

# Open Grid: Current Status and Future Directions

Ian Foster

Argonne National Laboratory

University of Chicago

Globus Alliance

## Abstract

In February 2002, the Globus Alliance and IBM introduced the Open Grid Services Architecture (OGSA) and Infrastructure (OGSI), to begin the process of building out a standard set of Grid protocols layered on Web Services. Subsequent concerted effort within the Global Grid Forum culminated in the delivery of the OGSI v1.0 specification in July 2003. Yet while tremendous progress has been made on the acceptance of Web Services and OGSI as the basis for the Grid, there remains some confusion about the relationship of "Grid Services" as defined by the OGSI v1.0 specification and "Web Services". I explain what the Grid is, the relationship of OGSA and OGSI v1.0 to Web Services, the evolution of OGSI to better exploit emerging Web Services standards, requirements Grid is placing on those emerging Web Services standards, and the landscape of protocols that are being defined upon Web Services to meet Grid requirements.

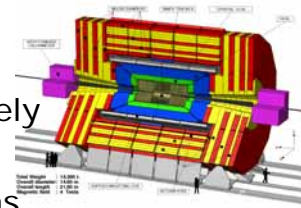
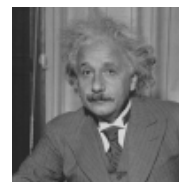
## Overview

- What is Grid anyway?
- Grid meets Web services
- The power of open source
- Grid in practice
- Summary

3

## Why the Grid? Origins: Revolution in Science

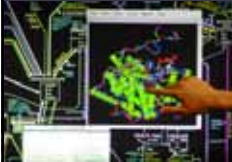
- Pre-Internet
  - Theorize &/or experiment, alone or in small teams; publish paper
- Post-Internet
  - Construct and mine large databases of observational or simulation data
  - Develop simulations & analyses
  - Access specialized devices remotely
  - Exchange information within distributed multidisciplinary teams



4

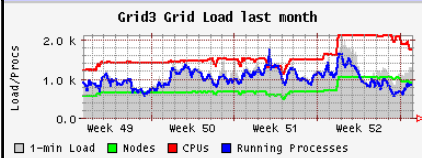
## Origins in Science

Engage via telepresence in an experiment at a remote facility



Discover & access a genome analysis service (running on high-end computer)

Integrate data from multiple sources in support of global change research



Harness computers across sites to process data from a physics experiment

the globus alliance  
www.globus.org

## Why the Grid? New Driver: Revolution in Business

- Pre-Internet
  - Central data processing facility
- Post-Internet
  - Enterprise computing is highly distributed, heterogeneous, inter-enterprise (B2B)
  - Business processes increasingly computing- & data-rich
  - Outsourcing becomes feasible → service providers of various sorts
  - Growing complexity & need for more efficient management



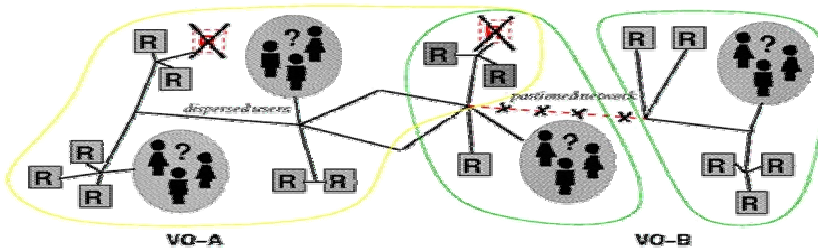
## Common Requirements

- Dynamically link resources/services
  - From collaborators, customers, eUtilities, ...  
(members of evolving “virtual organization”)
- Into a “virtual computing system”
  - Dynamic, multi-faceted system spanning institutions and industries
  - Configured to meet instantaneous needs, for:
- Multi-faceted QoX for demanding workloads
  - Security, performance, reliability, ...

7

## The Grid

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



1. Enable integration of distributed resources
2. Using general-purpose protocols & infrastructure
3. To achieve better-than-best-effort service

8



## Overview

- What is Grid anyway?
- **Grid meets Web services**
- The power of open source
- Grid in practice
- Summary

