Grid Computing

Ruini Xue

School of Computer Science and Engineering

University of Electronic Science and Technology of China

2016

Evolution of the Scientific Process

- Pre-electronic
  - Theorize &/or experiment, alone or in small teams; publish paper
- Post-electronic
  - Construct and mine very large databases of observational or simulation data
  - Develop computer simulations & analyses
  - Exchange information quasi-instantaneously within large, distributed, multidisciplinary teams
- Collaboration & Computation

Difficulties in present systems

- As technology is constantly changing there is a need for regular upgrade/enhancement
- Cluster/Servers are not fail safe and fault tolerant.
- Many systems are dedicated to a single application, thus idle when the application has no load
- Many clusters in the organization remain idle
- For operating a computer centre 75% cost comes from environment upkeep, staffing, operation and maintenance.
- Computers, Networks, Clusters, Parallel Machines and Visual systems are not tightly coupled by software; difficult for users to use it

The Ultimate Goal

- In future I will not know or care where my application will be executed as I will acquire and pay to use these resources as I need them
- Water, power: utility computing

Why Grids?

- Large-scale science and engineering are done through the interaction of people, heterogeneous computing resources, information systems, and instruments, all of which are geographically and organizationally dispersed.

- The overall motivation for "Grids" is to facilitate the routine interactions of these resources in order to support large-scale science and Engineering.

Why Grids?

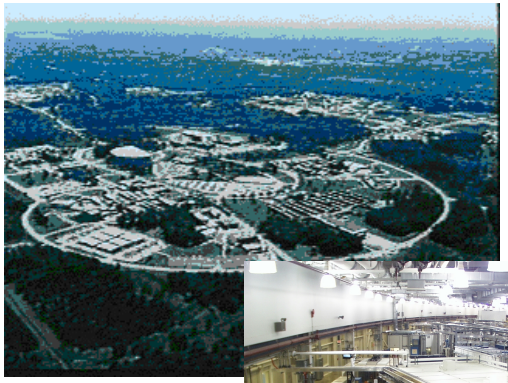
- A biochemist exploits 10,000 computers to screen 100,000 compounds in an hour
- 1,000 physicists worldwide pool resources for petaop analyses of petabytes of data
- Civil engineers collaborate to design, execute, & analyze shake table experiments
- Climate scientists visualize, annotate, & analyze terabyte simulation datasets
- An emergency response team couples real time data, weather model, population data

Why Grids? (contd)

- A multidisciplinary analysis in aerospace couples code and data in four companies
- A home user invokes architectural design functions at an application service provider
- An application service provider purchases cycles from compute cycle providers
- Scientists working for a multinational soap company design a new product
- A community group pools members' PCs to analyze alternative designs for a local road

Online Access to Scientific Instruments

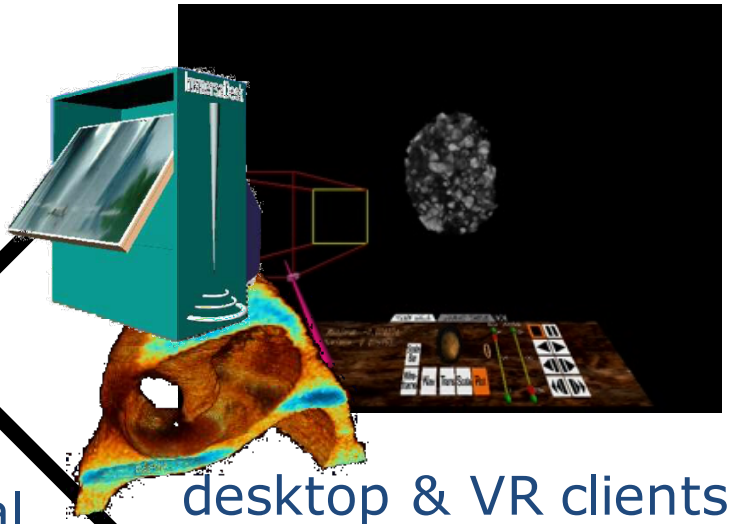
Advanced Photon Source



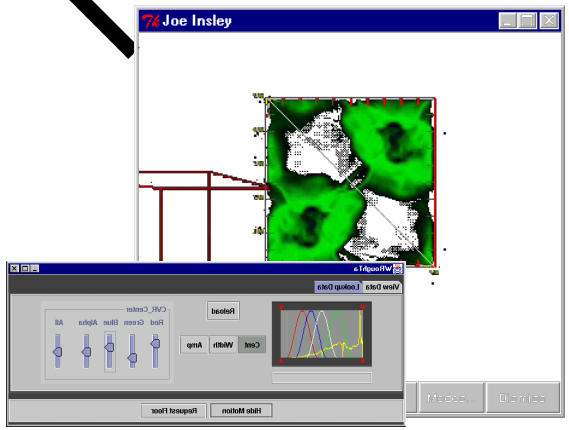
real-time collection

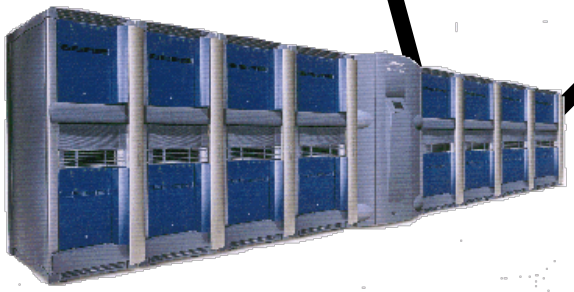
wide-area dissemination

archival storage



desktop & VR clients with shared controls



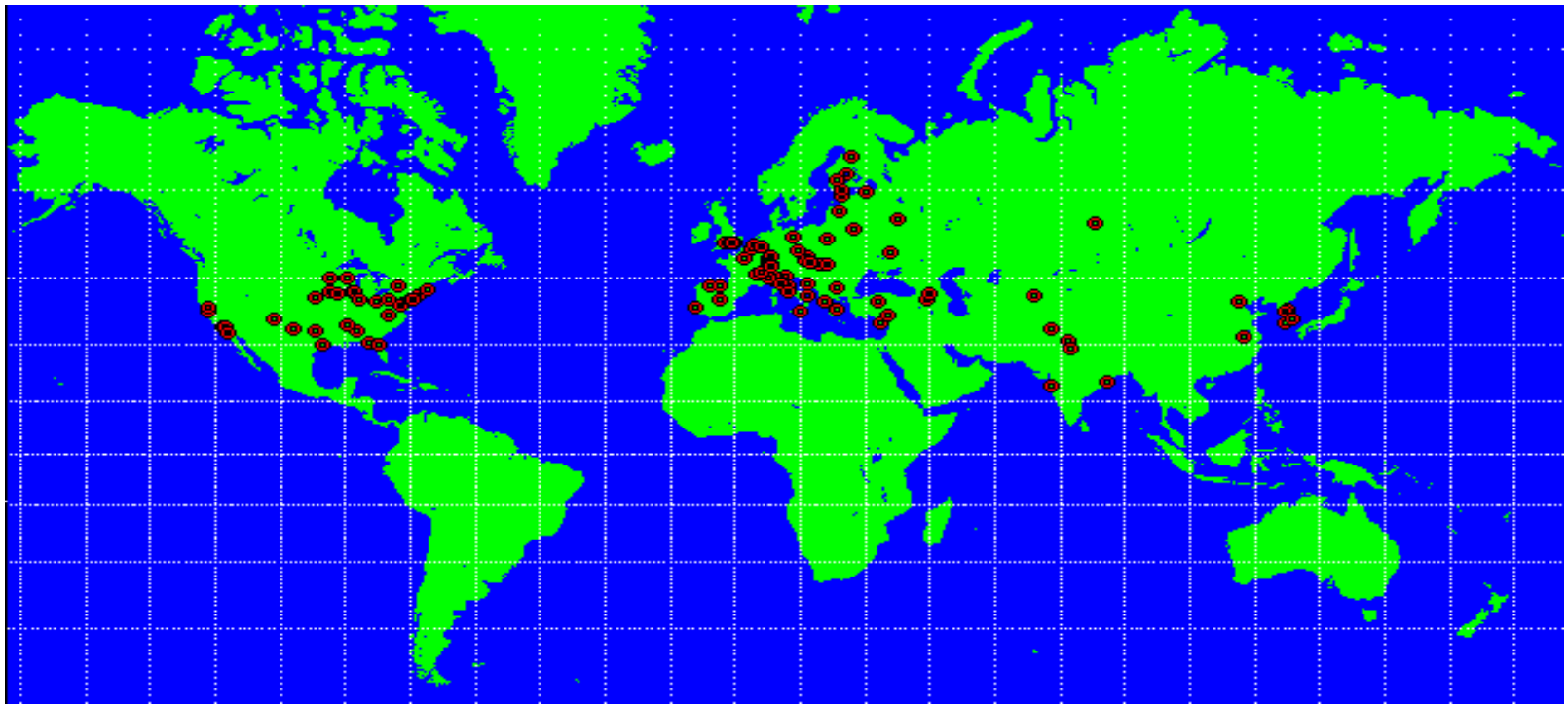


tomographic reconstruction

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

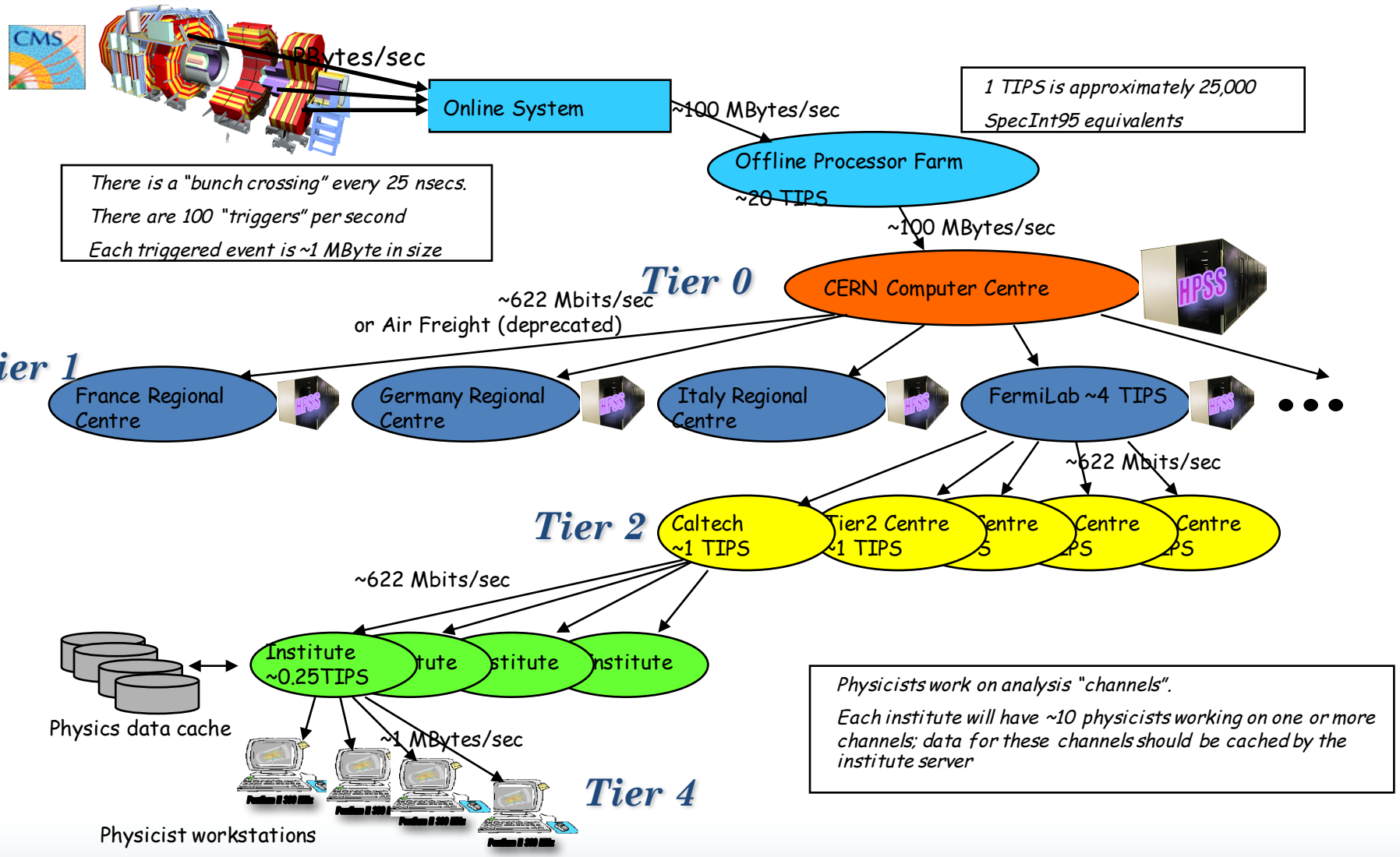
An Example Virtual Organization: CERN's Large Hadron Collider

