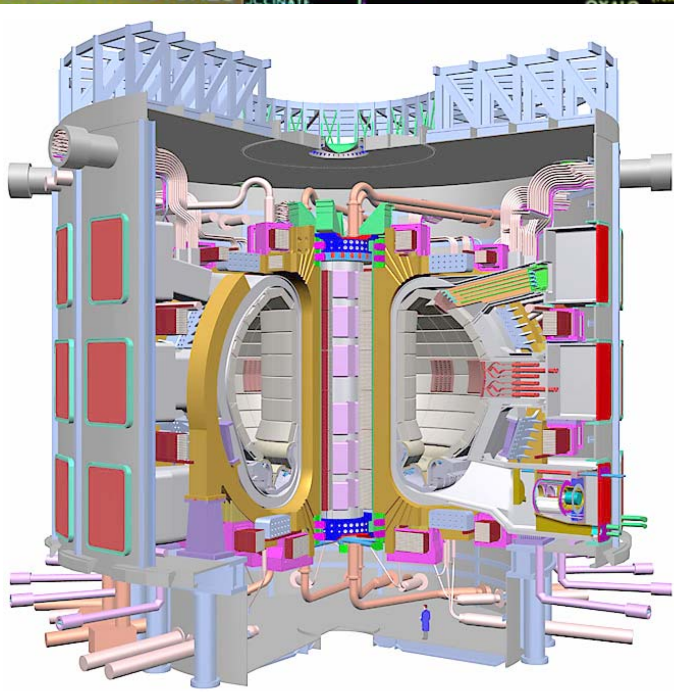
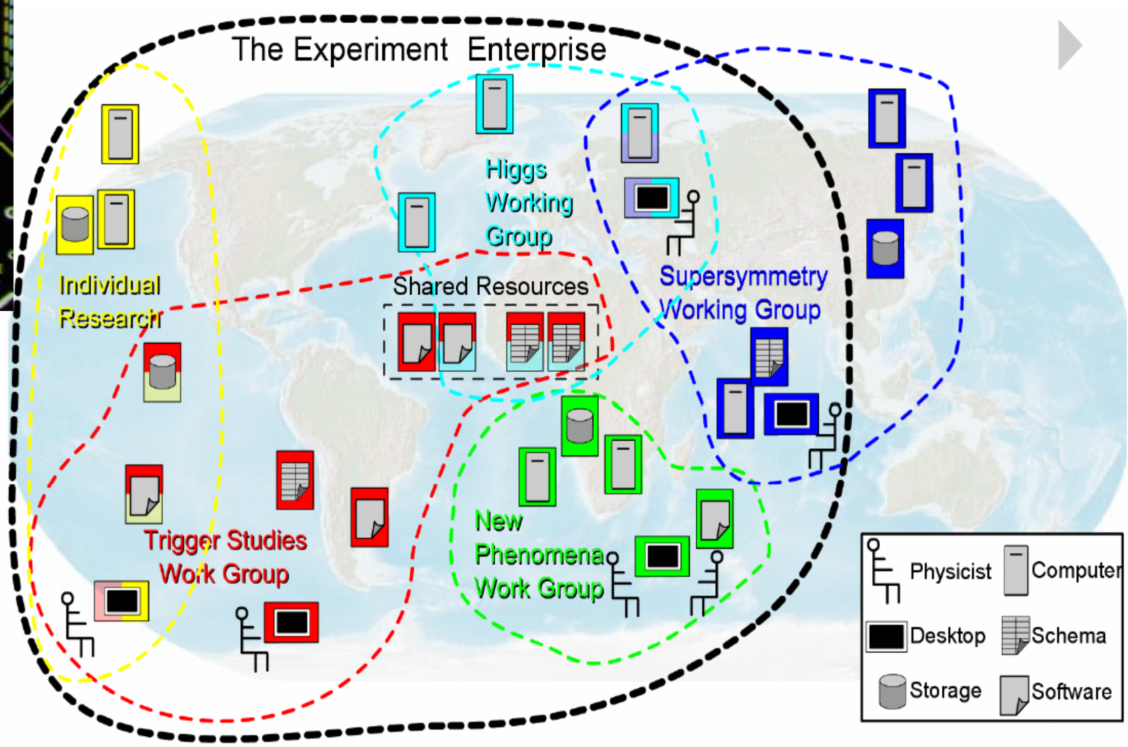
Introduction to Grids, Grid Middleware and Applications