9

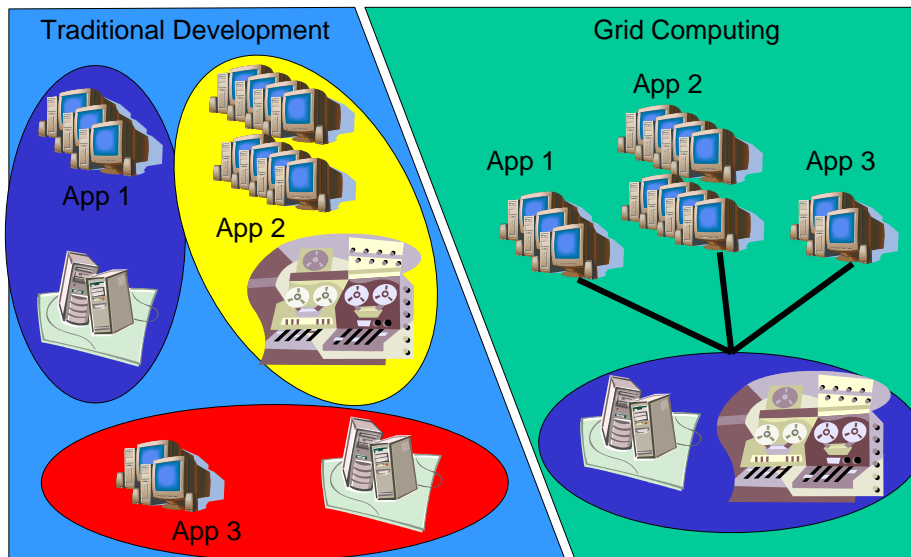


## Key Grid Requirements (Financial Services Industry)

- Standards
- Heterogeneous distributed resources
- Silo avoidance
- Virtualization
- Service Level Agreements (SLAs)
- Dynamic adaptation to demand
- Performance
- Management

10

## Silo Avoidance

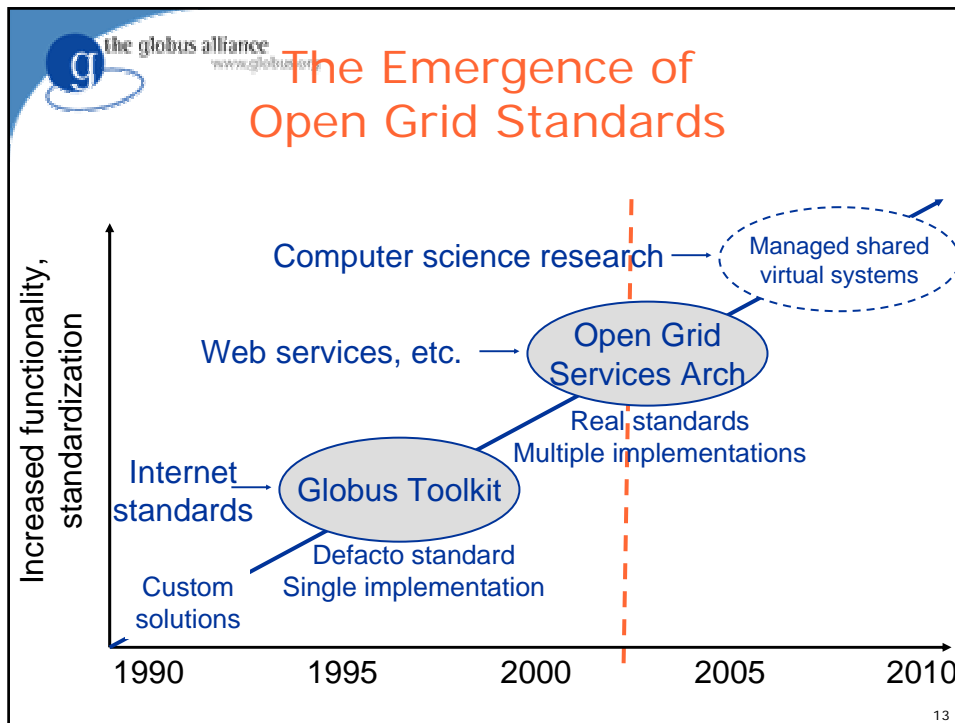



11

## Silo Avoidance

- Various sources of silos
  - Resource classes: Computers vs. data vs. ...
  - Architectures: IA32 vs. Sun cluster vs. ...
  - Operating systems: Linux vs. Windows vs. ...
  - Services: Relational databases vs. filesystems
  - Applications: May require specific silos
- Keys to silo avoidance
  - Virtualization: Service Oriented Infrastructure
  - Frameworks ...
  - Standards: Interoperability, re-use, and component replacement

12



 the globus alliance  
www.globus.org

## Open Grid Services Architecture

- Define a service-oriented architecture ...
  - the key to effective virtualization
- ... to address vital "Grid" requirements
  - AKA utility, on-demand, system management, collaborative computing
- ... building on Web services standards
  - extending those standards where needed
  - enabling ubiquity and thus a rich ecology of platform and solution providers

14



## Web Services as a Foundation

- Provide
  - Interface definition & service invocation
  - Nice semantics & extensibility
  - First steps towards security, workflow, etc.
  - Considerable commercial adoption
- But not
  - A silver bullet
  - A complete solution
  - Fixed in stone

15



## Core Web Services Specs Provide Important Infrastructure

- Web Services Description Language (WSDL)
- Simple Object Access Protocol (SOAP)
- Universal Description, Discovery and Integration (UDDI)
- WS-Security
- WS-ReliableMessaging
- WS-Addressing
- Others

16



## Web Service Description Language

- XML-based language for abstractly describing message exchanges between clients, services
  - Types defined using XML Schema
  - Message comprising one or more parts of XML Schema types/elements
  - Operation = input/output or input only msgs
  - Interface = named group of operations
- Binding the interfaces to concrete protocols
  - E.g. SOAP/HTTP
- It says nothing about what messages are sent

17



## WS Core Enables Frameworks: E.g., Resource Management

Applications of the framework  
(Compute, network, storage provisioning,  
job reservation & submission, data management,  
application service QoS, ...)

