and connectivity layers, which contain a relatively small number of key protocols and application programming interfaces that must be implemented everywhere.

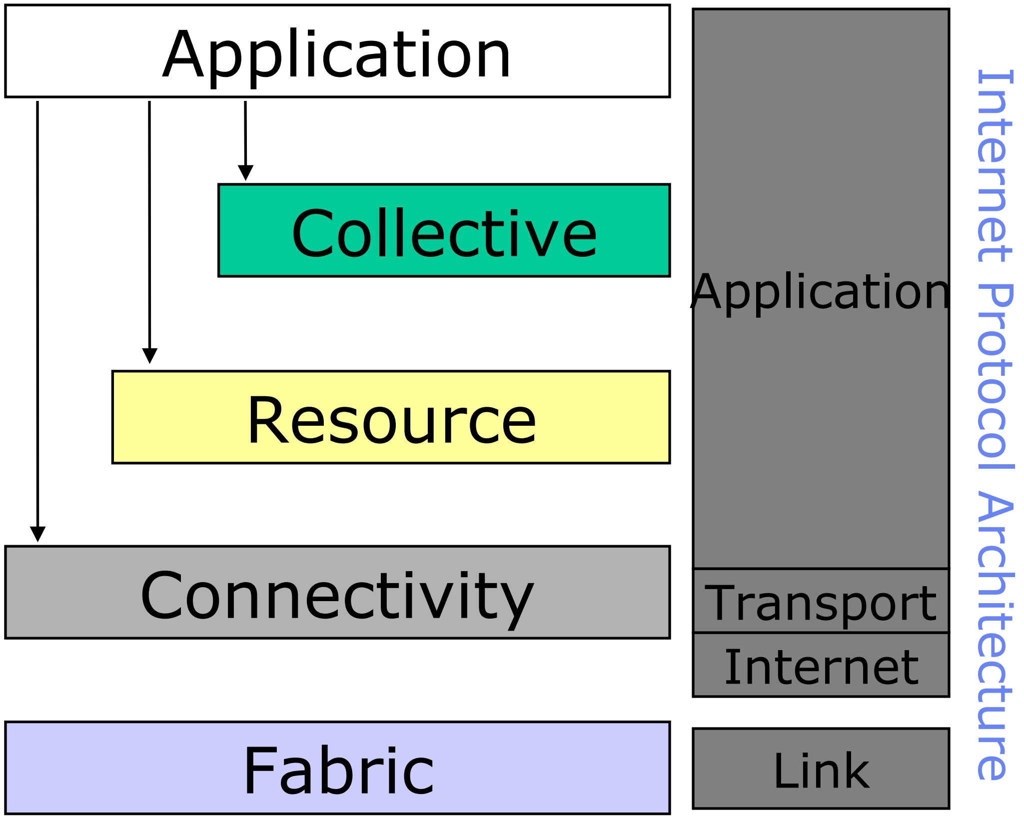
Layered Grid Architecture

“Coordinating multiple resources”: ubiquitous infrastructure services, app-specific distributed services

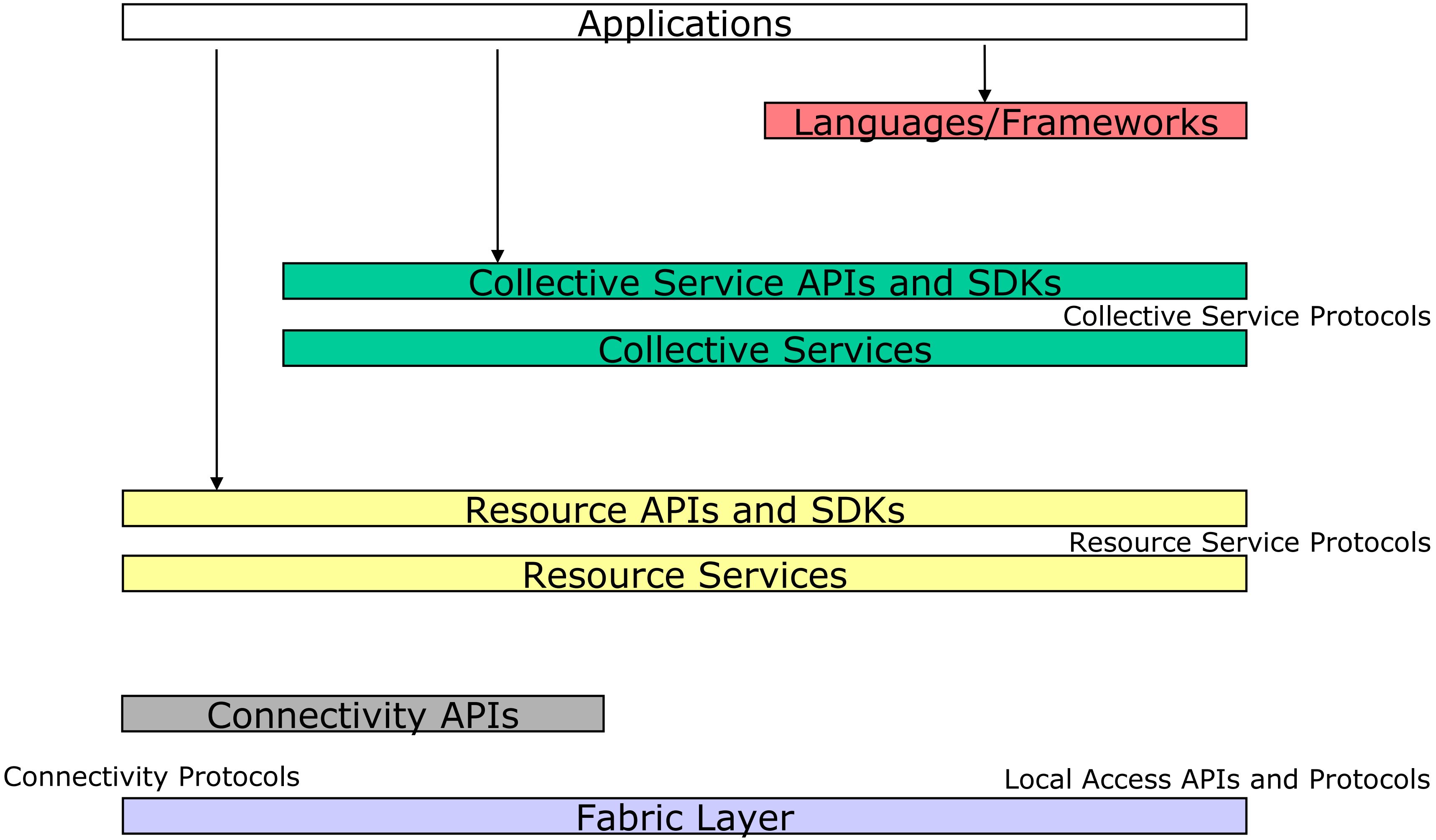
“Sharing single resources”: negotiating access, controlling use

“Talking to things”: communication (Internet protocols) & security

“Controlling things locally”: Access to, & control of, resources



Protocols, Services, and APIs Occur at Each Level



Fabric Layer
Protocols & Services

- Just what you would expect: the diverse mix of resources that may be shared
  - Individual computers, Clusters of computers, Condor pools, file systems, archives, metadata catalogs, networks, sensors, etc., 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 Protocols & Services