- 1800 Physicists, 150 Institutes, 32 Countries



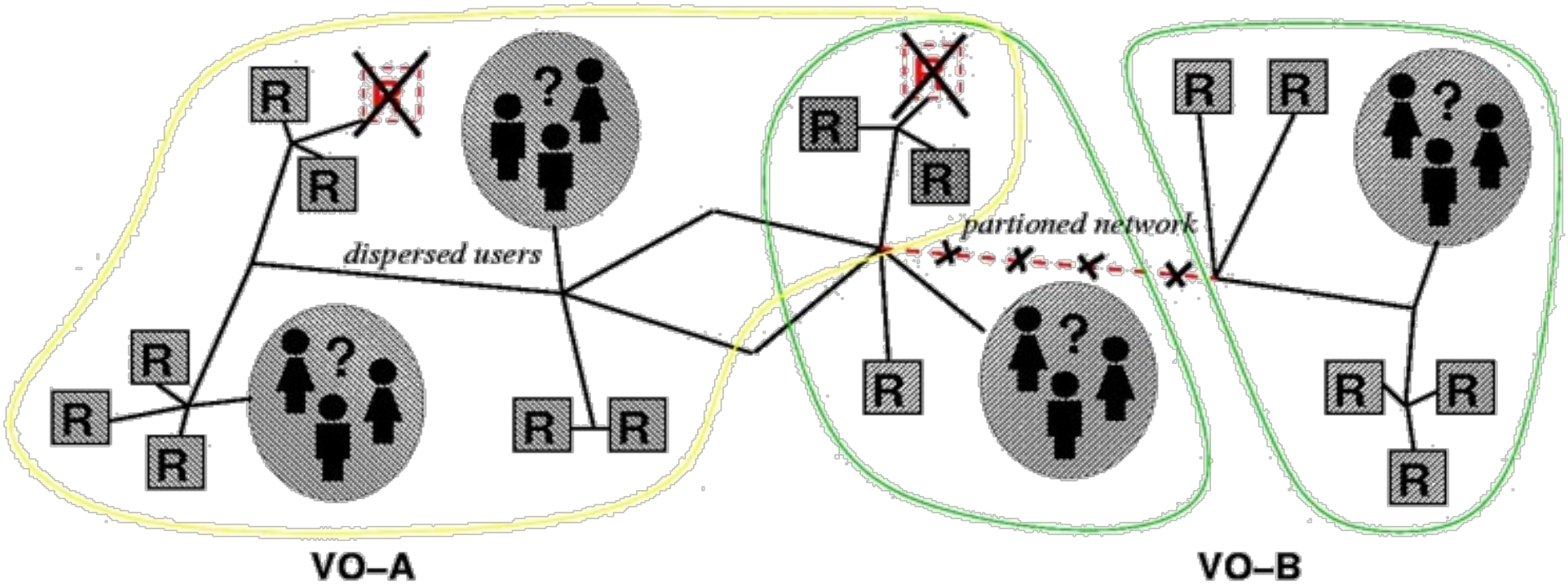
- 100 PB of data by 2010; 50,000 CPUs

Grid Communities & Applications: Data Grids for High Energy Physics



The Grid

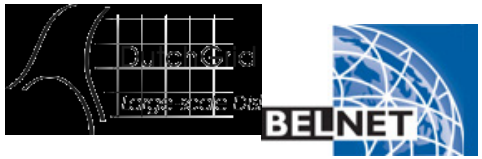
- "Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations"



The Grid Problem

- Flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resource
  - *From "The Anatomy of the Grid: Enabling Scalable Virtual Organizations"*
- Enable communities ("virtual organizations") to share geographically distributed resources as they pursue common goals -- assuming the absence of
  - central location,
  - central control,
  - omniscience,
  - existing trust relationships.

Grid initiatives: academy



CroGrid

Grid

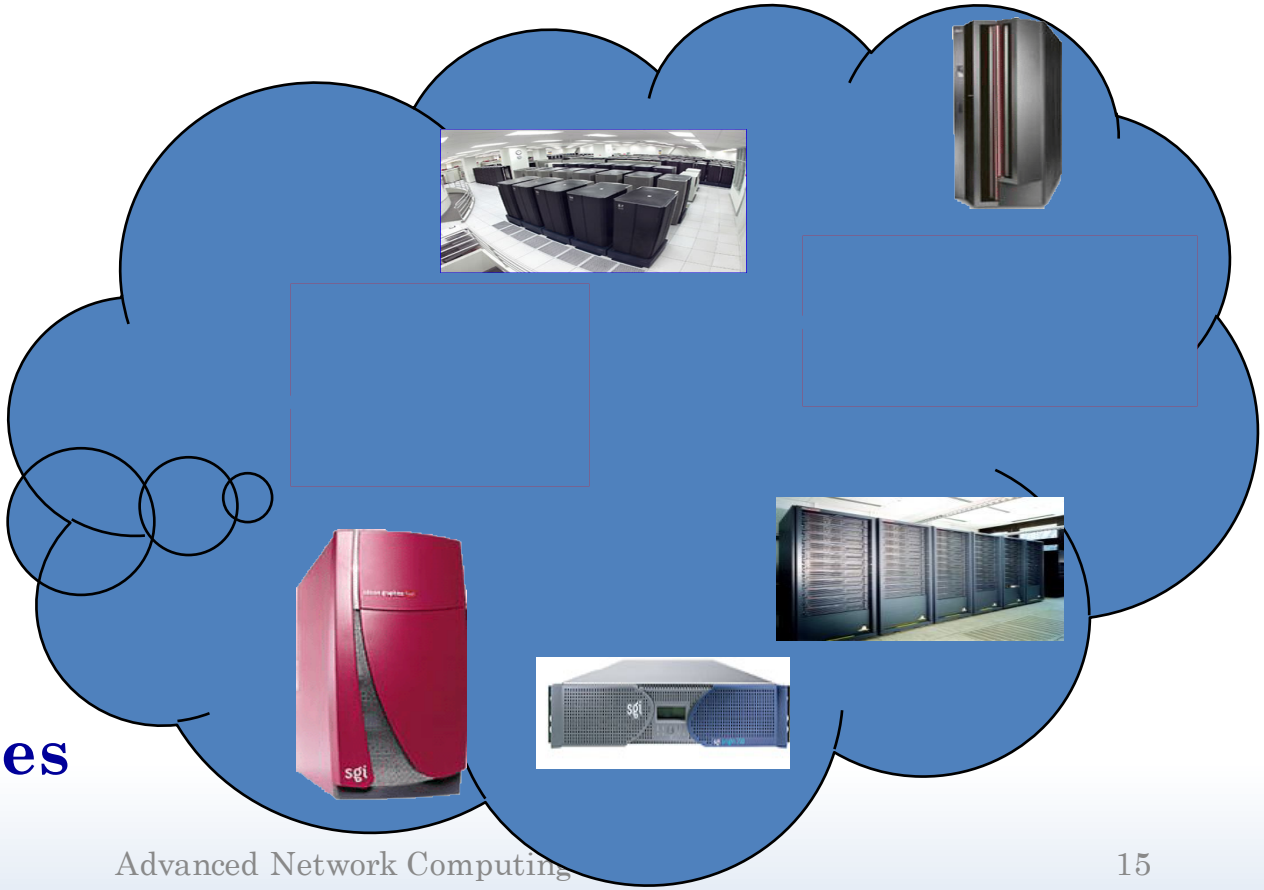
- A kind of open standard distributed infrastructure that enables flexible, secure, coordinated resource sharing among dynamic collections of trusted resources belonging to diverse organizations across the globe ensuring user's QoS requirements.
- Enables
  - Virtual Organization
  - Dynamic Resource sharing

Grid Concept

User Access Point

Result

Resource Broker



Grid Resources

Building Blocks

- Network
- Computing Nodes
- Storage


NPACI
Blue Horizon


ASCI White

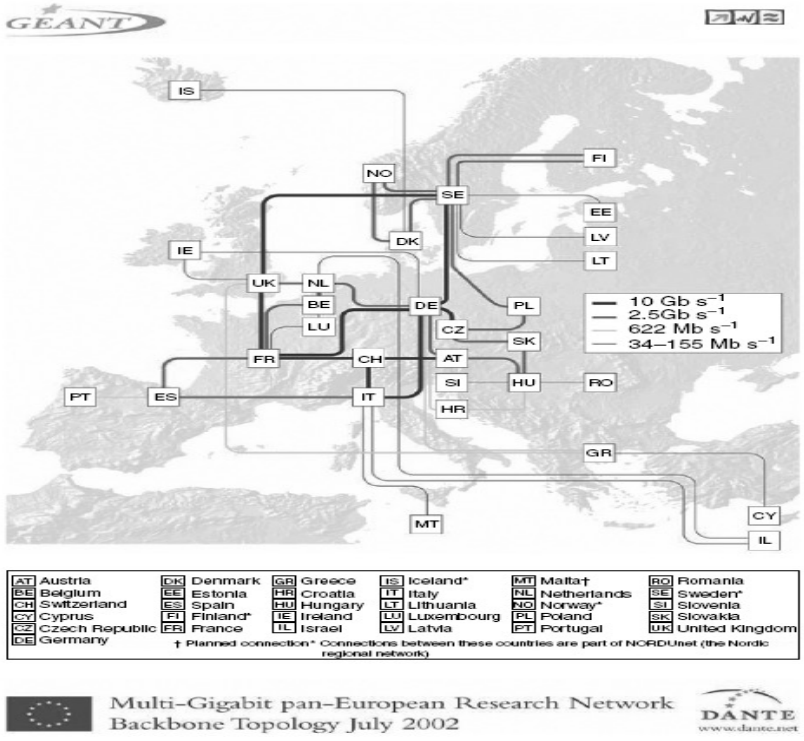

ASCI Q


ASCI
Blue Mountain


ASCI Cplant

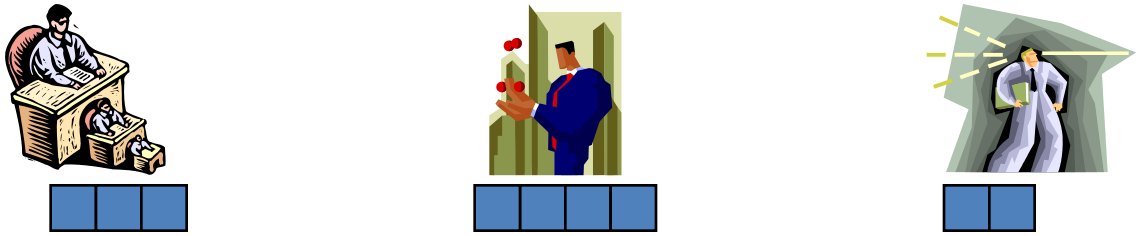

ASCI Blue-Pacific


ASCI Red



PARAM
PADMA

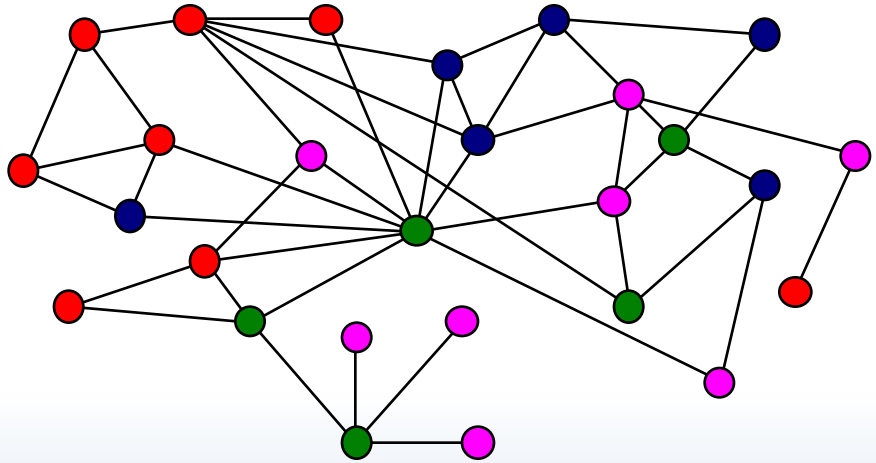
Context



jobs / legacy code / binary executables

Middleware

Resources



Map to resources

Elements of the Problem

- 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, …
  - Binary compability
- Dynamic, multi-institutional virtual orgs
  - Community overlays on classic org structures
  - Large or small, static or dynamic

Three Generations of Grid

1
- Local “metacomputers”: e.g., FAFNER, I-WAY
  - Distributed file systems
  - Site-wide single sign-on
- "Metacenters" explore inter-organizational integration
- Totally custom-made, top-to-bottom: proofs of concept

2
- Utilize software services and communications protocols developed by grid projects: Middleware
  - *Condor, Globus, UNICORE, Legion, etc.*
- Need significant customization to deliver complete solution
- Interoperability is still very difficult!