WS-Agreement  
(Agreement negotiation)

WS Distributed Management  
(Lifecycle, monitoring, ...)

WS-Resource Framework & WS-Notification  
(Resource identity, lifetime, inspection, subscription, ...)

Web services  
(WSDL, SOAP, WS-Security, WS-ReliableMessaging, ...)

18



## Web Services and Stateful Resources

- “State” appears in almost all applications
  - Data in a purchase order
  - Current usage agreement for resources
  - Metrics associated with work load on a server
- Web services can model, access and manage state in many different ways
  - Ad-hoc, per-application approaches
  - OGSI/WSRF proposes a standard approach

19



## Why Standardize An Approach?

- Building systems by composition of heterogeneous components demands that we standardize common patterns
  - Approach to resource identification
  - Resource lifetime management interfaces
  - Resource inspection & monitoring interfaces
  - Base fault representation
  - Service and resource groups
  - Notification
  - And many more...
- Standards encourage tooling & code re-use
  - Build services more quickly & reliably

20



## Standardizing State

- Open Grid Services Infrastructure (OGSI)
  - Global Grid Forum (2002), completed June 2003 with contributions from many
  - Multiple implementations in various stages
- WS Resource Framework (WSRF)
  - Refactor OGSI to decompose into separate standards, use WS-Addressing, WS-Resource
  - + WS-Notification (c.f. WS-Eventing)
  - OASIS (2004): Fujitsu, Globus, HP, IBM, etc.
  - WSRF interoperability event April 2004
  - All the same stuff: impact on users minor

21



## Core Ideas in WSRF

- Separate service from resource
  - Service is static and stateless
  - Resource is dynamic and stateful
- Leverage WS-Addressing
- Make WS-I compliant
  - But WS-I alone does not make problems go away, still need to worry about how to manage lifetime, naming, state, ...
- Preserve OGSI functionality
  - Lifetime, state publication, notification, error types

22

## WSRF & WS-Notification Overview

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

23

## Agreement Negotiation

- All interesting interactions will be based on *agreements* between requestors & services
  - Encode a negotiated quality of experience
- Challenge is to establish the framework within which agreements can be:
  - Established in an oversubscribed environment
  - Transformed, composed, decomposed
  - Managed like any other resource
  - Evolved as a result of faults
- WS-Agreement is the next step on this path

24



## WS-Agreement Applicability

- All interesting Web/Grid services interactions will be governed by agreements
- WS-Agreement to be used/specialized by specs that define domain-specific services
  - Data management services
    - > Storage provisioning, transfer mgt, replication, etc.
  - Job/execution management services
    - > Cluster provisioning, image/service deployment, job reservation and submission, specialized services, etc.
  - Network provisioning services
    - > Optical path, firewall traversal, etc.
  - Application- and domain-specific services

25



## WS Distributed Management

- OASIS Web Services Distributed Management (WSDM) technical committee
  - Management using Web Services
  - HP submitted its Web Services Management Framework (WSMF) to WSDM in July 2003
- Framework that builds on WSRF to provide (among other things):
  - Service lifecycle (state) models
  - Service monitoring abstractions
  - Service relationships

26



the globus alliance  
www.globus.org

## Grid Requires that We Define WS-Based Frameworks

Applications of the framework  
(Compute, network, storage provisioning,  
job reservation & submission, data management,  
application service QoS, ...)

WS-Agreement  
(Agreement negotiation)

WS Distributed Management  
(Lifecycle, monitoring, ...)

WS-Resource Framework & WS-Notification  
(Resource identity, lifetime, inspection, subscription, ...)

Web services  
(WSDL, SOAP, WS-Security, WS-ReliableMessaging, ...)