- 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, …

GSI: www.gridforum.org/security/gsi

Resource Layer Protocols & Services

- 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

GRAM, GridFTP, GRIS: www.globus.org

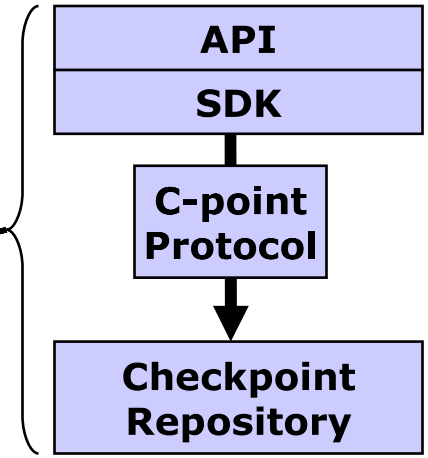
Collective Layer Protocols & Services

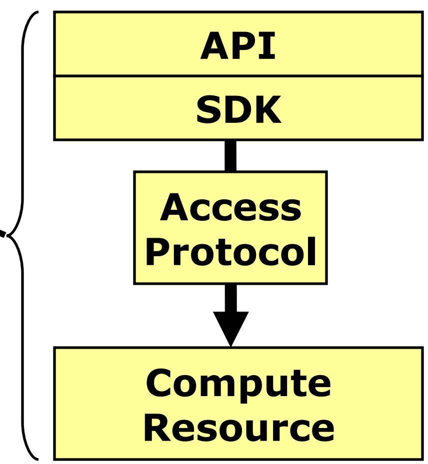
- Index servers aka metadirectory services
  - Custom views on dynamic resource collections assembled by a community
- Resource brokers (e.g., Condor Matchmaker)
  - Resource discovery and allocation
- Replica catalogs
- Replication services
- Co-reservation and co-allocation services
- Workflow management services
- Etc.

Condor: www.cs.wisc.edu/condor

Example: High-Throughput Computing System

App	High Throughput Computing System
Collective (App)	Dynamic checkpoint, job management, failover, staging
Collective (Generic)	Brokering, certificate authorities
Resource	Access to data, access to computers, access to network performance data
Connect	Communication, service discovery (DNS), authentication, authorization, delegation
Fabric	Storage systems, schedulers





Example: Data Grid Architecture

App	Discipline-Specific Data Grid Application
Collective (App)	Coherency control, replica selection, task management, virtual data catalog, virtual data code catalog, …
Collective (Generic)	Replica catalog, replica management, co-allocation, certificate authorities, metadata catalogs,
Resource	Access to data, access to computers, access to network performance data, …
Connect	Communication, service discovery (DNS), authentication, authorization, delegation
Fabric	Storage systems, clusters, networks, network caches, …

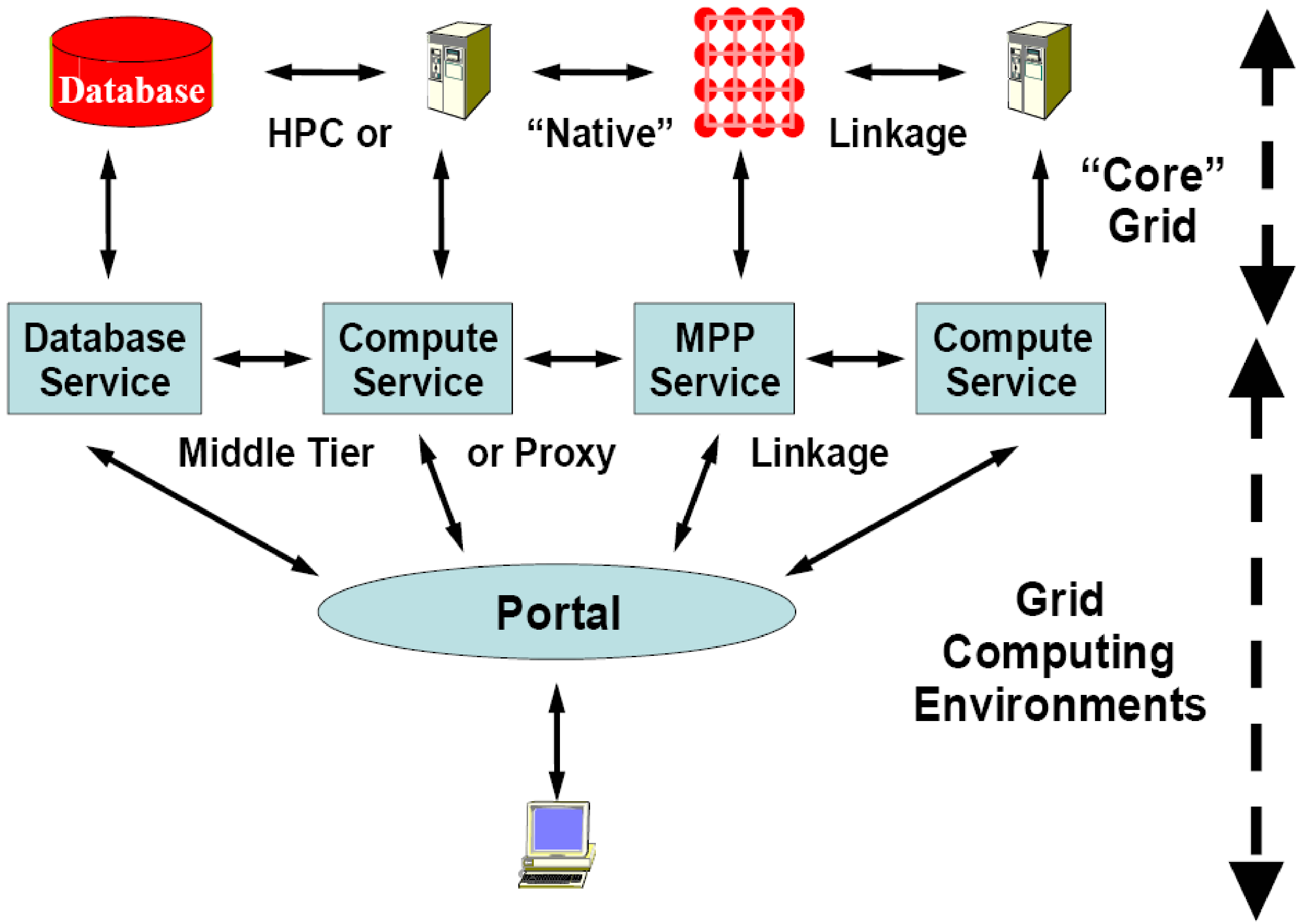
Outline

- Introduction to Grid Computing
- Some Definitions
- Grid Architecture
- The programming problem
- The Globus Toolkit™
  - Security
  - Resource Management,
  - Information Services
  - Data Management

The Programming Problem

- But how do I develop robust, secure, long-lived, well-performing applications for dynamic, heterogeneous Grids?
- I need, presumably:
  - Abstractions and models to add to speed/robustness/etc. of development
  - Tools to ease application development and diagnose common problems
  - Code/tool sharing to allow reuse of code components developed by others

Grid Computing Environments (GCE)



Management of resources and interaction between them, security and other such capabilities

Set of tools and technologies that allow users *easy* access to Grid resources and applications

GCE functions

- Programming the User Side of the Grid
  - Handling the basic components of a distributed computing system
  - Security (Authentication, Authorization and privacy)
  - Data Management
- Controlling user interaction – rendering any output and allowing user input in some web page.
  - This includes aggregation of multiple data sources in a single portal page

Grid Programming Technologies

- "Grid applications" are incredibly diverse (data, collaboration, computing, sensors, …)
  - Seems unlikely there is one solution
- Most applications have been written "from scratch," with or without Grid services
- Application-specific libraries have been shown to provide significant benefits
- No new language, programming model, etc., has yet emerged that transforms things
  - But certainly still quite possible