Carl Kesselman

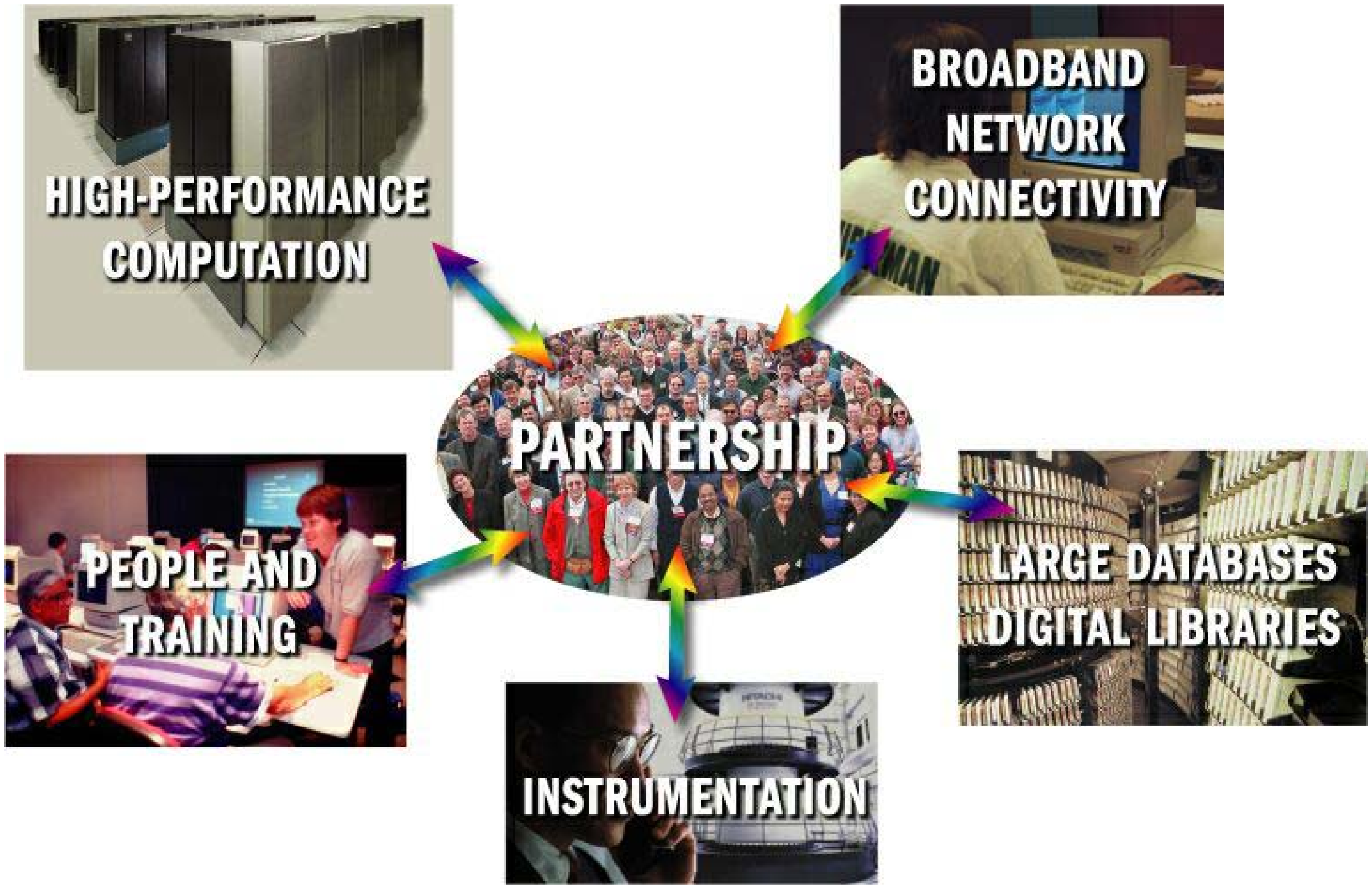
the globus alliance

Science Today is a Team Sport





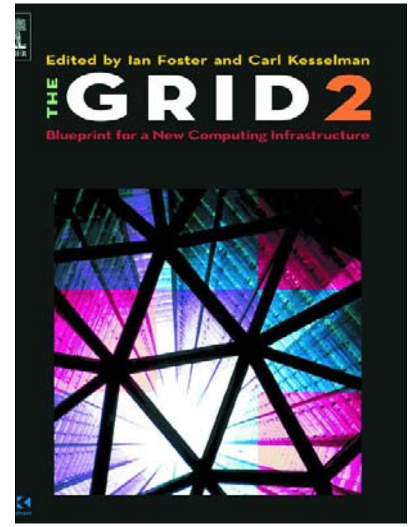
We Must be able to Assemble Required Expertise & Resources When Needed!