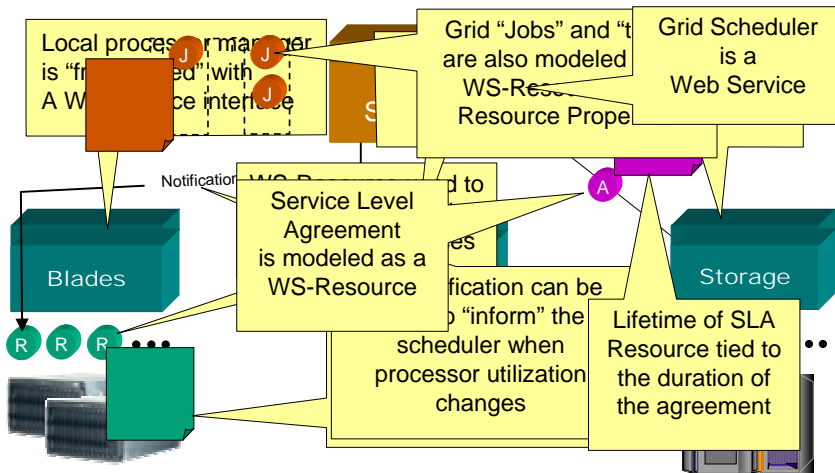
27



the globus alliance  
www.globus.org

## Bringing it All Together

*Scenario: Resource management & scheduling*



28

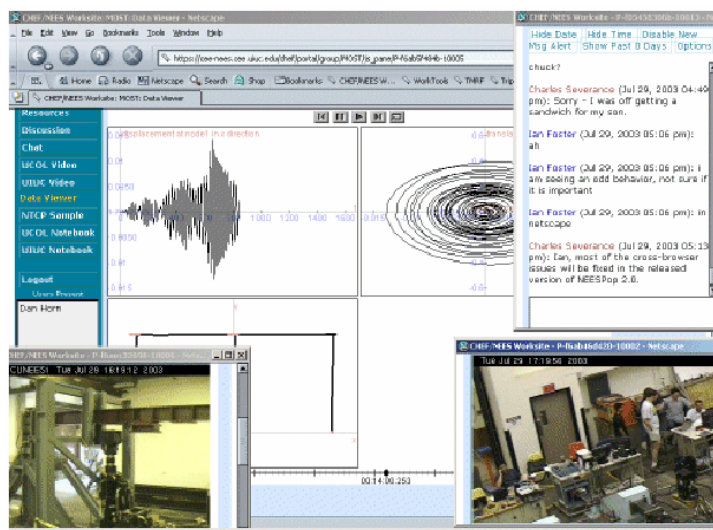
## Overview

- What is Grid anyway?
- Grid meets Web services
- **The power of open source**
- Grid in practice
- Summary