Examples of Grid Programming Technologies

- MPICH-G2: Grid-enabled message passing
- CoG Kits, GridPort: Portal construction, based on N-tier architectures
- GDMP, Data Grid Tools, SRB: replica management, collection management
- Condor-G: workflow management
- Legion: object models for Grid computing
- Cactus: Grid-aware numerical solver framework
  - Note tremendous variety, application focus

High-Throughput Computing and Condor

- High-performance: CPU cycles/second under ideal circumstances.
  - "How fast can I run simulation X on this machine?"
- High-throughput computing
  - CPU cycles/day (week, month, year?) under non-ideal circumstances
  - "How many times can I run simulation X in a month using all available machines?"
- Condor converts collections of distributively owned workstations and dedicated clusters into a distributed high-throughput computing facility
- Emphasis on policy management and reliability

Condor

" … Since the early days of mankind the primary motivation for the establishment of *communities* has been the idea that by being part of an organized group the capabilities of an individual are improved. The great progress in the area of inter-computer communication led to the development of means by which stand-alone processing sub-systems can be integrated into multi-computer *'communities'*. … "

Miron Livny, " *Study of Load Balancing Algorithms for Decentralized Distributed Processing Systems*.", Ph.D thesis, July 1983.

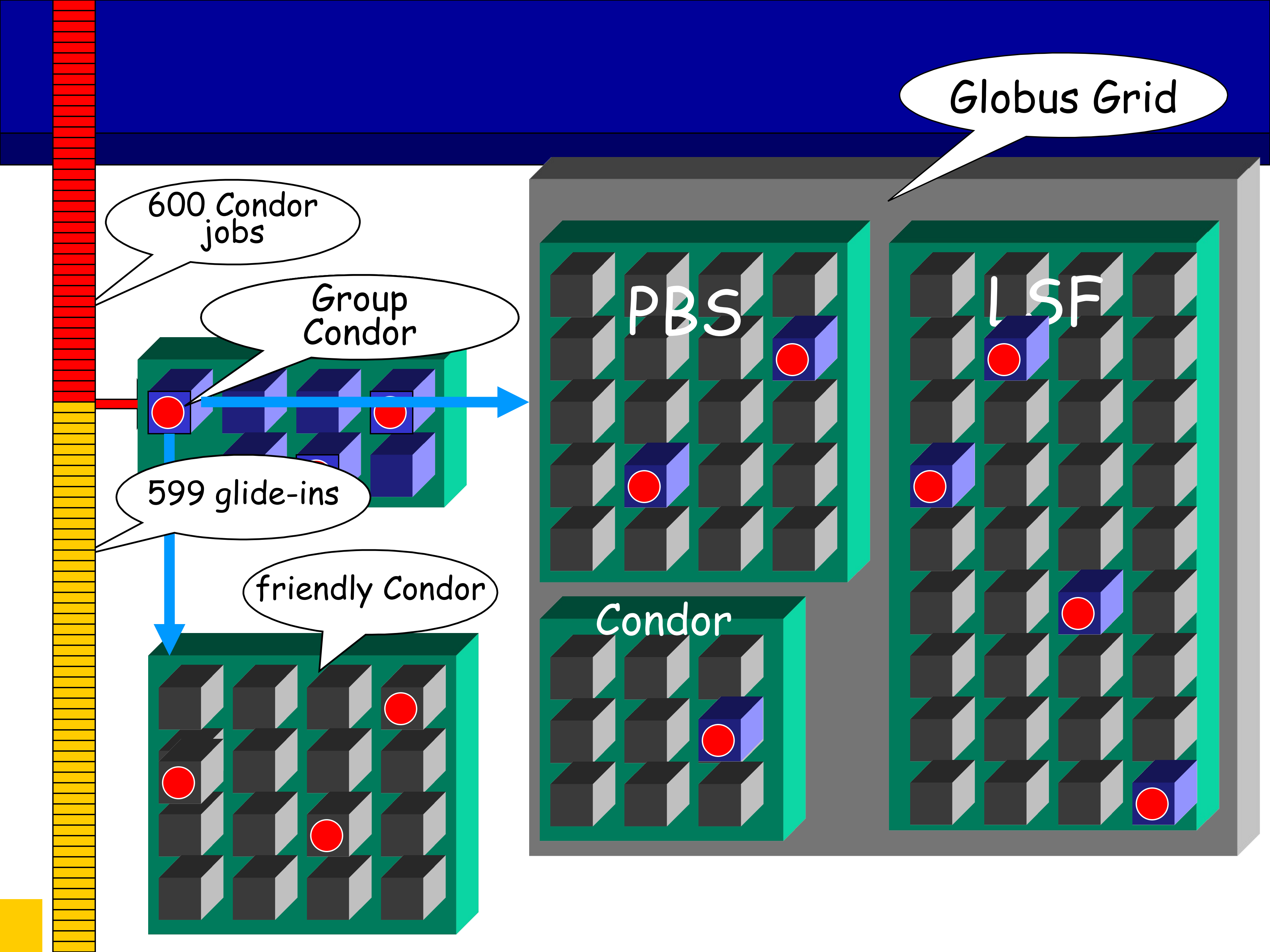
Main Condor capabilities

- Management of large collections of distributively owned heterogeneous resources (CPU, storage, network, software)
- Management of large (10K) collections of jobs.
- Remote Execution
- Remote I/O
- Checkpointing
- Matchmaking
- System administration

Condor and Globus

- job scheduling across multiple resources
- strong fault tolerance with checkpointing and migration
- layered over Globus as "personal batch system" for the Grid

- middleware deployed across entire Grid
- remote access to computational resources
- dependable, robust data transfer


Globus Grid
600 Condor jobs
Group Condor
PBS
LSF
599 glide-ins
friendly Condor
Condor

Portals

- N-tier architectures enabling thin clients, with middle tiers using Grid functions
  - Thin clients = web browsers
  - Middle tier = e.g. Java Server Pages, with Java CoG Kit, GPDK, GridPort utilities
  - Bottom tier = various Grid resources
- Numerous applications and projects, e.g.
  - Unicore, Gateway, Discover, Mississippi Computational Web Portal, NPACI Grid Port, Lattice Portal, Nimrod-G, Cactus, NASA IPG Launchpad, Grid Resource Broker, …

Outline

- Introduction to Grid Computing
- Some Definitions
- Grid Architecture
- The Programming Problem
- The Globus Toolkit™
  - Security
  - Resource Management
  - Information Services
  - Data Management

Globus 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
  - Make available under liberal open source license

General Approach

- Define Grid protocols & APIs
  - Protocol-mediated access to remote resources
  - Integrate and extend existing standards
  - “On the Grid” = speak “Intergrid” protocols
- Develop a reference implementation
  - Open source Globus Toolkit
  - Client and server SDKs, services, tools, etc.
- Grid-enable wide variety of tools
  - Globus Toolkit, FTP, SSH, Condor, SRB, MPI, …
- Learn through deployment and applications