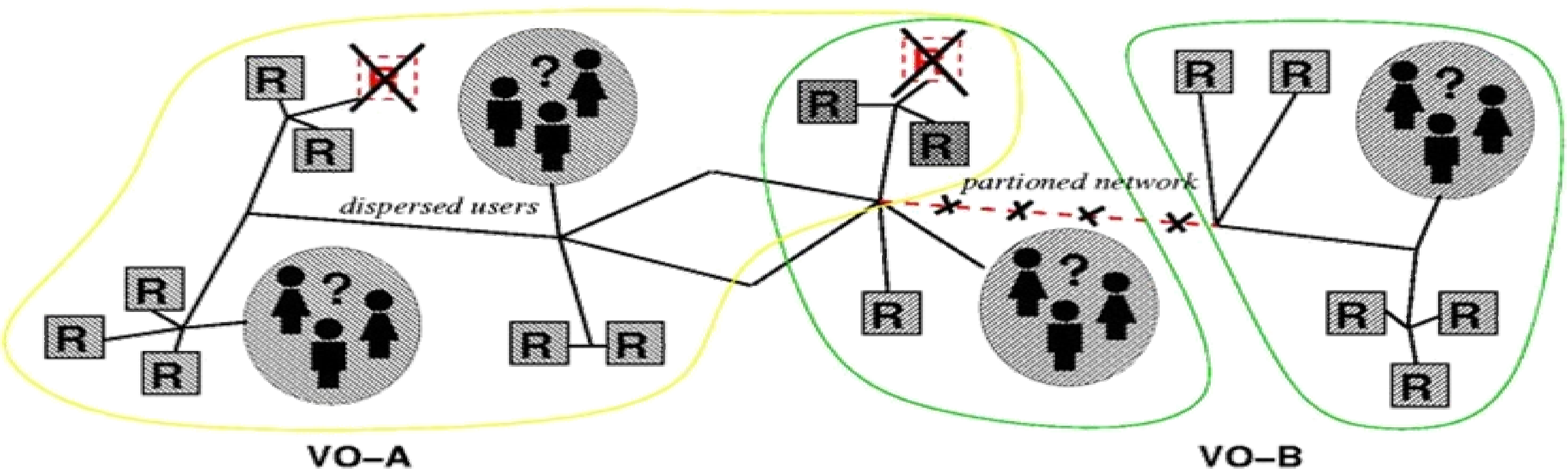


Transform resources into on-demand services accessible to any individual or team

A Unifying Concept: 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

What Problems is the Grid Intended to Address?

The Grid is a highly pragmatic field.

- It arose from *applied* computer science.
- It is focused on *enabling* new types of applications.
- Funding and investment in the Grid has been motivated by the promise of *new capabilities*—not in computer science, but in other fields and in other areas of work.

What Kinds of Applications?

- Computation intensive
  - Interactive simulation (climate modeling)
  - Very large-scale simulation and analysis (galaxy formation, gravity waves, battlefield simulation)
  - Engineering (parameter studies, linked component models)
- Data intensive
  - Experimental data analysis (high-energy physics)
  - Image and sensor analysis (astronomy, climate study, ecology)
- Distributed collaboration
  - Online instrumentation (microscopes, x-ray devices, etc.)
  - Remote visualization (climate studies, biology)
  - Engineering (large-scale structural testing, chemical engineering)
- In all cases, the problems were big enough that they required people in several organization to collaborate and share computing resources, data, instruments.

What Types of Problems?

- Your system administrators can't agree on a uniform authentication system, but you have to allow your users to authenticate once (using a single password) then use services on all systems, with per-user accounting.
- You need to be able to offload work during peak times to systems at other companies, but the volume of work they'll accept changes from day-to-day.

What Types of Problems?

- You and your colleagues have 6000 datasets from the past 50 years of studies that you want to start sharing, but no one is willing to submit the data to a centrally-managed storage system or database.
- You need to run 24 experiments that each use six large-scale physical experimental facilities operating together in real time.