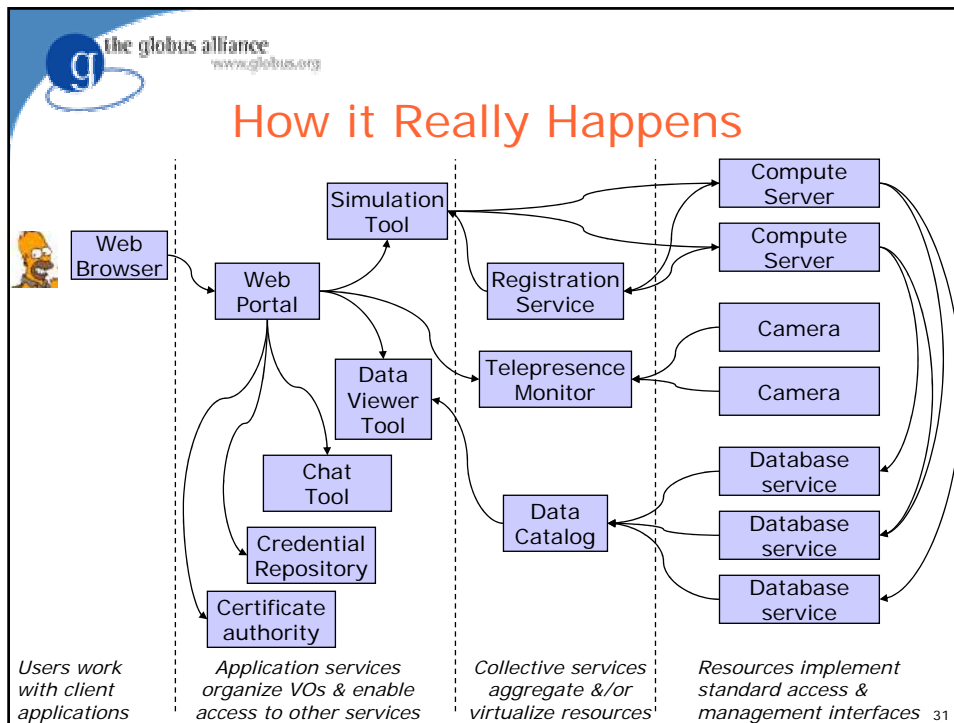
29

## What End Users Want

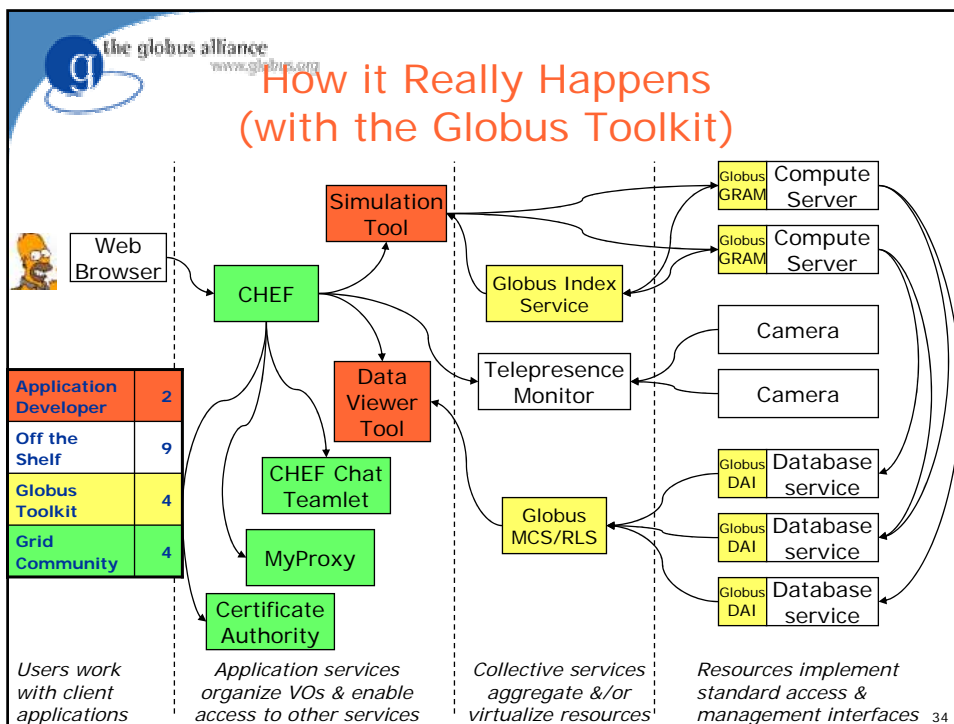
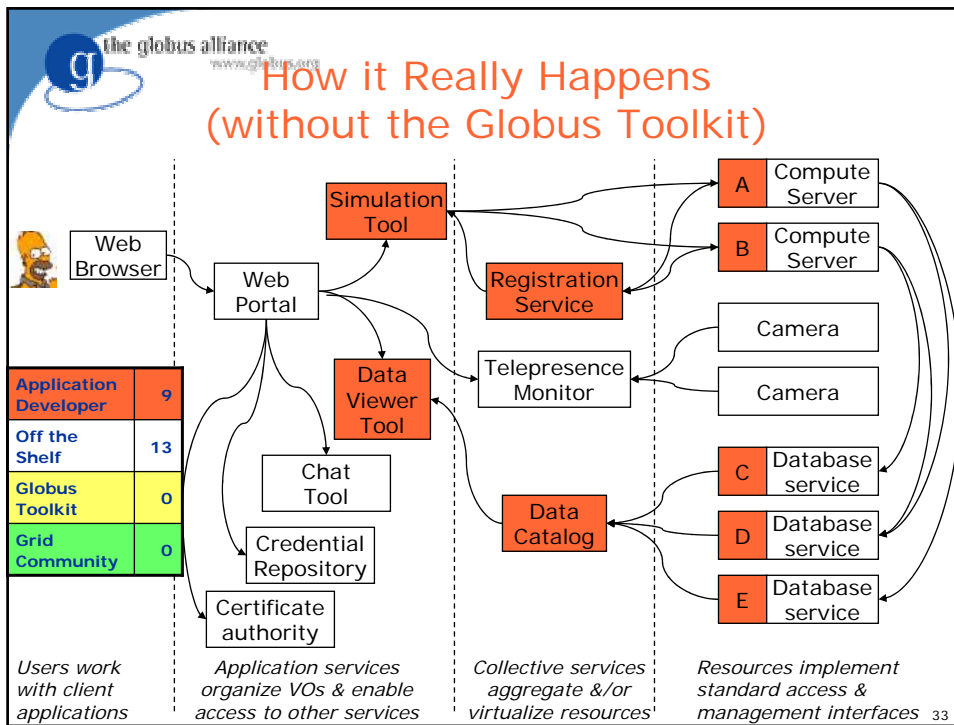
Secure, reliable, on-demand access to data, software, people, and other resources (ideally all via a Web Browser!)



30



- the globus alliance  
www.globus.org
- ## How it Really Happens
- Implementations are provided by a mix of
    - Application-specific code
    - “Off the shelf” tools and services
    - Tools and services from the Globus Toolkit
    - GT-compatible tools and services from the Grid community
  - Glued together by ...
    - Application development
    - System integration
  - Deployed and supported
- 32



## The Open Source Globus Toolkit

### Grid infrastructure

- OGSI/WSRF mechanisms
- Grid security infrastructure

### + A wide range of services

- GRAM resource management, data management, registries, etc., etc.
- WS and pre-WS versions both supported

### + Rich ecosystem of supporting technologies

- Security, tools, vertical solutions

### ➔ Enabler of many different capabilities

35

## Globus Toolkit >> WSRF/OGSI

- WSRF/OGSI impl is just one small part of GT
  - GT = Services for execution mgt, data mgt, monitoring, and discovery
  - E.g., GRAM, GridFTP, RFT, RLS, OGSA-DAI
- GT services are largely unaffected by WSRF
  - Translate dynamic service to dynamic resource
  - Interfaces and semantics largely unchanged
  - But not interoperable due to protocol changes
- Pre-WS components continue to be supported
  - GridFTP, RLS, pre-WS GRAM