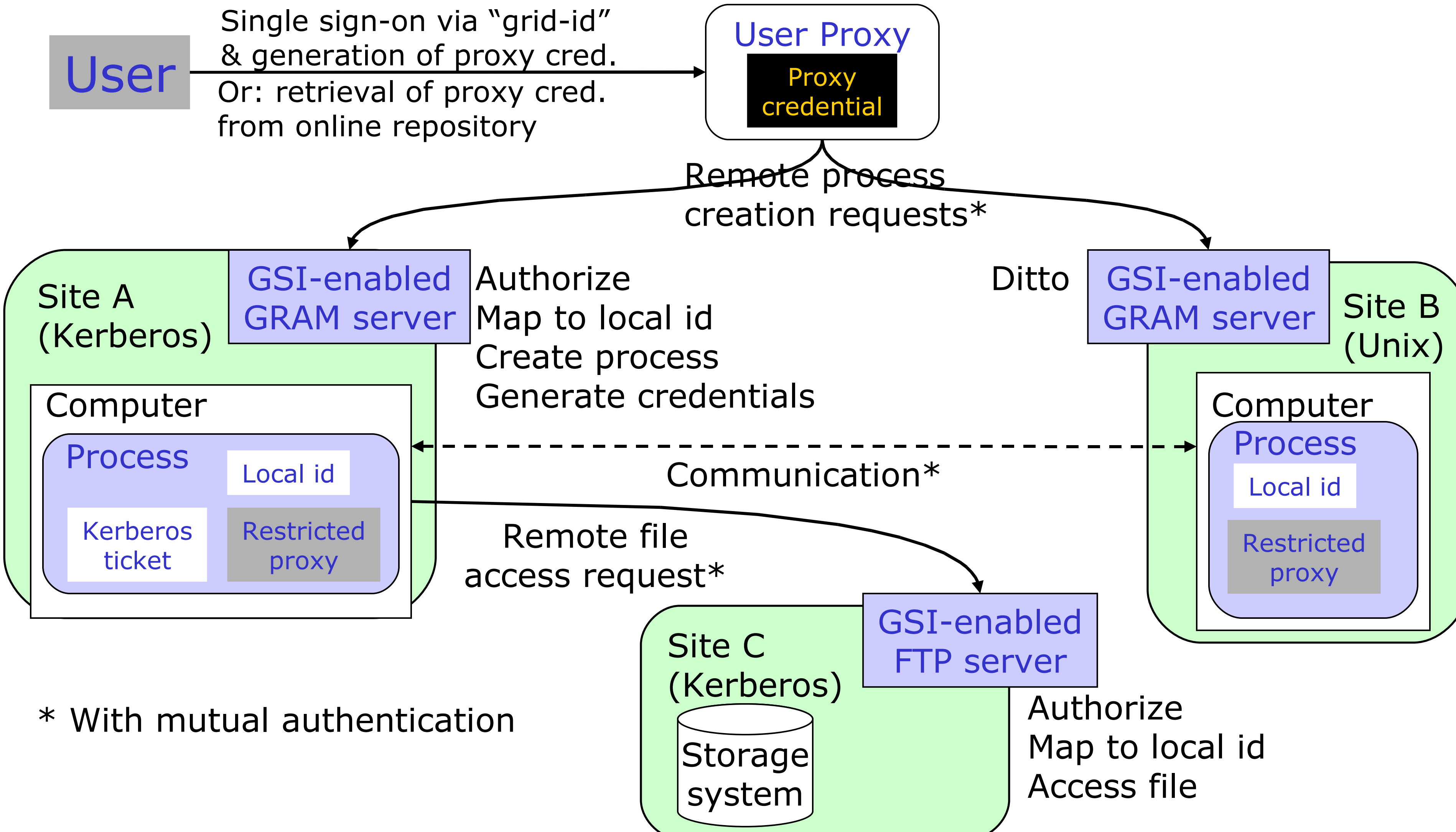
Key Protocols

- The Globus Toolkit™ centers around four key protocols
  - Connectivity layer:
    - *Security*: Grid Security Infrastructure (GSI)
  - Resource layer:
    - *Resource Management*: Grid Resource Allocation Management (GRAM)
    - *Information Services*: Grid Resource Information Protocol (GRIP)
    - *Data Transfer*: Grid File Transfer Protocol (GridFTP)
- Also key collective layer protocols
  - Info Services, Replica Management, etc.

Grid Security Infrastructure (GSI)

- Globus Toolkit implements GSI protocols and APIs, to address Grid security needs
- GSI protocols extends standard public key protocols
  - Standards: X.509 & SSL/TLS
  - Extensions: X.509 Proxy Certificates & Delegation
- GSI extends standard GSS-API

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

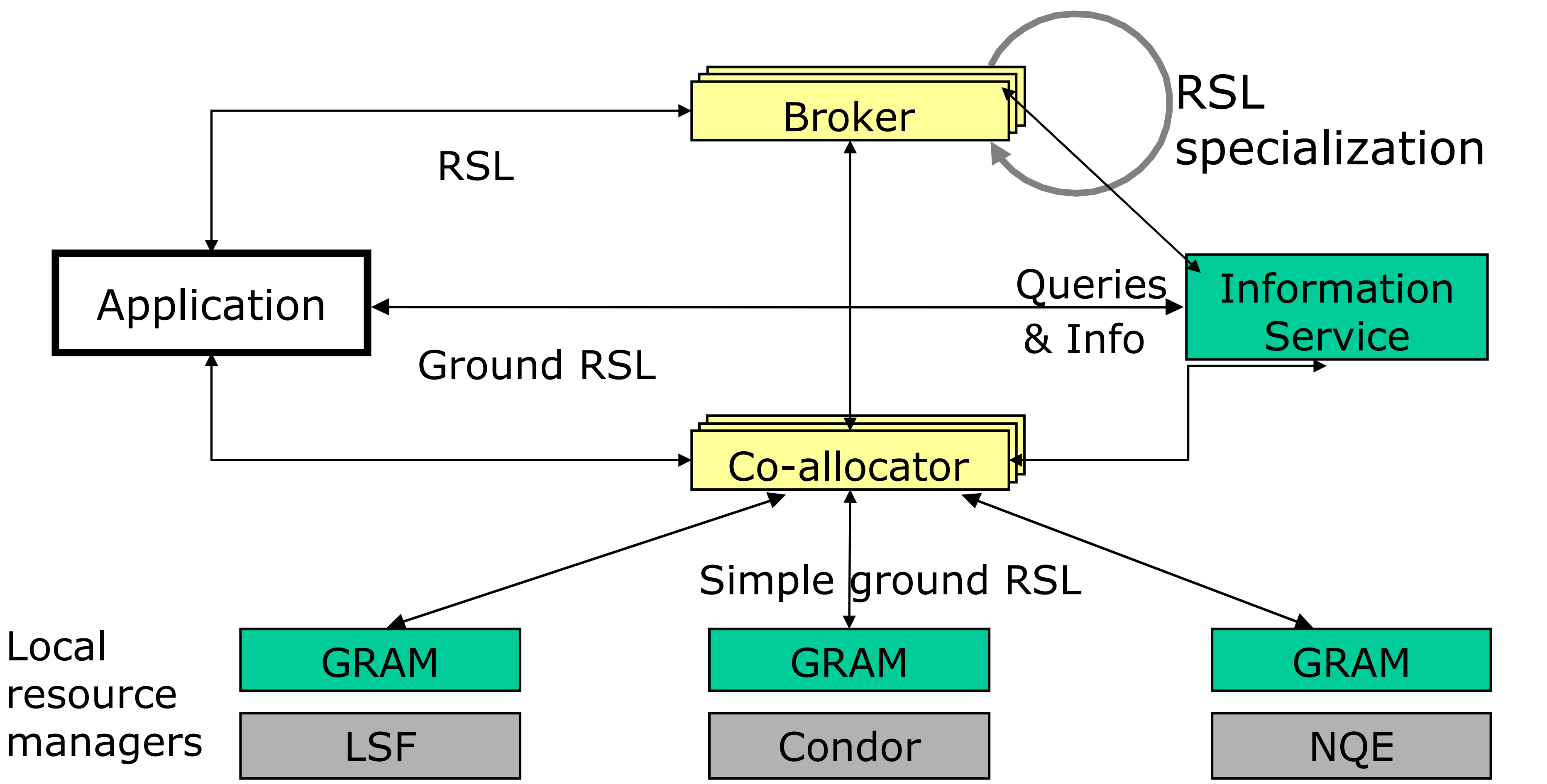


* With mutual authentication

Resource Management

- The Grid Resource Allocation Management (GRAM) protocol and client API allows programs to be started and managed on remote resources, despite local heterogeneity
- Resource Specification Language (RSL) is used to communicate requirements
- A layered architecture allows application-specific resource brokers and co-allocators to be defined in terms of GRAM services
  - Integrated with Condor, PBS, MPICH-G2, …