3
- **Common interface specifications** support interoperability of discrete, independently developed services
- **Competition** and **interoperability** among **applications**, **toolkits, *and*** implementations of **key services**

Architecture *keeps evolving!*

Why Discuss Architecture?

- Descriptive
  - Provide a common vocabulary for use when describing Grid systems
- Guidance
  - Identify key areas in which services are required
- Prescriptive
  - Define standard “Intergrid” protocols and APIs to facilitate creation of interoperable Grid systems and portable applications

One View of Requirements

- Identity & authentication
- Authorization & policy
- Resource discovery
- Resource characterization
- Resource allocation
- (Co-)reservation, workflow
- Distributed algorithms
- Remote data access
- High-speed data transfer
- Performance guarantees
- Monitoring
- Adaptation
- Intrusion detection
- Resource management
- Accounting & payment
- Fault management
- System evolution
- …

The nature of grid architecture

- A grid architecture identifies fundamental system components, specifies the purpose and function of these components, and indicate how these components interact.

The Nature of Grid Architecture

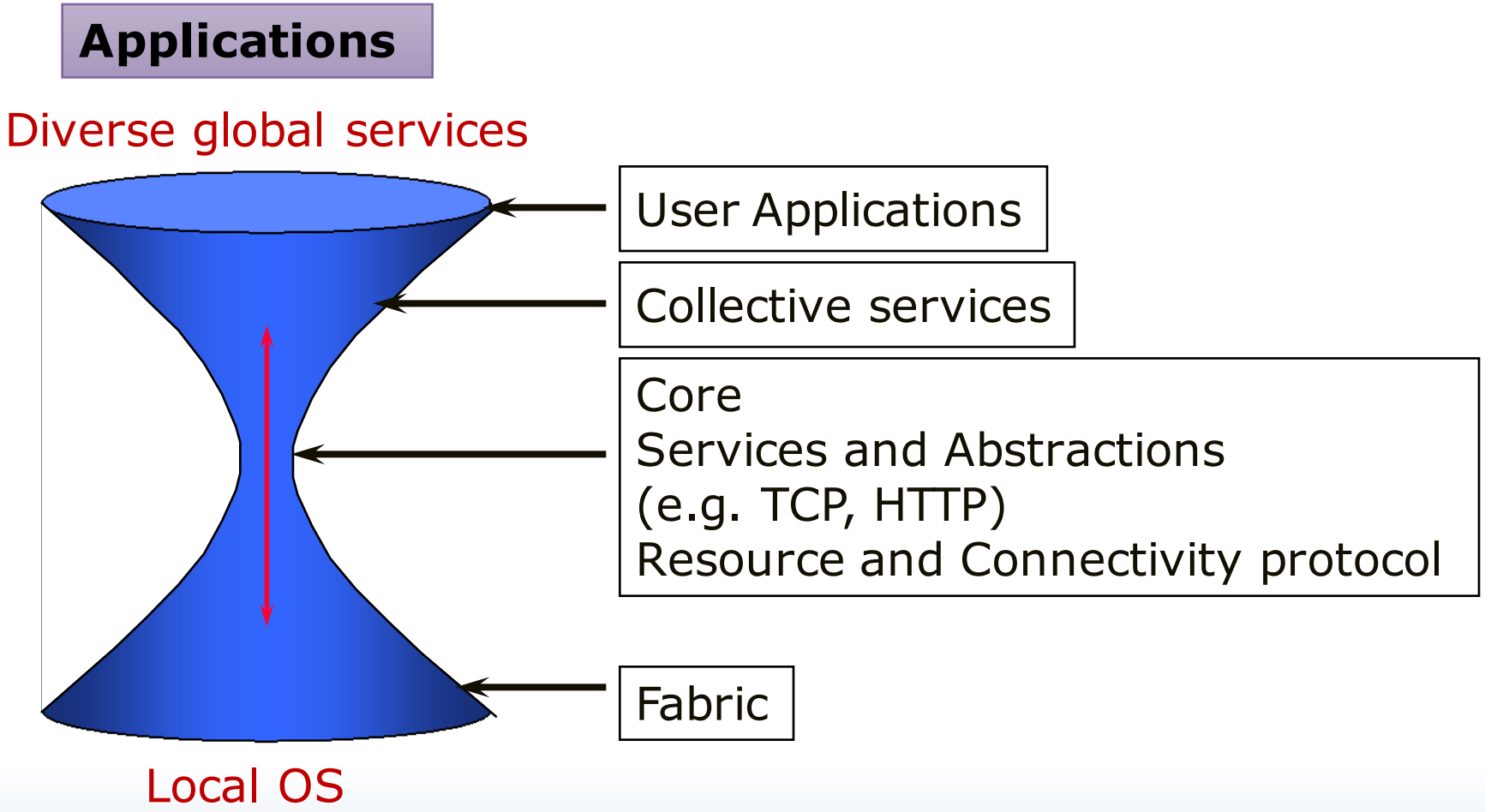
- Grid's protocols allow VO users and resources to negotiate, establish, manage and exploit sharing relationships.
  - Interoperability is a fundamental concern
  - The protocols are critical to interoperability
  - Services are important
  - We need to consider APIs and SDKs

- VO: Virtual Organization

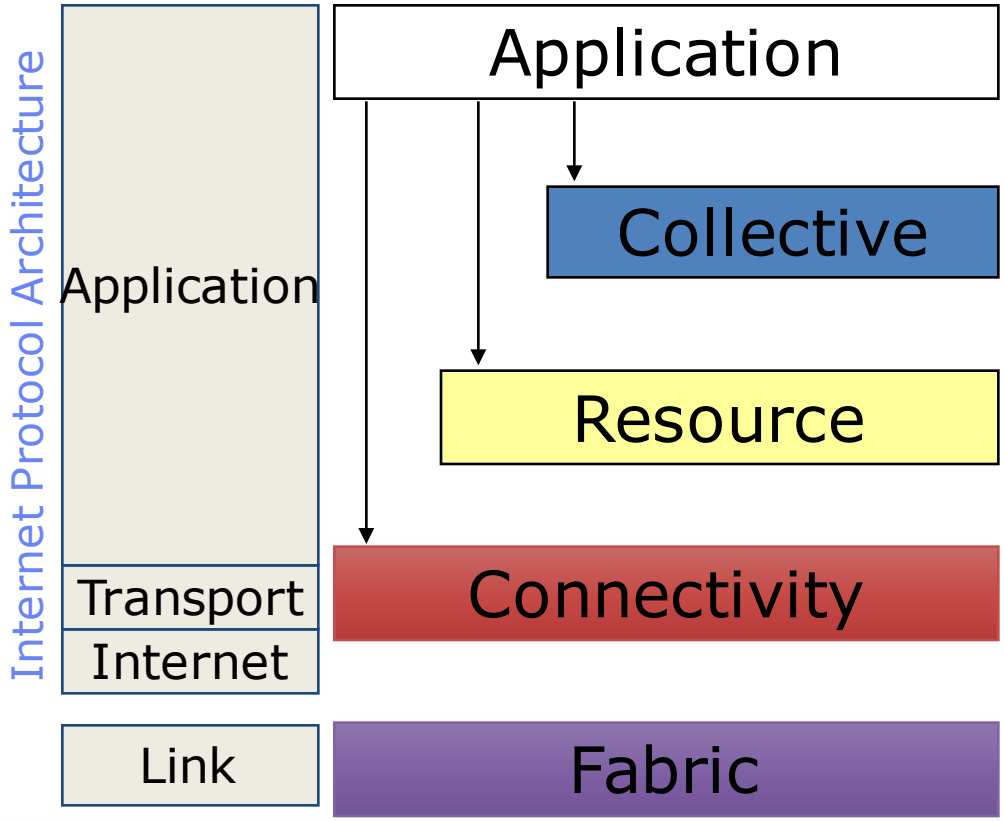
Grid architecture requirements

- The components are
  - numerous
  - owned and managed by different, potentially mutually distrustful organisations and individuals
  - may be potentially faulty
  - have different security requirements and policies
  - heterogeneous
  - connected by heterogeneous, multilevel networks
  - have different resource management policies
  - are likely to be geographically separated

The Hourglass Model



Layered Grid Architecture (By Analogy to Internet 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

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., etc.
- 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, …

GSI: www.gridforum.org/security

Resource Layer

- This is for the secure negotiation, initiation, monitoring, control, accounting, and payment of sharing operations on individual resources.
  - Information Protocols (inform about the structure and state of the resource)
  - Management Protocols (negotiate access to a shared resource)

Resource Layer

- Grid Resource Allocation Mgmt (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
- Network reservation, monitoring, control
- All built on connectivity layer: GSI & IP

GridFTP: www.gridforum.org
GRAM, GRIS: www.globus.org

Collective layer

- Coordinating multiple resources
- Contains protocols and services that capture interactions among a collection of resources
- It supports a variety of sharing behaviours without placing new requirements on the resources being shared
- Sample services: directory services, co-allocation, brokering and scheduling services, data replication services, workload management services, collaboratory services

Collective Layer

- 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

Applications layer

- There are user applications that operate within the VO environment
- Applications are constructed by calling upon services defined at any layer
- Each of the layers are well defined using protocols, provide access to services
- Well-defined APIs also exist to work with these services

Services in the Web and the Grid
Web services

- Define a technique for describing software components to be accessed, methods for accessing these components, and discovery methods that enable the identification of relevant service providers
- A distributed computing technology (like CORBA, RMI…)
- They allow us to create loosely coupled client/server applications.

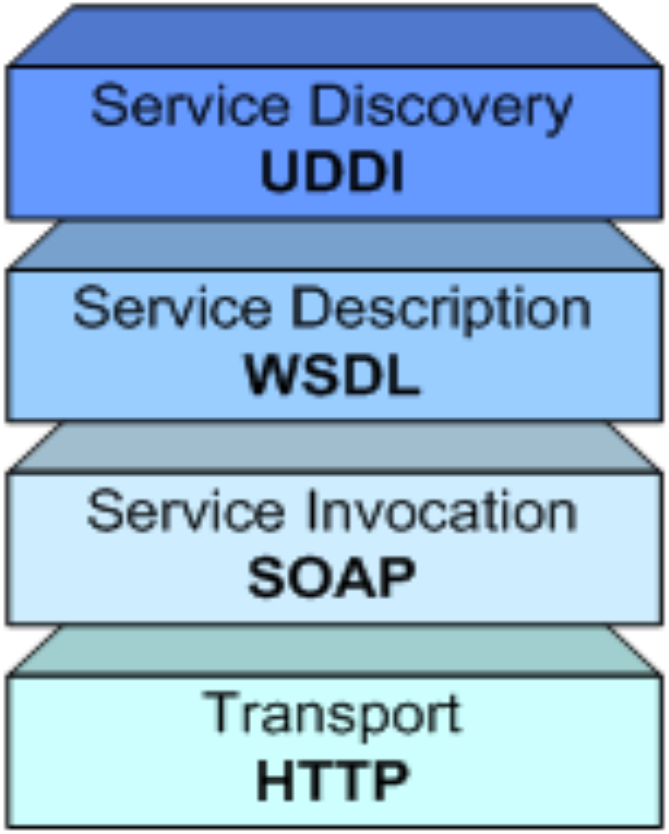
Services in the Web and the Grid
Web Services: Advantages

- Platform and language independent since they use XML language.
- Most use HTTP for transmitting messages (such as the service request and response)

Services in the Web and the Grid
Web Services: Disadvantages

- Overhead : Transmitting data in XML is not as convenient as binary codes.
- Lack of versatility: They allow very basic forms of service invocation (Grid services make up this versatility).
  - Stateless: *They can't remember what you have done from one invocation to another*
  - Non-transient: *They outlive all their clients.*