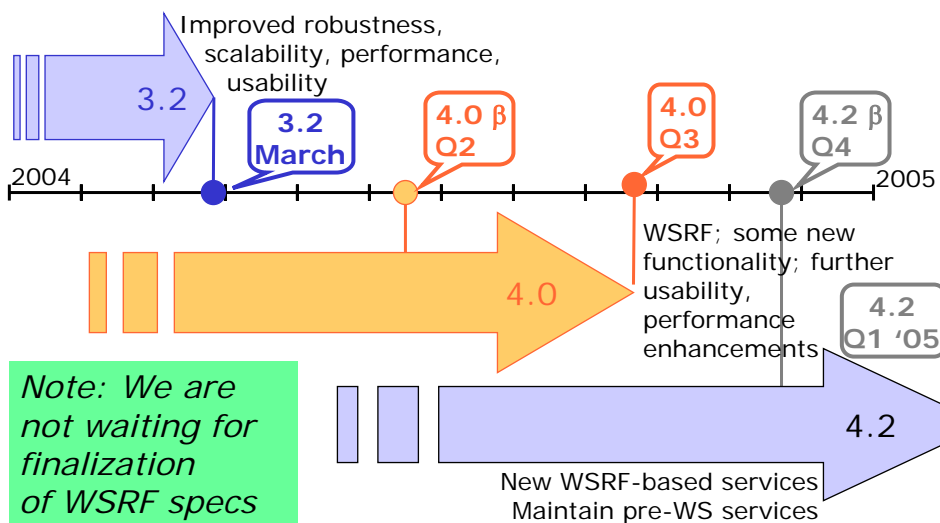
36

## Globus Priorities for 2004

- Stabilize Web services-based implementation
  - Greatly improve usability, performance, reliability, testing, scalability, documentation,
  - WSRF support
  - Leverage Apache even more: Axis, WS-Security, WS-Addressing, ...
- Bring to fruition new functionality in pipeline
  - Data access & integration (tech preview)
  - Enhanced GridFTP
  - Community scheduling framework (contribution)
  - Monitoring & discovery capabilities
- GT2→GT3 bump >> GT3→GT4 bump