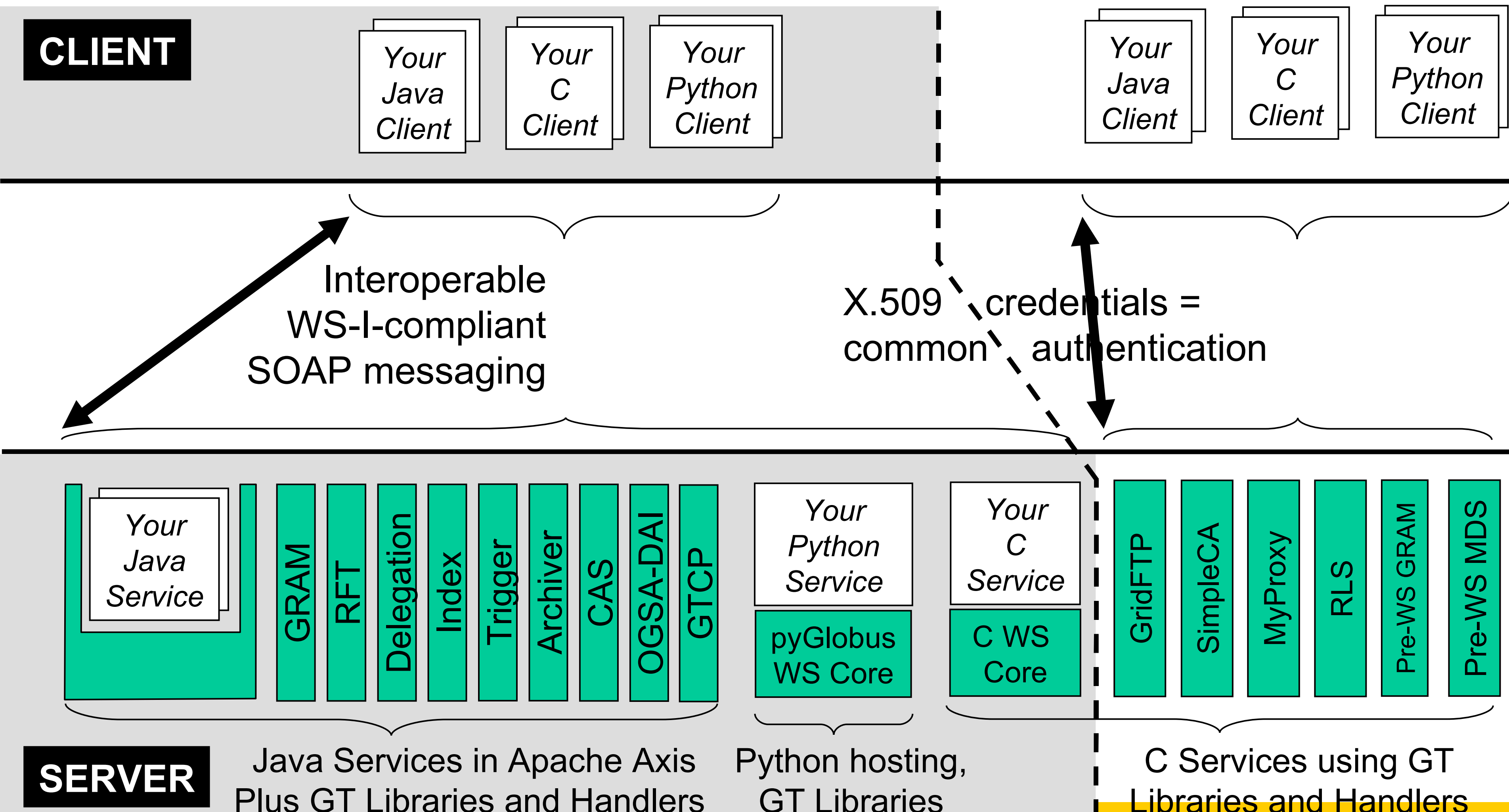
Resource Management Architecture



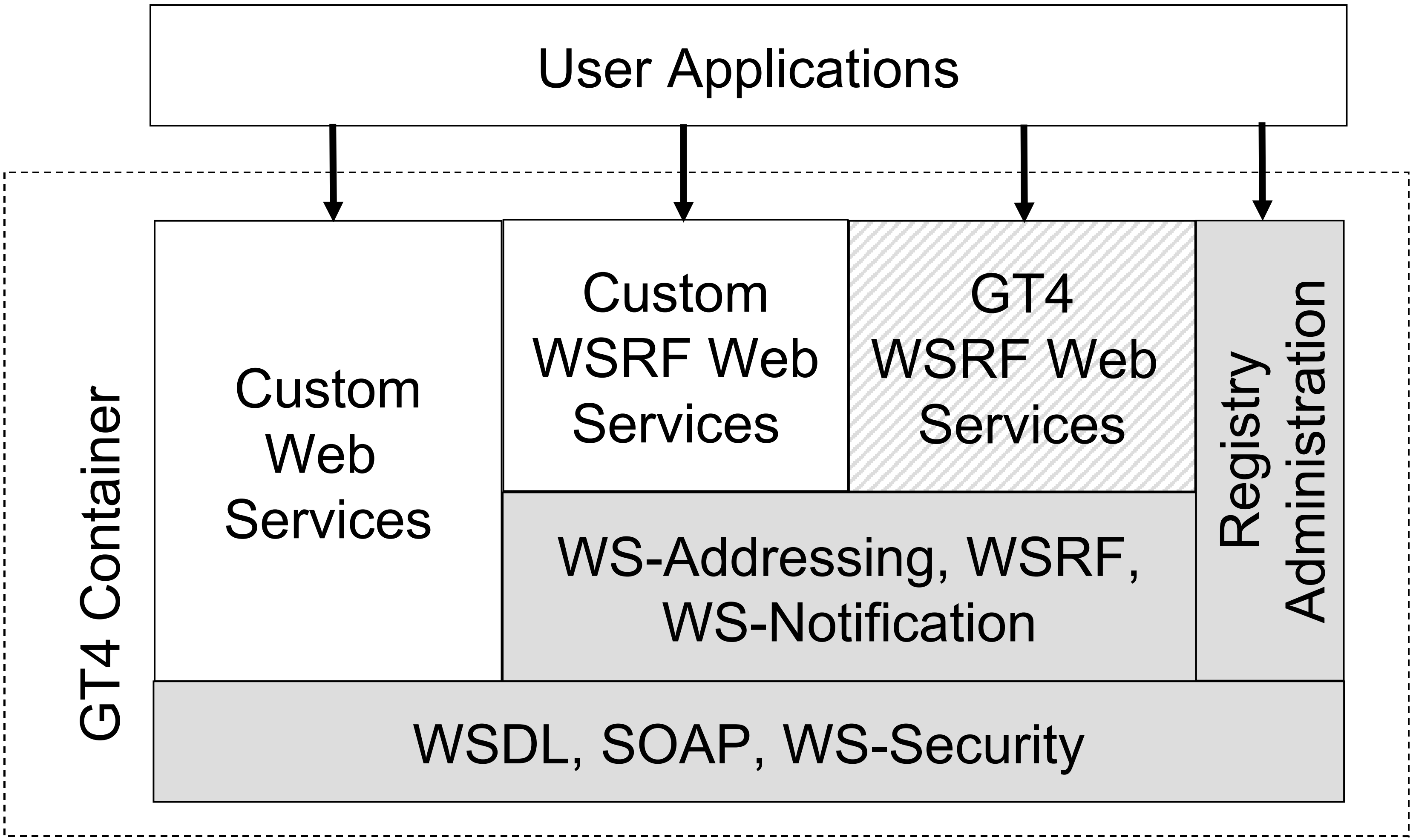
Data Access & Transfer

- GridFTP: extended version of popular FTP protocol for Grid data access and transfer
- Secure, efficient, reliable, flexible, extensible, parallel, concurrent, e.g.:
  - Third-party data transfers, partial file transfers
  - Parallelism, striping (e.g., on PVFS)
  - Reliable, recoverable data transfers
- Reference implementations
  - Existing clients and servers: wuftpd, ncftp
  - Flexible, extensible libraries in Globus Toolkit