Services in the Web and the Grid
Web Services Architecture



Find Web services which meet certain requirements
(Universal Description, Discovery and Integration)

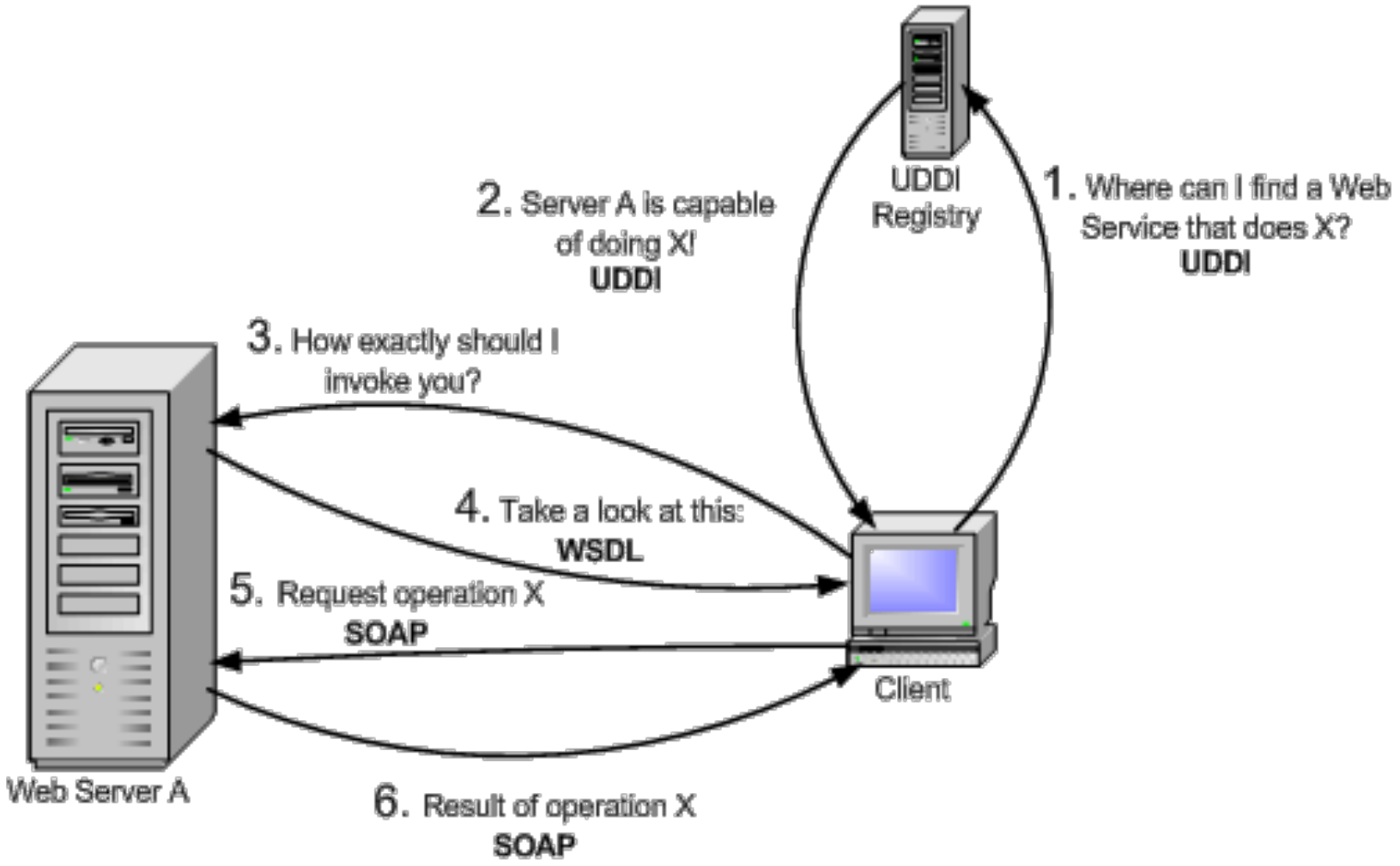
Services describe their own properties and methods
(Web Services Description Language)

Format of requests(client) and responses (server)
(Simple Object Access Protocol)

Message transfer protocol

(Hypertext Transfer Protocol)

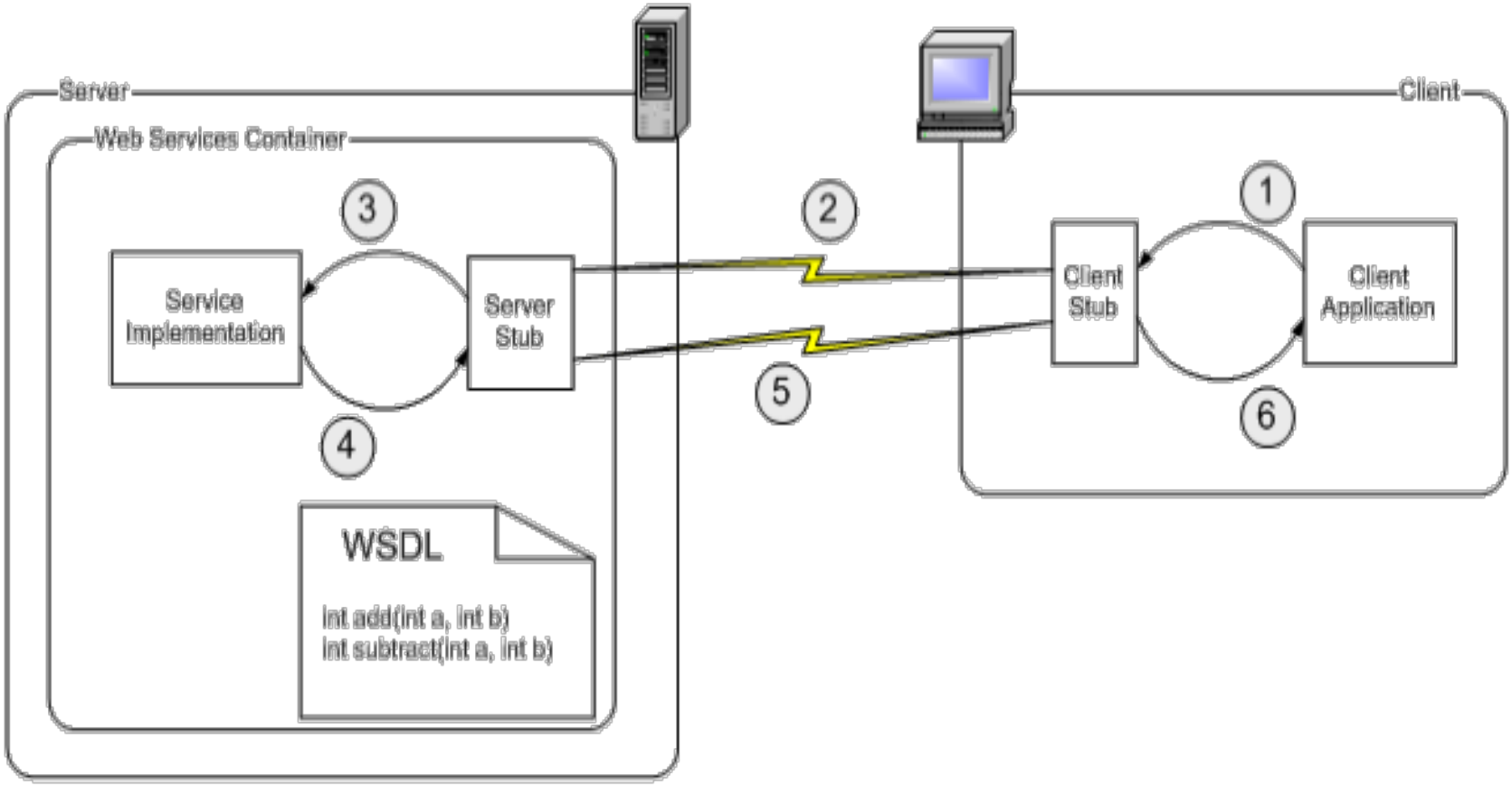
Services in the Web and the Grid
Invoking a Typical Web Service



Services in the Web and the Grid
Web Service Addressing

- URI: Uniform Resource Identifiers
- URI and URL are practically the same thing.
  - http://webservices.mysite.com/weather/us/WeatherService
- It can not be used with web browsers, it is meant for softwares.

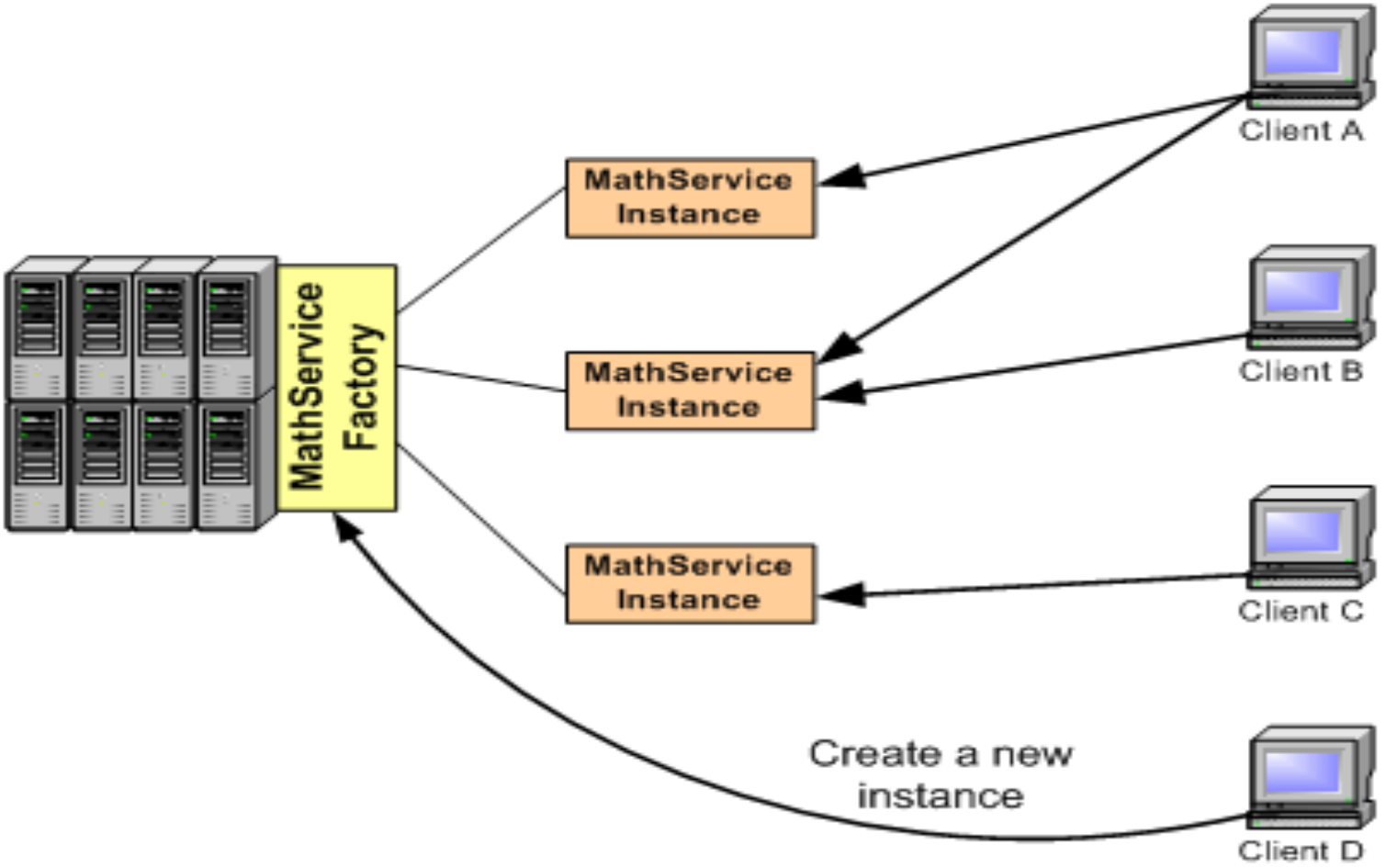
Services in the Web and the Grid
Web Service Application



What is a Grid Service?