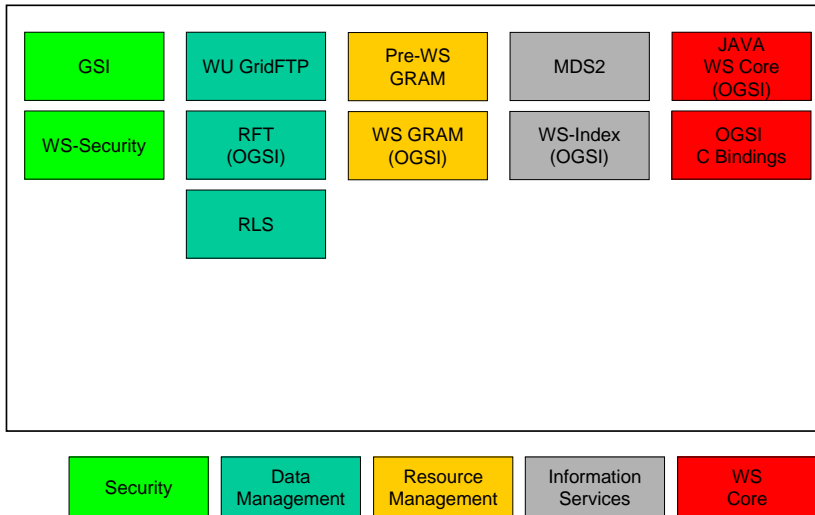
37

## GT Release Schedule



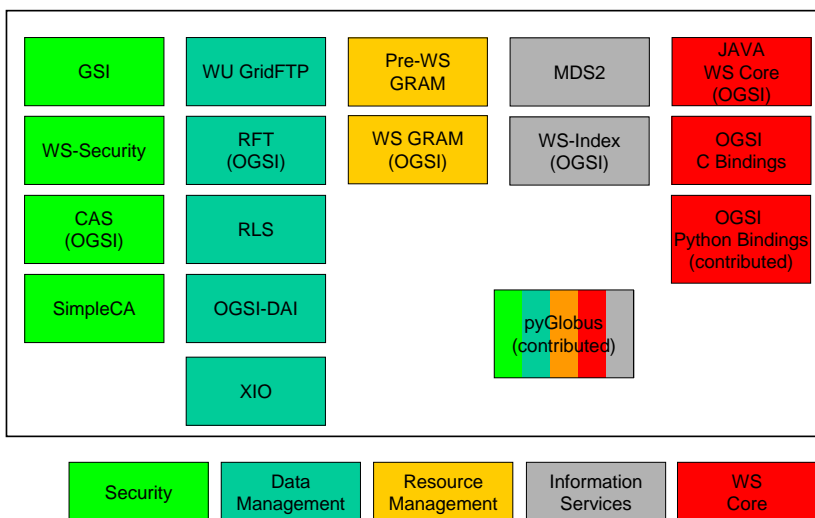
38

## Components in Globus Toolkit 3.0



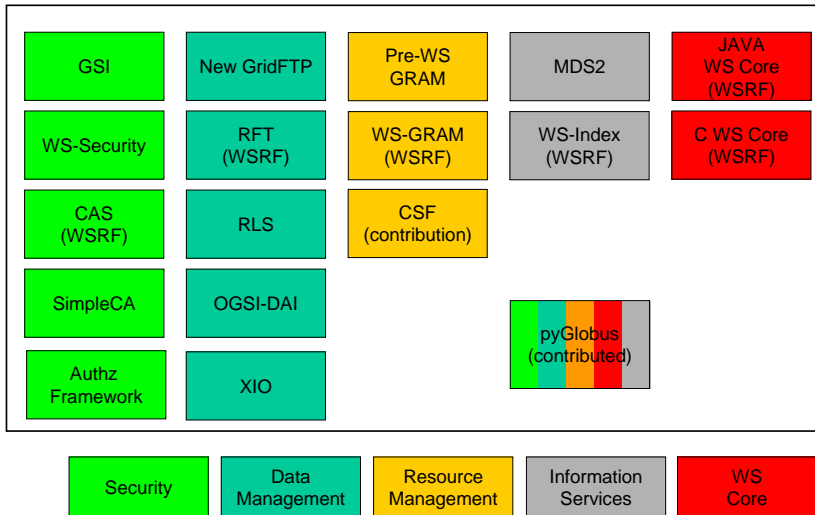
39

## Components in Globus Toolkit 3.2



40

## Planned Components in GT 4.0



41

## Overview

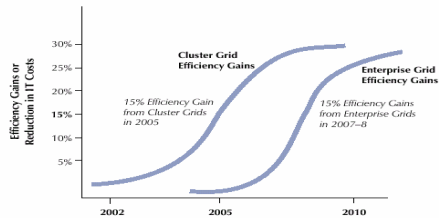
- What is Grid anyway?
- Grid meets Web services
- The power of open source
- **Grid in practice**
- Summary

42

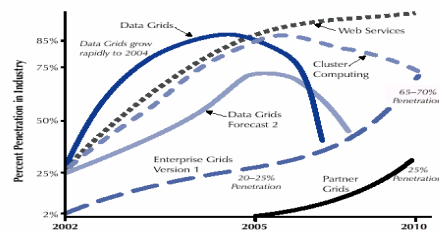
## Cost Savings from Grids

- The size of cost savings from grids will come in two waves:
  - First from the adoption of clusters
  - Then from the adoption of Enterprise Grids
- Firms using Clusters estimate that cost savings will be small at first, but will grow to 15% to 30% savings in IT Costs in 2005-2008.
- Firms planning to use Enterprise Grids estimate that they will experience a second wave of benefits. Savings will grow to 15% to 30% by 2007-2010.

Efficiency Gains from Grids, 2002–2010

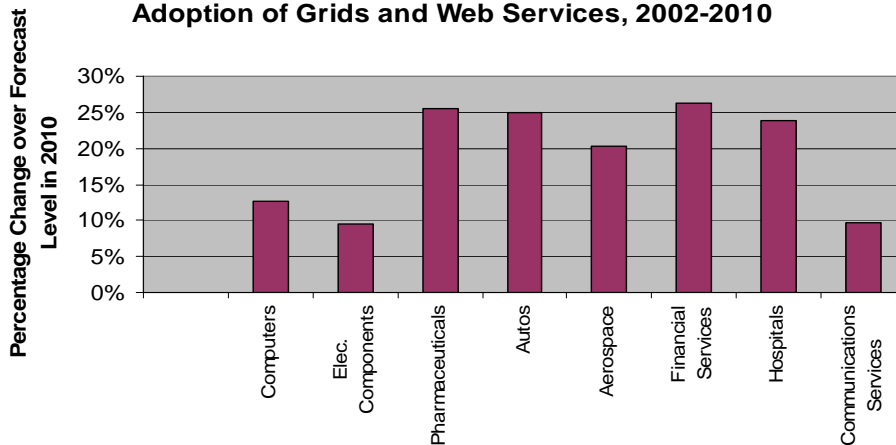


Grids and Web Services in the Pharmaceutical Industry, 2002–2010



Source: Robert Cohen, "Grid Computing: Projected Impact on North Carolina's Economy & Broadband Use through 2010," Rural Internet Access Authority, September 2003. <http://www.e-nc.org>

## Changes in U.S. Industry Productivity Due to the Adoption of Grids and Web Services, 2002-2010




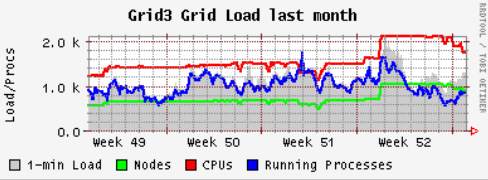
Because of significant cost savings, in addition to productivity gains and changes in supply chain relationships (especially in autos), many early adopter industries show dramatic gains in productivity over the levels expected in the 2010 economic forecast. This is one of the more striking findings of the research. There is a real possibility that once Web services & grids converge, firms will begin to transform their business processes & achieve even higher productivity benefits than the ones that we describe.