GT4 Components



GT4 Web Services Runtime



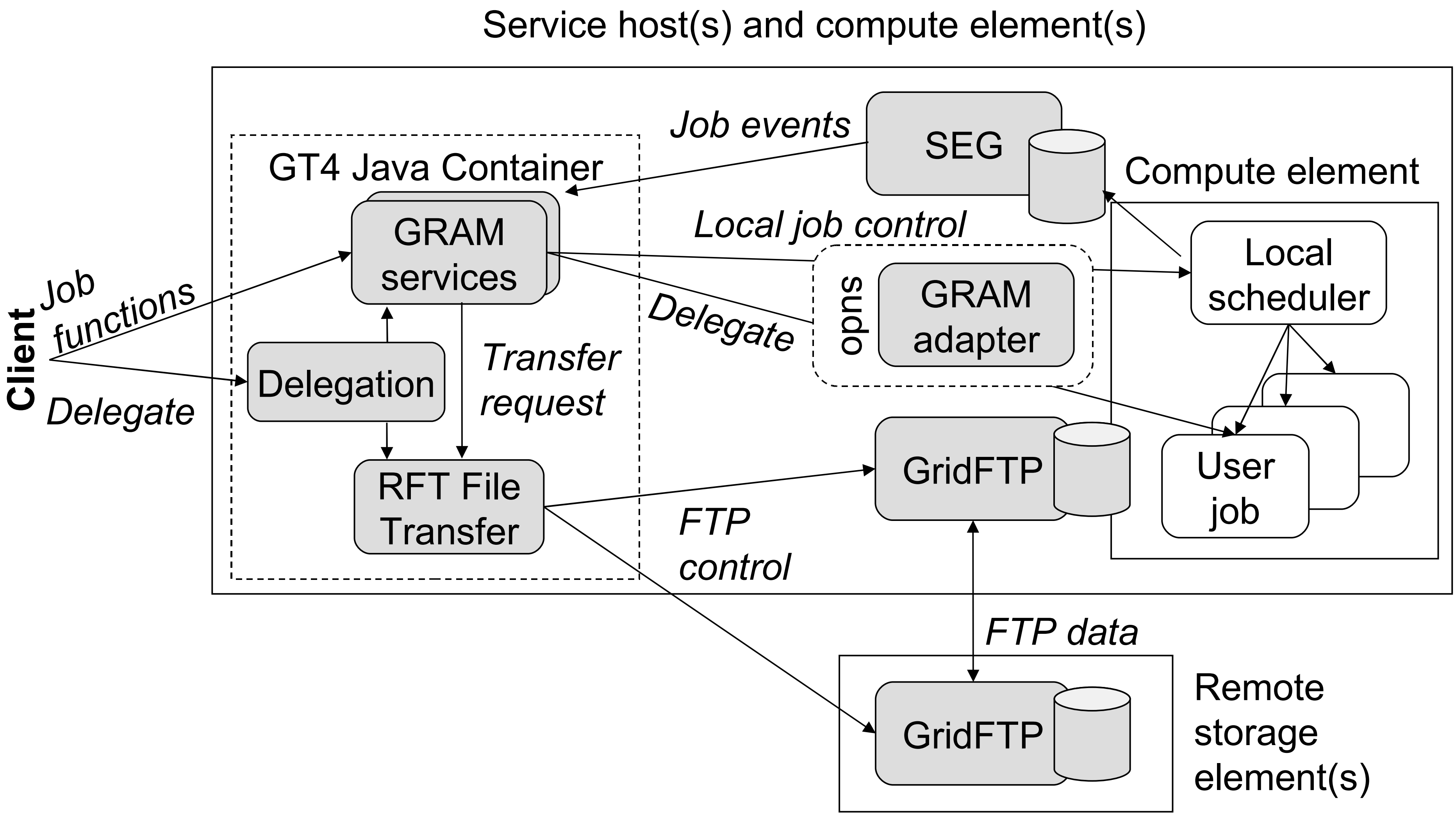
GT4 Data Management

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

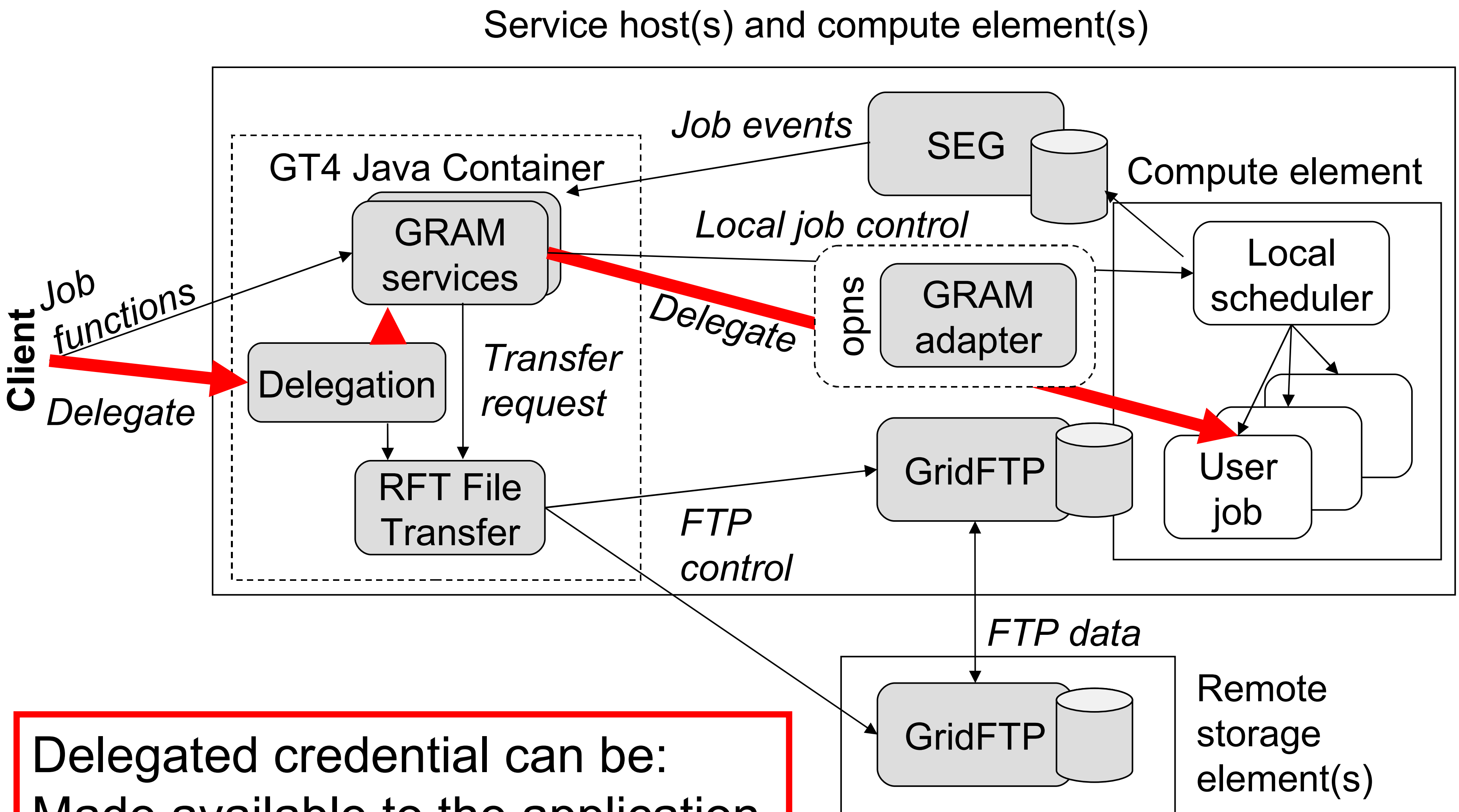
GT4 WS GRAM