- It provides a set of well defined interfaces and that follows specific conventions.
- It is a web service with improved characteristics and services.
  - Improvement:
    - Potentially Transient
    - Stateful
    - Delegation
    - Lifecycle management
    - Service Data
    - Notifications
- Examples : computational resources, programs, databases…

Services in the Web and the Grid Factories



Services in the Web and the Grid GSH & GSR

- GSH: Grid Service Handle (URI)
  - Unique
  - Shows the location of the service
- GSR: Grid Service Reference
  - Describes how to communicate with the service
  - As WS use SOAP, our GSR is a WSDL file.

Open Grid Services Architecture (OGSA)

- OGSA defines what Grid services are, what they should be capable of, what type of technologies they should be based on.
- OGSA does not give a technical and detailed specification. It uses WSDL.

Open Grid Services Infrastructure (OGSI)

- It is a formal and technical specification of the concepts described in OGSA.
  - The Globus Toolkit 3 is an implementation of OGSI.
- OGSI specification defines grid services and builds upon web services.

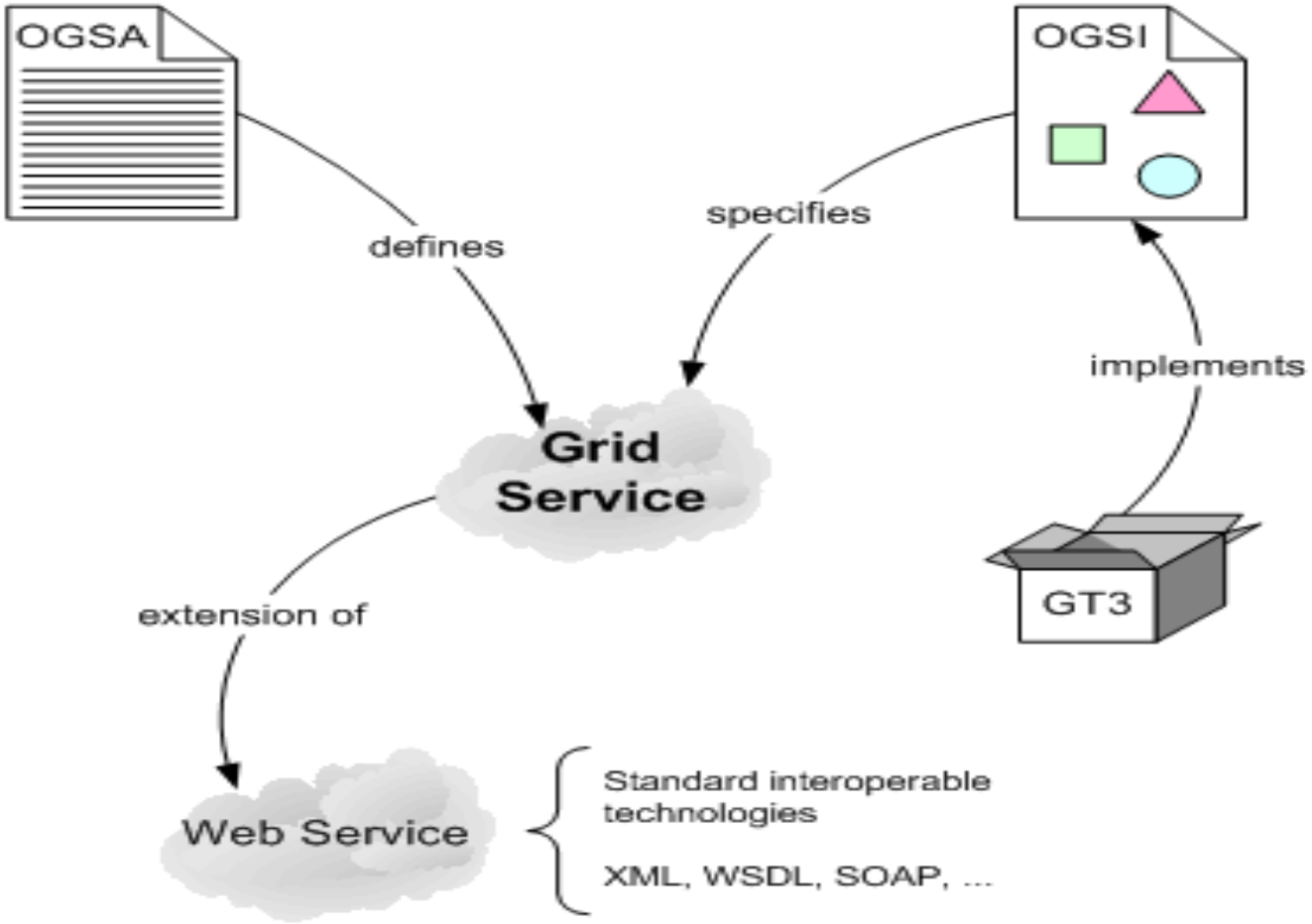
OGSI

- OGSI creates an extension model for WSDL called GWSDL (Grid WSDL). The reason is:
  - Interface inheritance
  - Service Data (for expressing state information)
- Components:
  - Lifecycle
  - State management
  - Service Groups
  - Factory
  - Notification
  - HandleMap

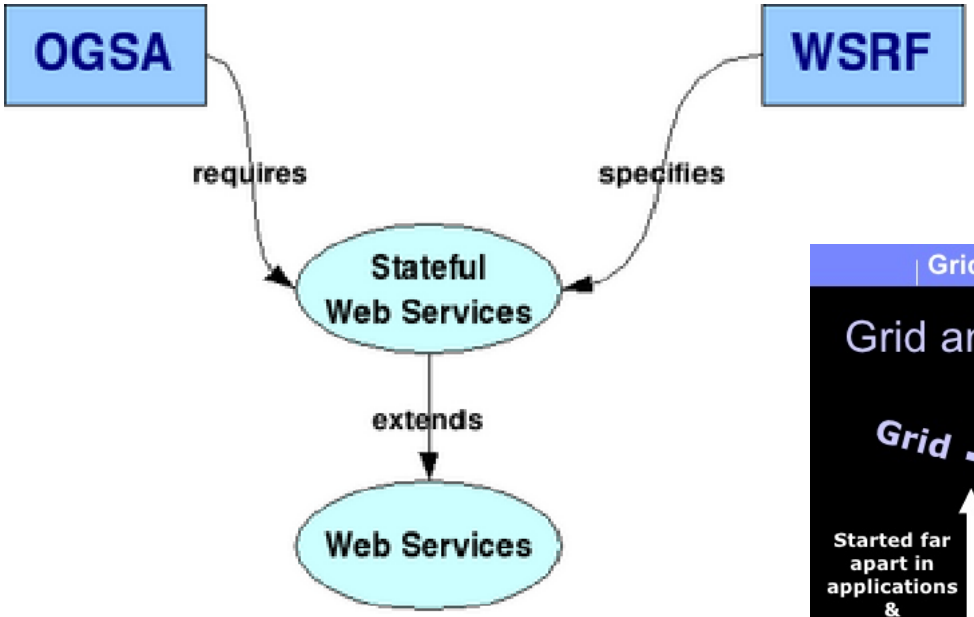
Services in the Web and the Grid Service Data Structure

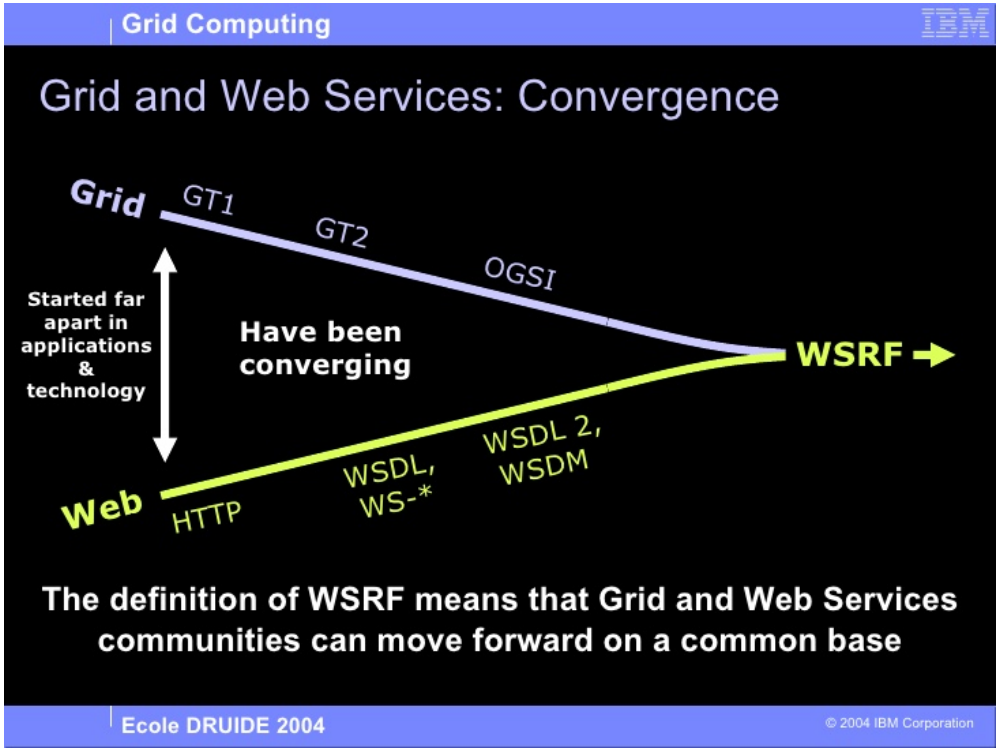
- <wsdl:definitions xmlns:tns="abc" targetNamespace="mynamespace">
- <gwsdl:portType name="AbstractSearchEngine">
- <wsdl:operation name="search" />
- .....
- <sd:serviceData name="cachedURL" type="tns: cachedURLType“ mutability="mutable" nillable="true", maxOccurs="1" minOccurs="0“ modifiable="true"/>
- </gwsdl:portType>
- </wsdl:definitions>
- nillable: allows the element to have no value
- modifiable: allows user override of the model element
- mutable: service data element can change