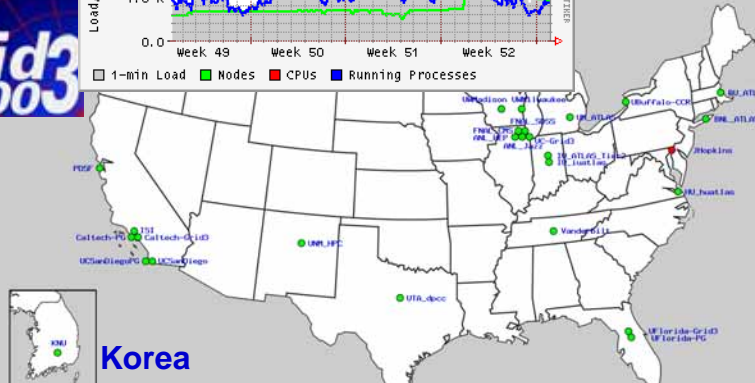
Source: Robert Cohen


## Grid2003: An Operational Grid

- > 28 sites (2100-2800 CPUs) & growing
- > 400-1300 concurrent jobs
- > 7 substantial applications + CS experiments
- > Running since October 2003










**Korea**  
South Korea

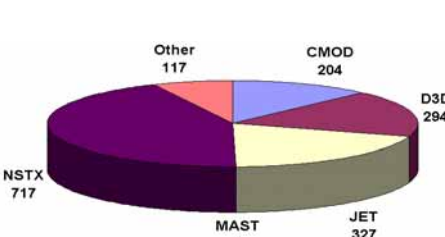
<http://www.ivdgl.org/grid2003>

Mon Nov 19 07:43:52 EST 2003
45




## Fusion Grid

Remote control room demo @ SC'03

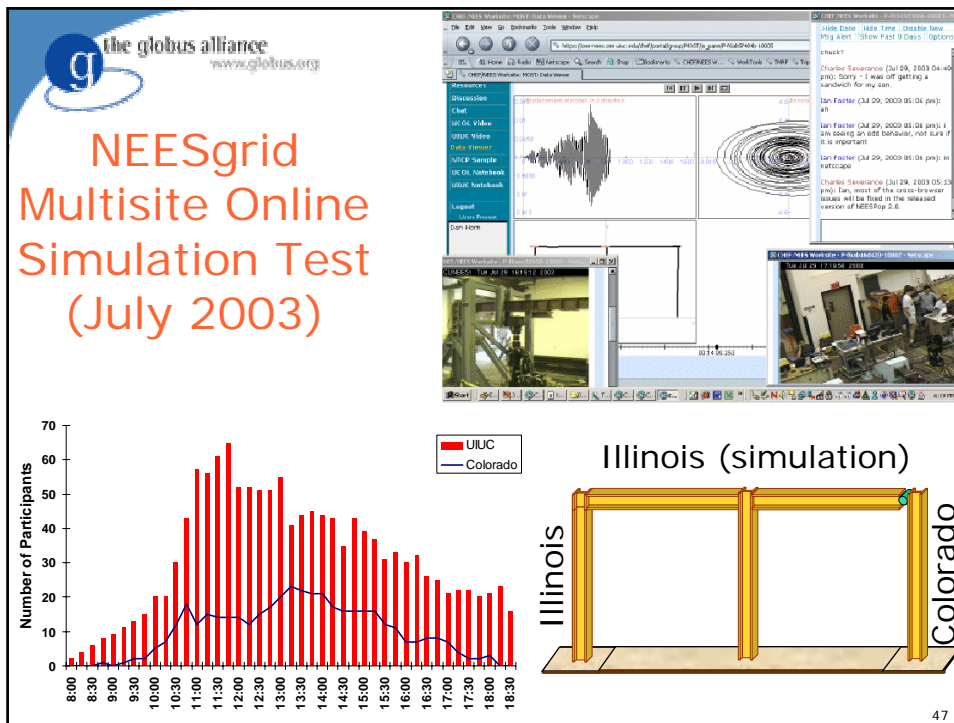


**TransP production service: 1662 runs in FY03**



Run ID	Code	File	Taskbook Name	Last Update	Comments
NR0006	TRANSF	test01	SET	Running	Run at 11:51:00 - success = 0.995.000
NR0007	TRANSF	test02	SET	Completed	Run at 12:00:00 - success = 0.995.000
NR0008	TRANSF	test03	SET	Completed	Run at 12:05:00 - success = 0.995.000
NR0009	TRANSF	test04	SET	Completed	Run at 12:10:00 - success = 0.995.000
NR0010	TRANSF	test05	SET	Completed	Run at 12:15:00 - success = 0.995.000

46



the globus alliance  
www.globus.org

## Overview

- What is Grid anyway?
- Grid meets Web services
- The power of open source
- Grid in practice
- **Summary**

48



## Summary

- Virtualization & management are key issues for Grid
  - Realization of service-oriented architecture
- Web services provide a powerful foundation on which to build an open Grid
  - Additional features required for Grid are being defined, e.g., WSRF, OGSA
- Open source implementations enable rapid dissemination & adoption



## For More Information

- Globus Alliance
  - [www.globus.org](http://www.globus.org)
- Global Grid Forum
  - [www.ggf.org](http://www.ggf.org)
- Background information
  - [www.mcs.anl.gov/~foster](http://www.mcs.anl.gov/~foster)