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

GT4 WS GRAM Architecture

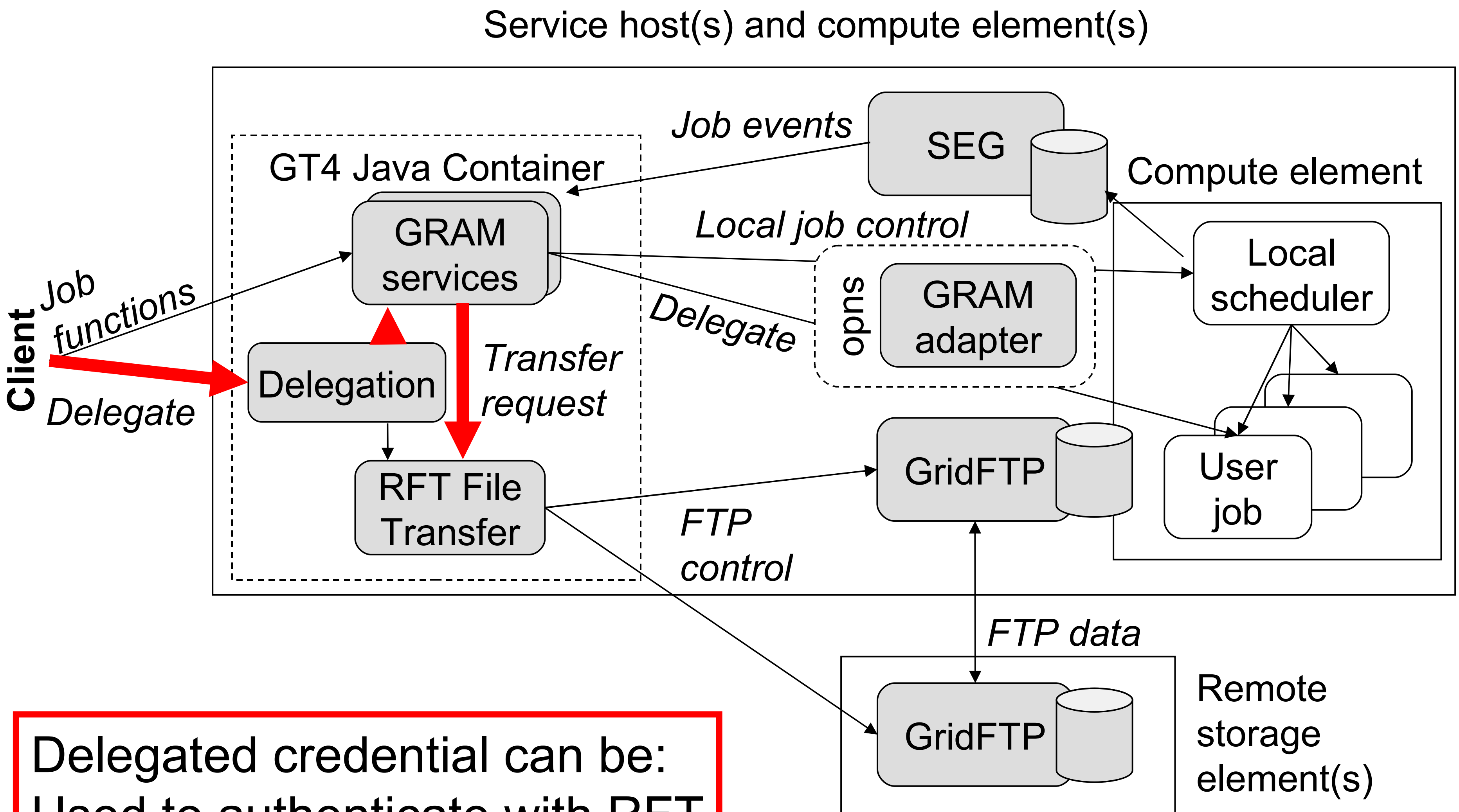


GT4 WS GRAM Architecture



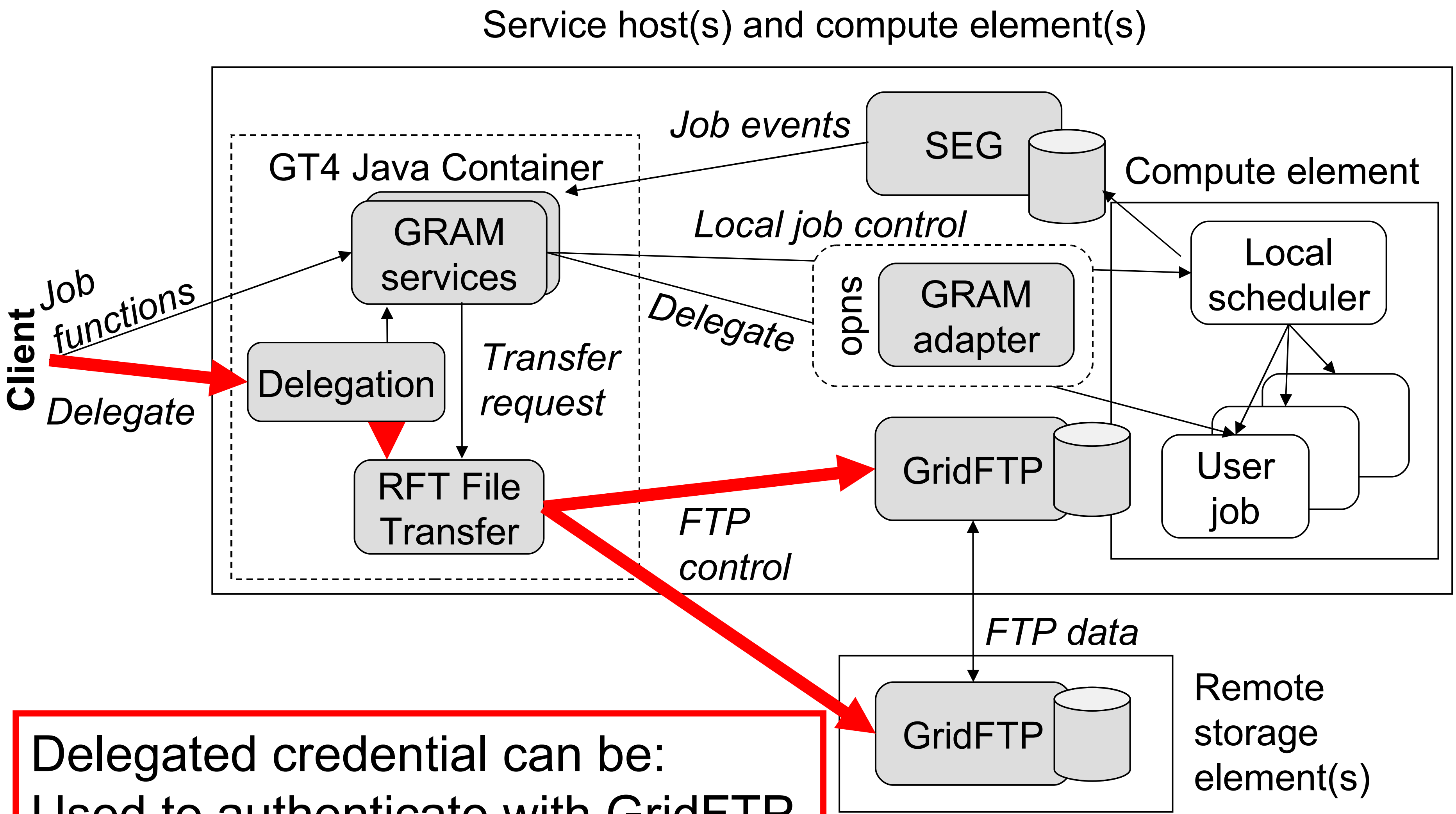
Delegated credential can be:
Made available to the application