Services in the Web and the Grid OGSA, OGSI, GT3

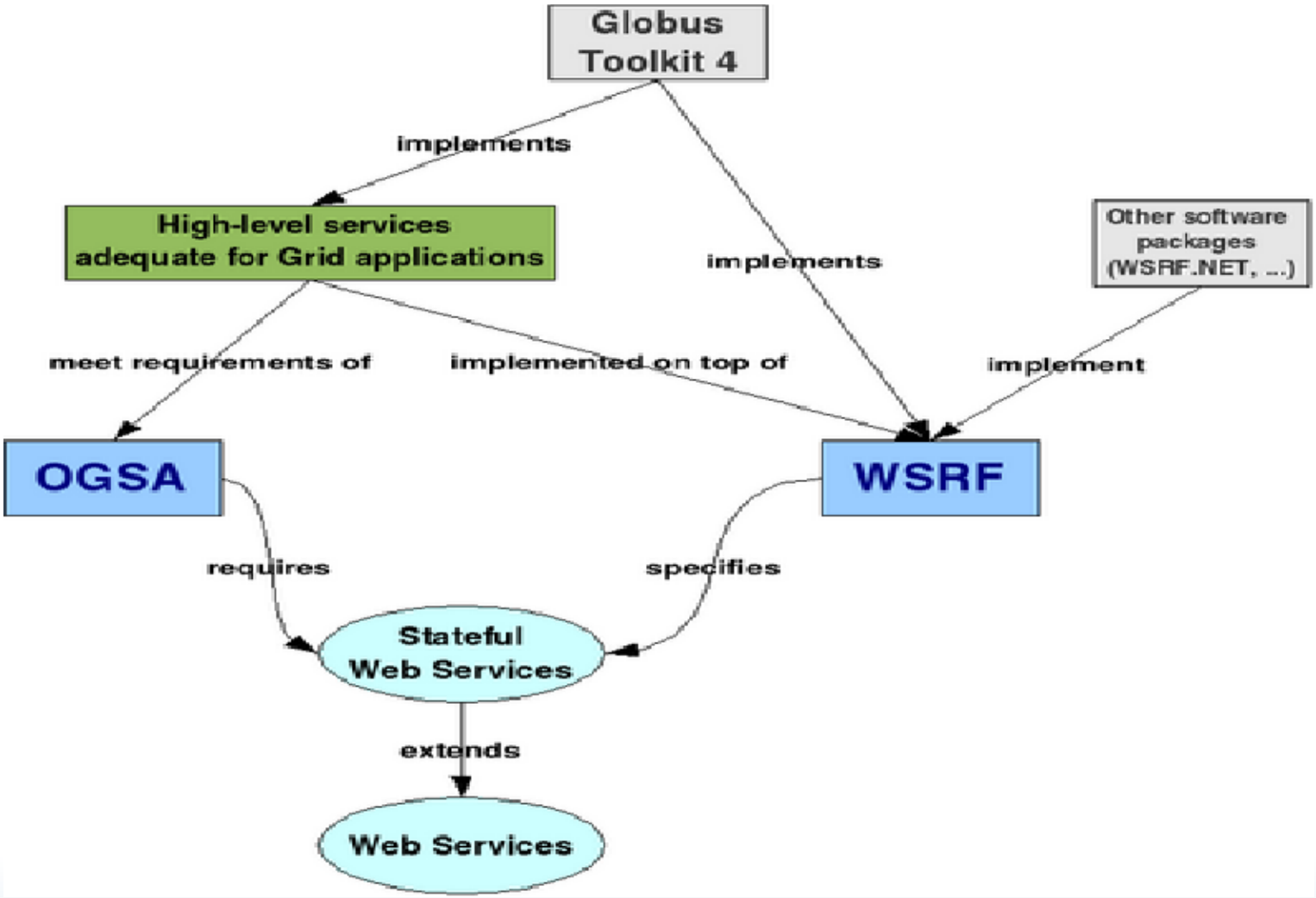


Services in the Web and the Grid OGSA, WSRF





Web services and the Grid
OGSA, WSRF, GT4

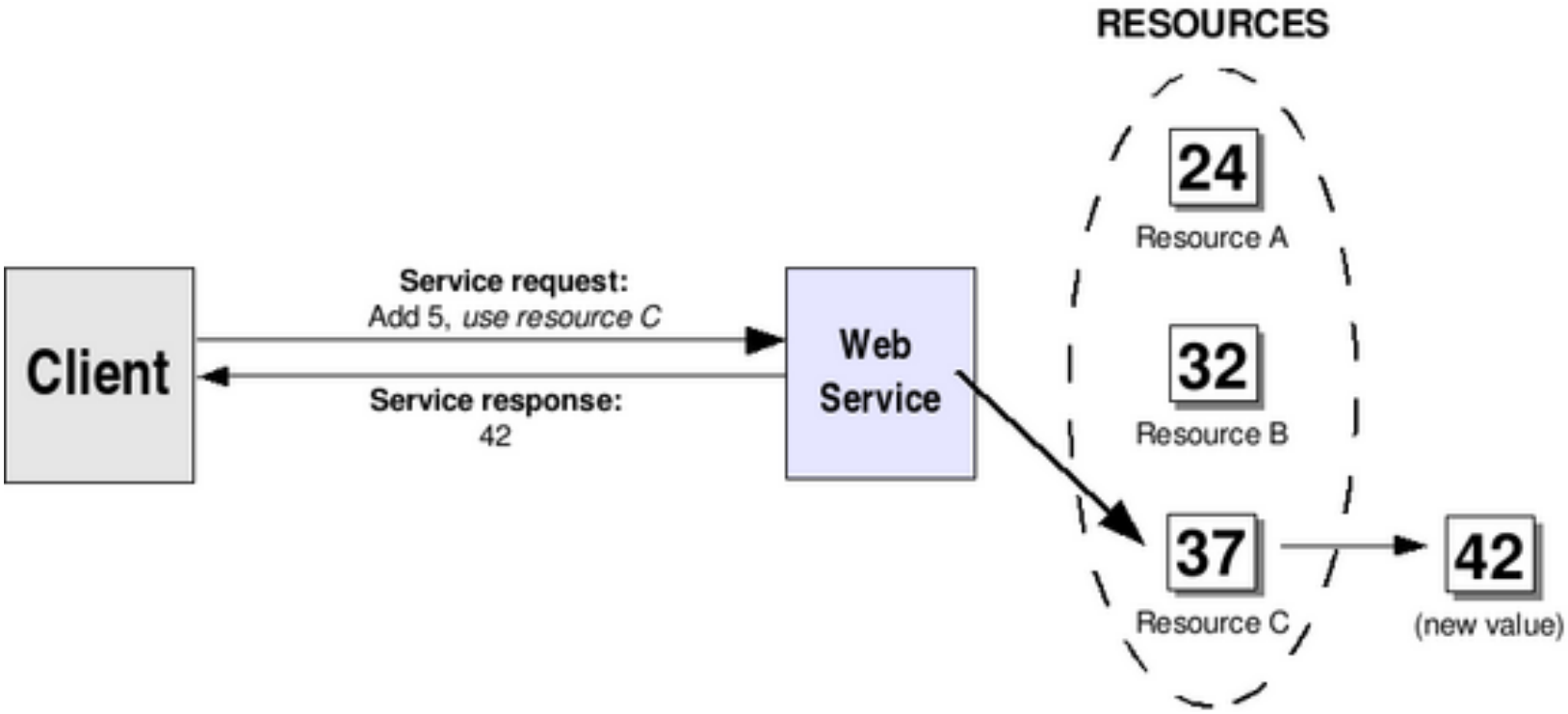


Web services and the Grid

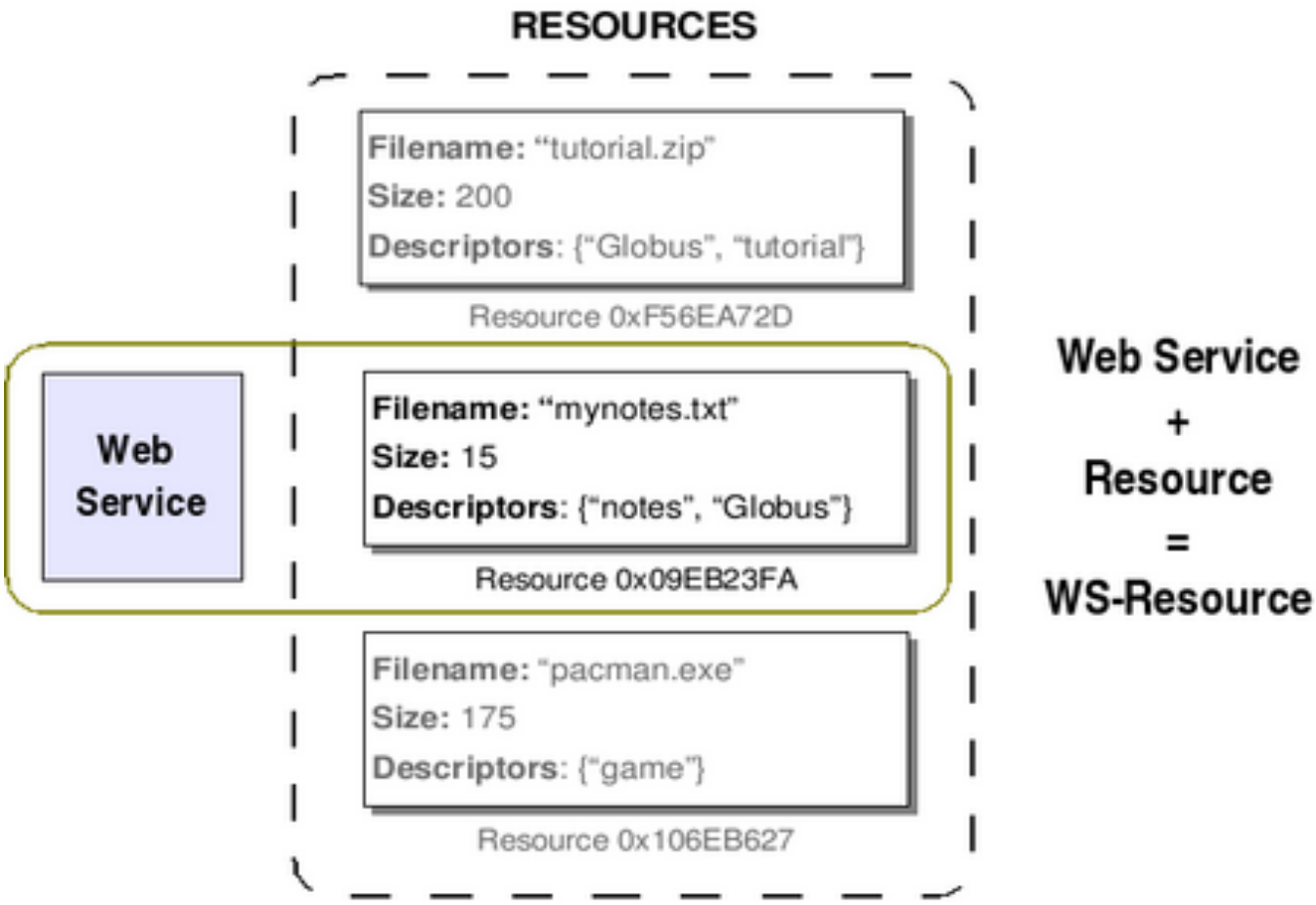
- GT4 replaced OGSI by WSRF (Web Service Resource Framework)
- Framework developed as a joint effort of W3C and OGF groups

How to model states using WS

- A resource is associated to each web service



WS-Resource


RESOURCES
Filename: "tutorial.zip"
Size: 200
Descriptors: {"Globus", "tutorial"}
Resource 0xF56EA72D
Web Service
Filename: "mynotes.txt"
Size: 15
Descriptors: {"notes", "Globus"}
Resource 0x09EB23FA
Filename: "pacman.exe"
Size: 175
Descriptors: {"game"}
Resource 0x106EB627
Web Service
+
Resource
=
WS-Resource

How to access a WS-Resource

- URI used to access the web service
- WS-Addressing used to access WS-Resource
- The address of a particular WS-Resource is called an endpoint reference in WS-Addressing lingo

WSRF

- a specification developed by OASIS http://www.oasis-open.org
- WSRF specifies how one can make Web Services stateful
- Differences between OGSI and WSRF:
  - http://www.globus.org/wsrf/specs/ogsi_to_wsrf_1.0.pdf

WSRF

- 5 normative WSRF specifications:
  - WS-ResourceProperties
  - WS-ResourceLifetime
  - WS-RenewableReferences
  - WS-ServiceGroup
  - WS-BaseFault
  - WS-Notification family of specifications

WSRF

- WS-ResourceProperties: properties of resources. For example, a resource can have values of different types (properties)

WSRF

- WS-ResourceLifetime: a WS-Resource can be destroyed, either synchronously with respect to a destroy request or through a mechanism offering time-based (scheduled) destruction, and specified resource properties [WS-ResourceProperties] may be used to inspect and monitor the lifetime of a WS-Resource

WSRF

- WS-RenewableReferences: a Web service endpoint reference (WS-Addressing) can be renewed in the event the addressing or policy information contained within it becomes invalid or stale

WSRF

- WS-ServiceGroup: heterogeneous by-reference collections of Web services can be defined, whether or not the services are WS-Resources (for example, one can dynamically add a new resource to a group of resources)

WSRF

- WS-BaseFault: fault reporting can be made more standardized through the use of an XML Schema type for base faults and rules for how this base fault type is used and extended by Web services

WSRF

- WS-Notification family of specifications: Standard approaches to notification of changes

WSDL

- Types– a container for data type definitions using some type system (such as XSD).
- Message– an abstract, typed definition of the data being communicated.
- Operation– an abstract description of an action supported by the service.
- Port Type–an abstract set of operations supported by one or more endpoints.
- Binding– a concrete protocol and data format specification for a particular port type.
- Port– a single endpoint defined as a combination of a binding and a network address.
- Service– a collection of related endpoints.