GT4 WS GRAM Architecture



Delegated credential can be:
Used to authenticate with RFT

GT4 WS GRAM Architecture



Delegated credential can be:
Used to authenticate with GridFTP

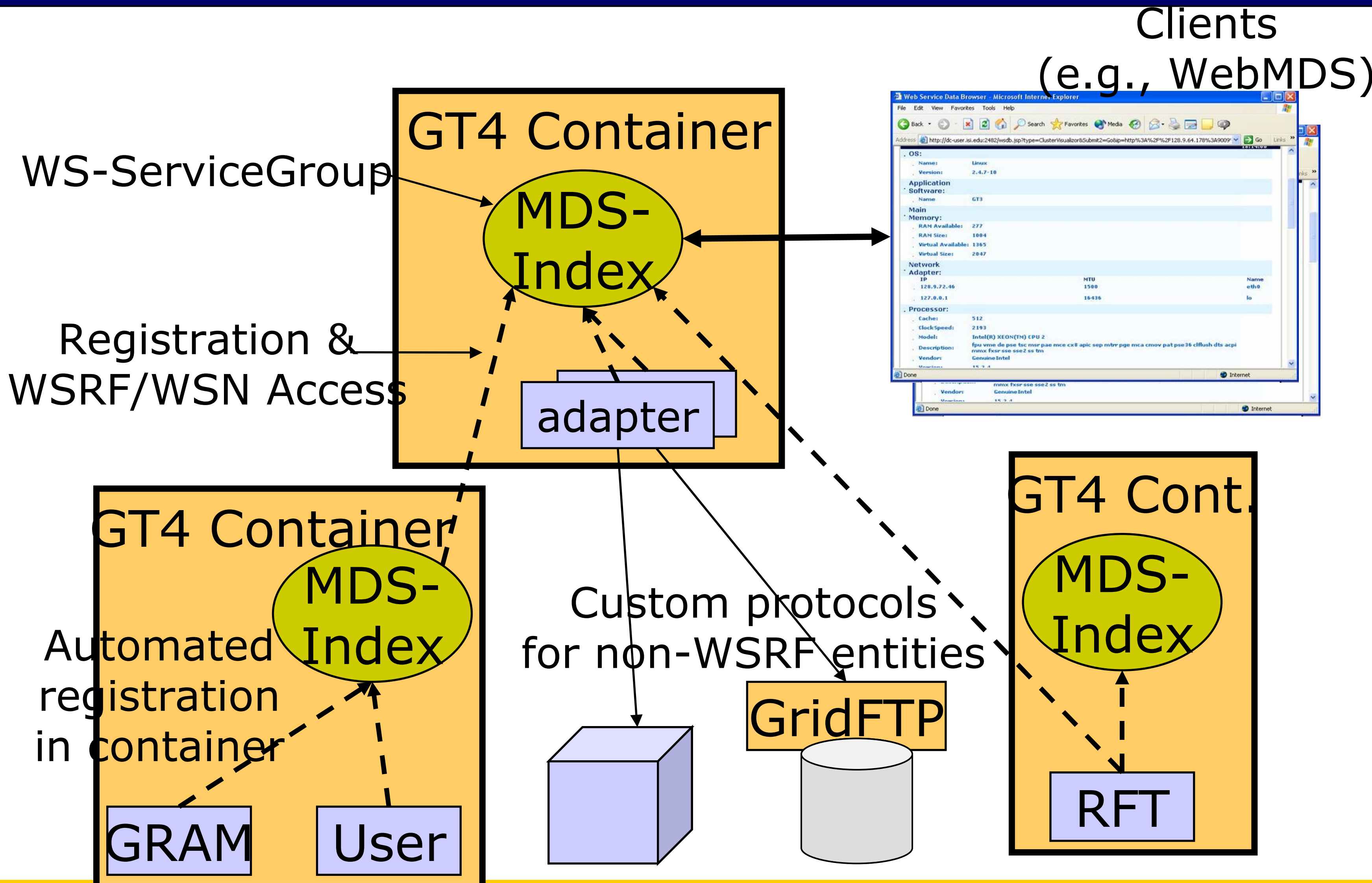
WS GRAM Performance

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

Monitoring and Discovery

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

GT4 Monitoring & Discovery



Index Server Performance

- As the MDS4 Index grows, query rate and response time both slow, although sublinearly
- Response time slows due to increasing data transfer size
  - Full Index is being returned
  - Response is re-built for every query

Summary

- The Grid problem: Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations
- Grid architecture emphasizes *systems problem*
  - Protocols & services, to facilitate interoperability and shared infrastructure services
- Globus Toolkit™: APIs, SDKs, and tools which implement Grid protocols & services
  - Provides basic software infrastructure for suite of tools addressing the *programming problem*

Summary (I)

- The Grid Computing Environment (GCE):
  - Handles the basic components of a Distributed Computing System
  - Programs the user side of the Grid
  - Controls the user interaction with the Grid
- The Core Grid handles all aspects related to the management of the Grid (it isn't part of the GCE)
- Globus Toolkit is based on the following protocols:
  - Resource Allocation and Management: Grid Resource Allocation Management (GRAM)
  - Information Services: Grid Resource Information Protocol (GRIP)
  - Data Transfer: Grid File Transfer Protocol (GridFTP)
  - Security: Grid Security Infrastructure (GSI)

Summary (I)

- Globus provides containers for executing C, Phyton and Java applications
- Globus Toolkit 4 is compliant to Open Grid Services Architecture (OGSA)
- The Information Service of Globus Toolkit 4 is based on the Trigger Service and the Web Monitoring and Discovering Service (WebMDS)
- The previous version of GT the Information Service were based on MDS+GIIS