Creation of a stateful web service

- MathService to perform operations:
  - Addition
  - Subtraction
- Have the ResourceProperties (RP):
  - Value (integer)
  - Last operation performed (string)
- Extra operation Get to get Value RP
- Once a new resource is created:
  - Value is set to zero
  - Last operation is set to “NONE”

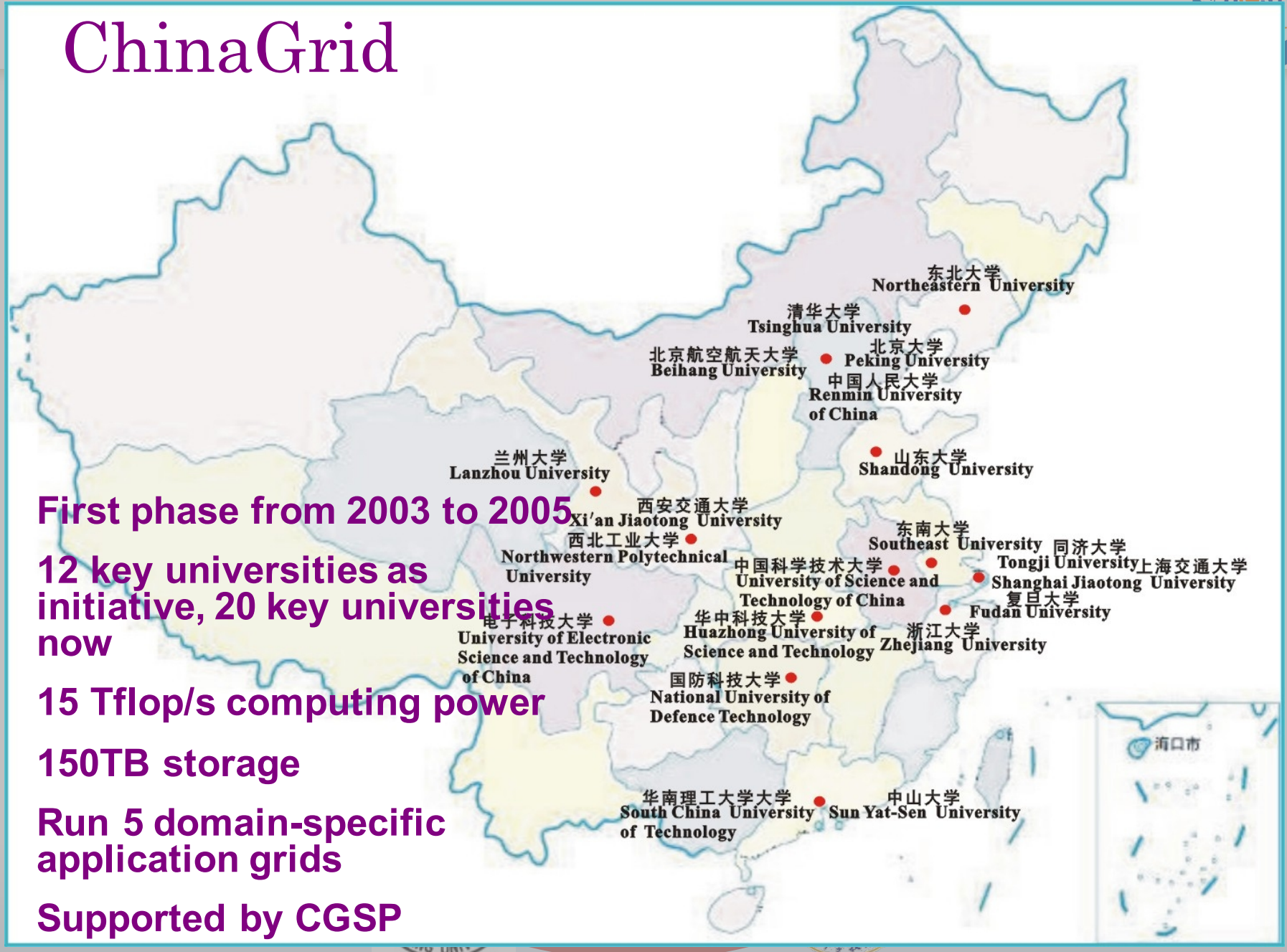
5 steps

- Define the service's interface. This is done with WSDL
- Implement the service. This is done with Java.
- Define the deployment parameters. This is done with WSDD and JNDI
- Compile everything and generate a GAR file. This is done with Ant
- Deploy service. This is also done with a GT4 tool

JNDI: Java Naming and Directory Interface

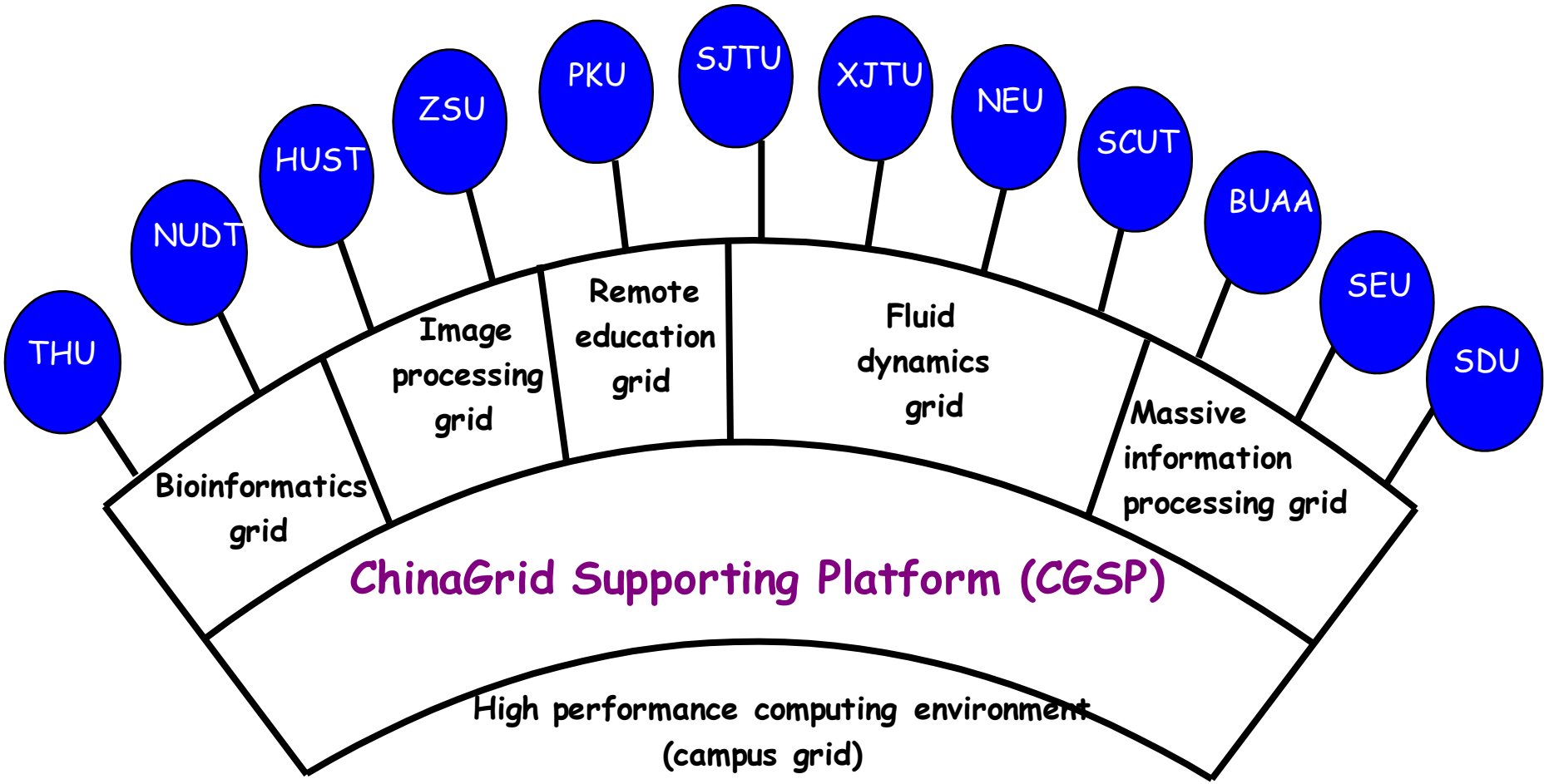
WSDD: Web Service Deployment Descriptor

ChinaGrid

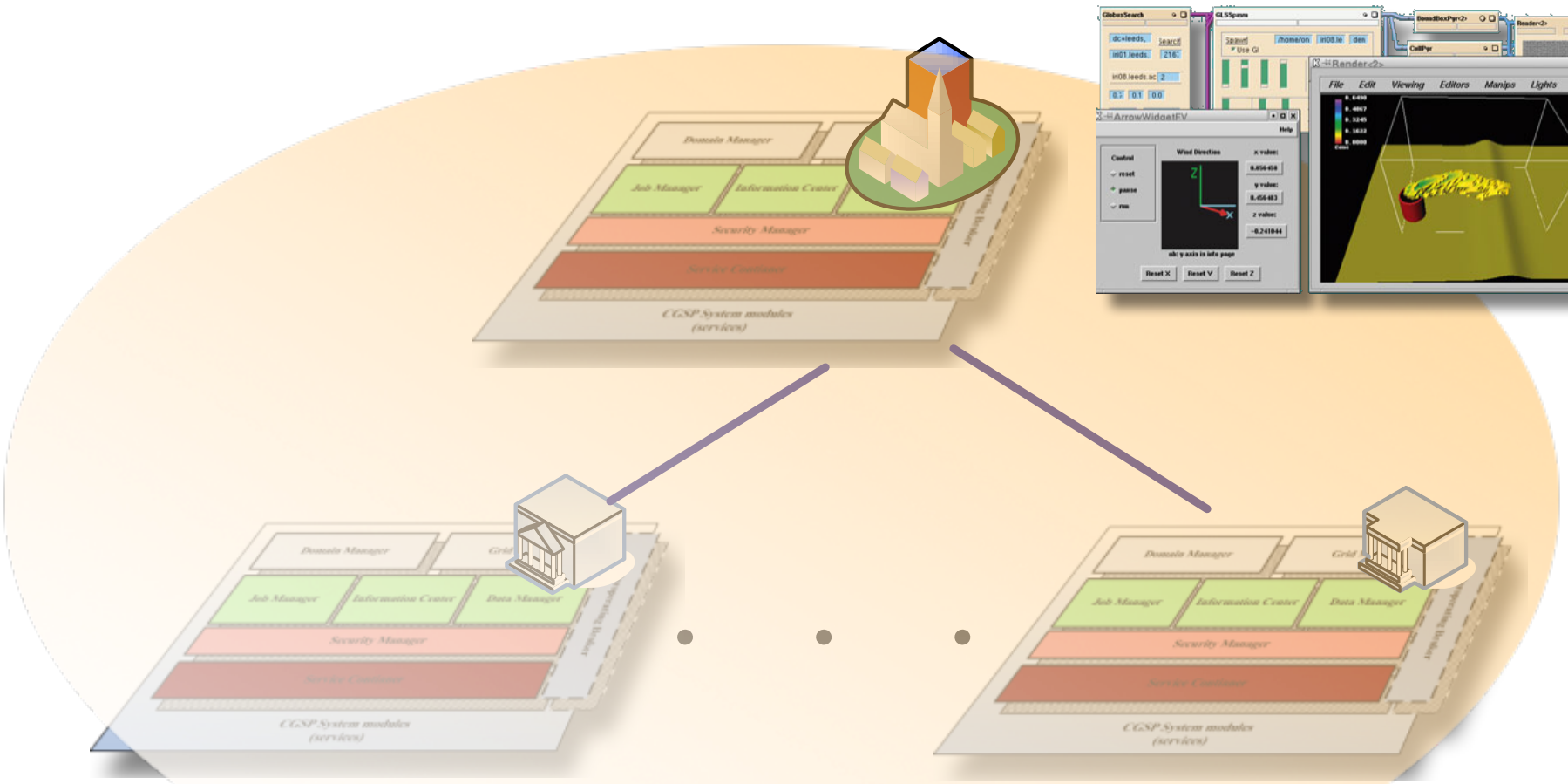


- First phase from 2003 to 2005
- 12 key universities as initiative, 20 key universities now
- 15 Tflop/s computing power
- 150TB storage
- Run 5 domain-specific application grids
- Supported by CGSP

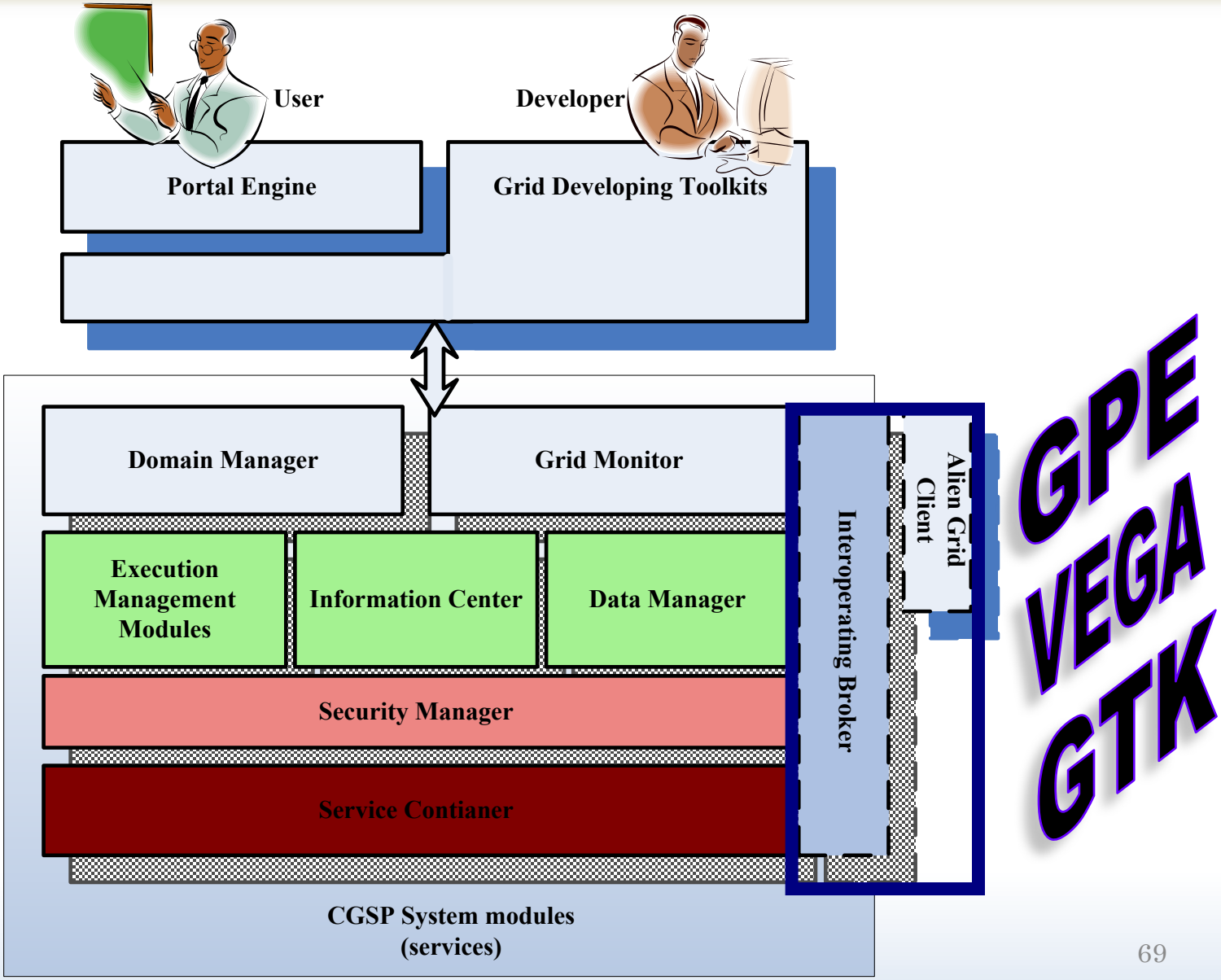
Layered Infrastructure of ChinaGrid



CGSP Architecture



CGSP Overview



Thanks

Are Grids a Solution?

"Grid Computing" means different things to different people.

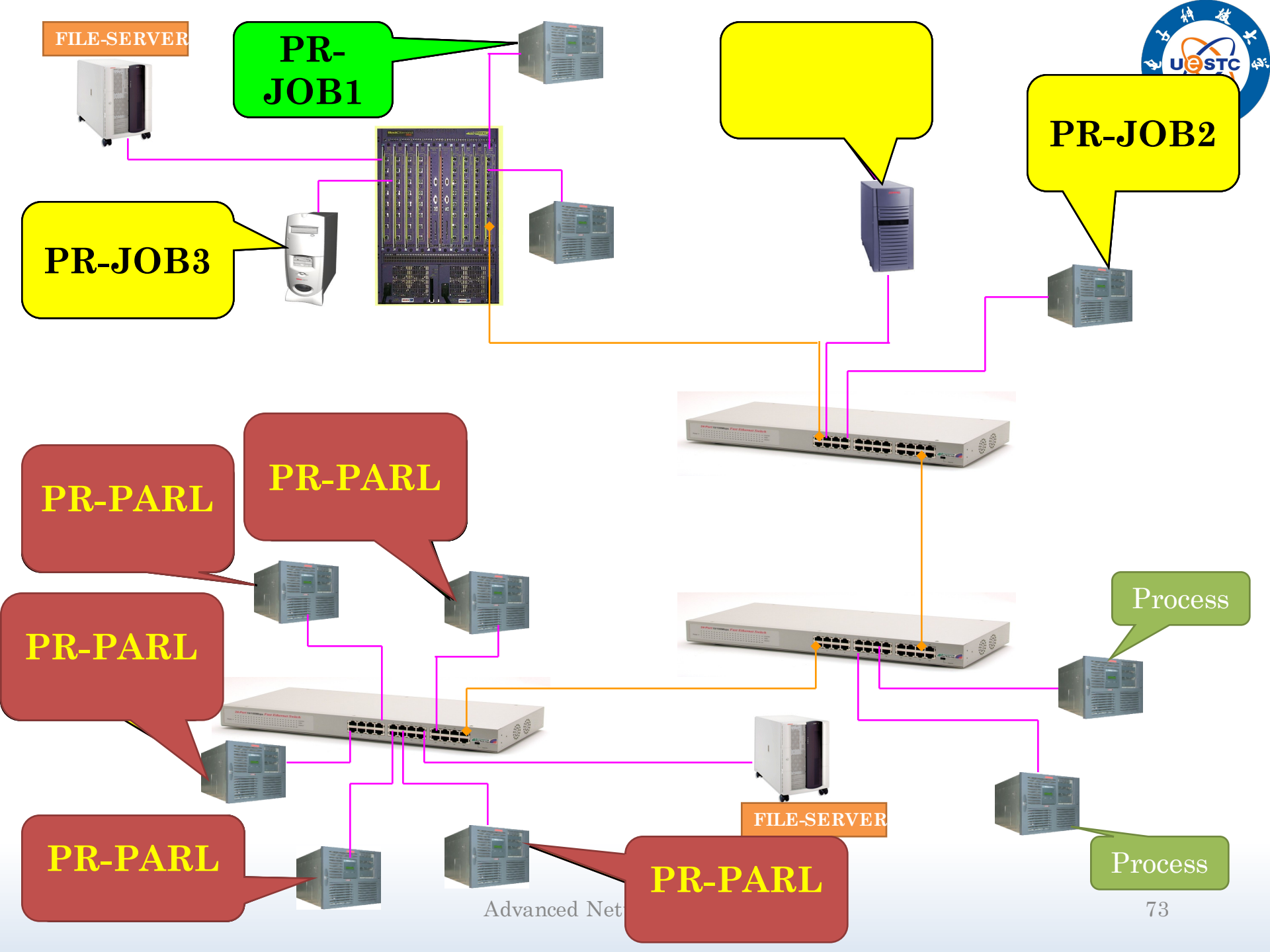
- Goals
- Reduce computing costs
- Increase computing resources
- Reduce job turnaround time
- Enable parametric analyses
- Reduce Complexity to Users
- Increase Productivity

- Technology Issues
- Clusters
- Internet infrastructure
- MPP solver adoption
- Administration of desktop
- Use middleware to automate
- Virtual Computing Centre

"Dependable, consistent, pervasive access to resources"

Why Migrate Processes ?

- Load balancing
  - Reduce average response time
  - Speed up individual jobs
  - Gain higher throughput
- Move process closer to its resources
  - Use resources effectively
  - Reduce network traffic
- Increase systems reliability
- Move process to a machine
  - Holding confidential data


FILE-SERVER
PR-JOB1
PR-JOB2
PR-JOB3
PR-PARL
PR-PARL
PR-PARL
PR-PARL
PR-PARL
Process
Process
FILE-SERVER

What does the Grid do for you?

- You submit your work
- And the Grid
  - Finds convenient places for it to be run
  - Organises efficient access to your data
    - Caching, migration, replication
  - Deals with authentication to the different sites that you will be using
  - Interfaces to local site resource allocation mechanisms, policies
  - Runs your jobs, Monitors progress, Recovers from problems,Tells you when your work is complete
- If there is scope for parallelism, it can also decompose your work into convenient execution units based on the available resources, data distribution

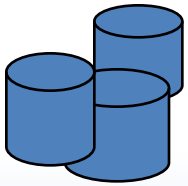
Main components

User Interface (UI): The place where users logon to the Grid

Resource Broker (RB): Matches the user requirements with the available resources on the Grid

Information System: Characteristics and status of CE and SE (Uses “GLUE schema”)

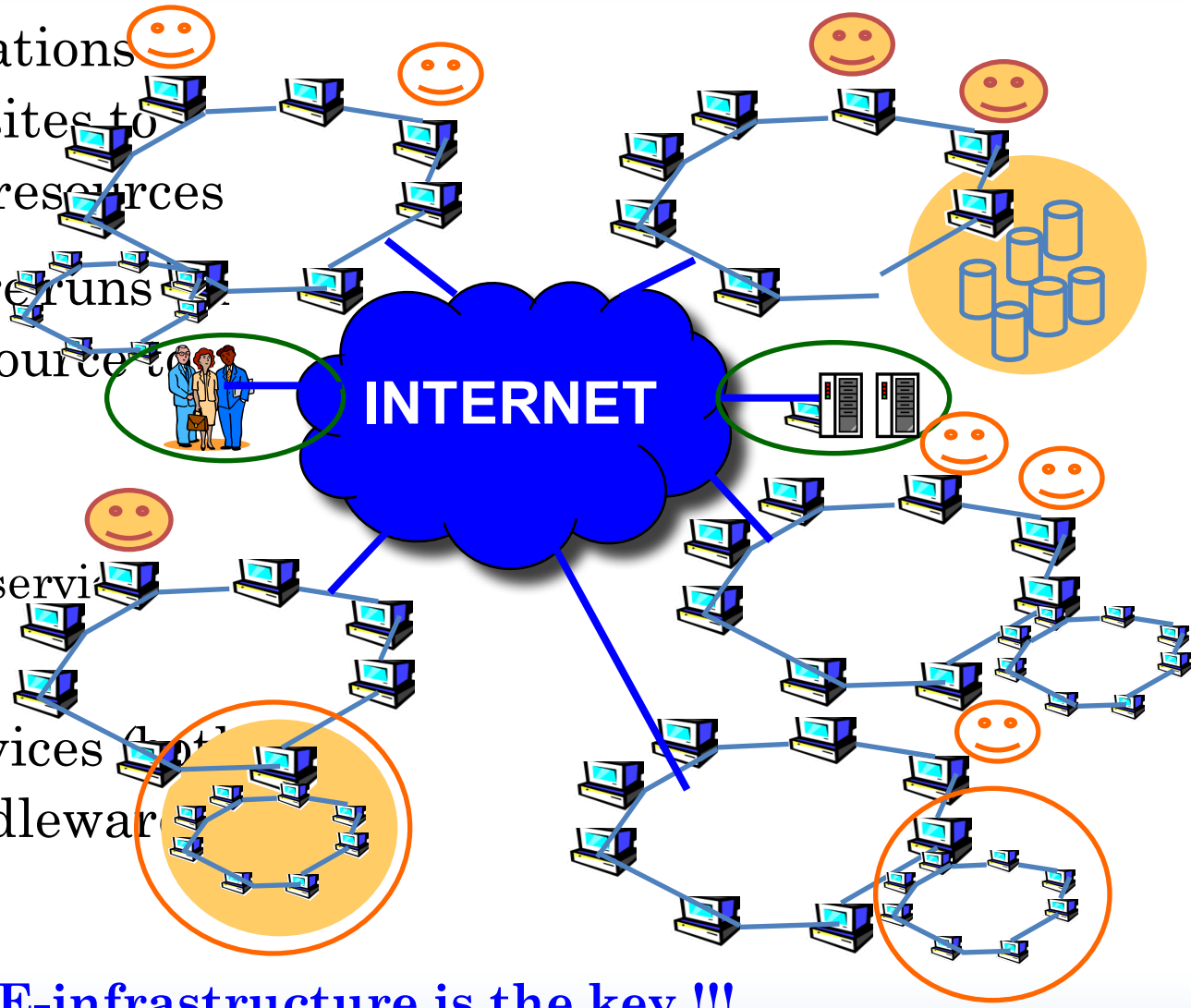
Computing Element (CE): A batch queue on a site’s computers where the user’s job is executed



Storage Element (SE): provides (large-scale) storage for files

Typical current grid

- Virtual organisations negotiate with sites to agree access to resources
- Grid middleware runs on each shared resource to provide
  - Data services
  - Computation services
  - Single sign-on
- Distributed services (both people and middleware) enable the grid



E-infrastructure is the key !!!