What Types of Problems?

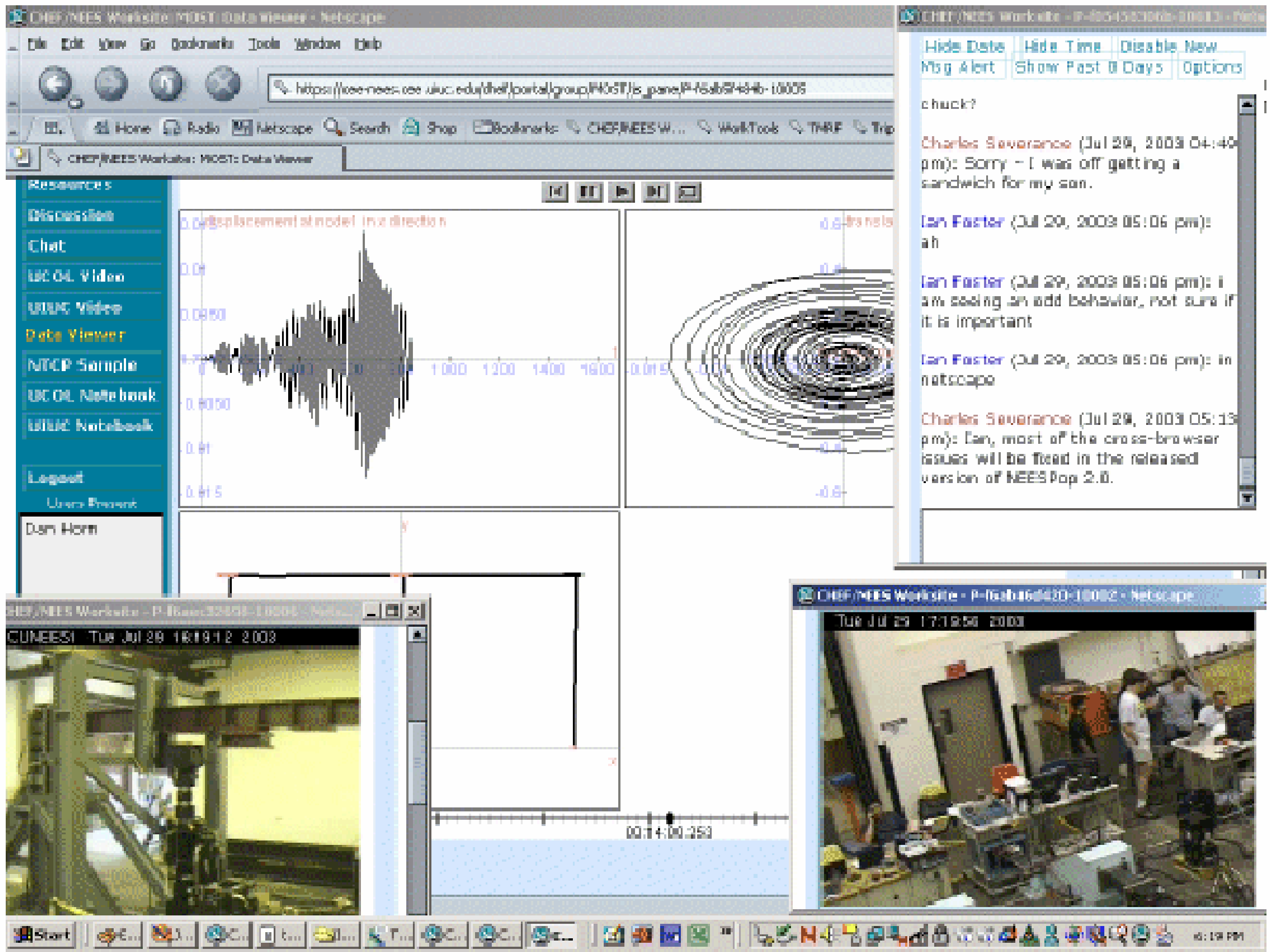
- Too hard to keep track of authentication data (ID/password) across institutions
- Too hard to monitor system and application status across institutions
- Too many ways to submit jobs
- Too many ways to store & access files and data
- Too many ways to keep track of data
- Too easy to leave “dangling” resources lying around (robustness)

Requirements "Themes"

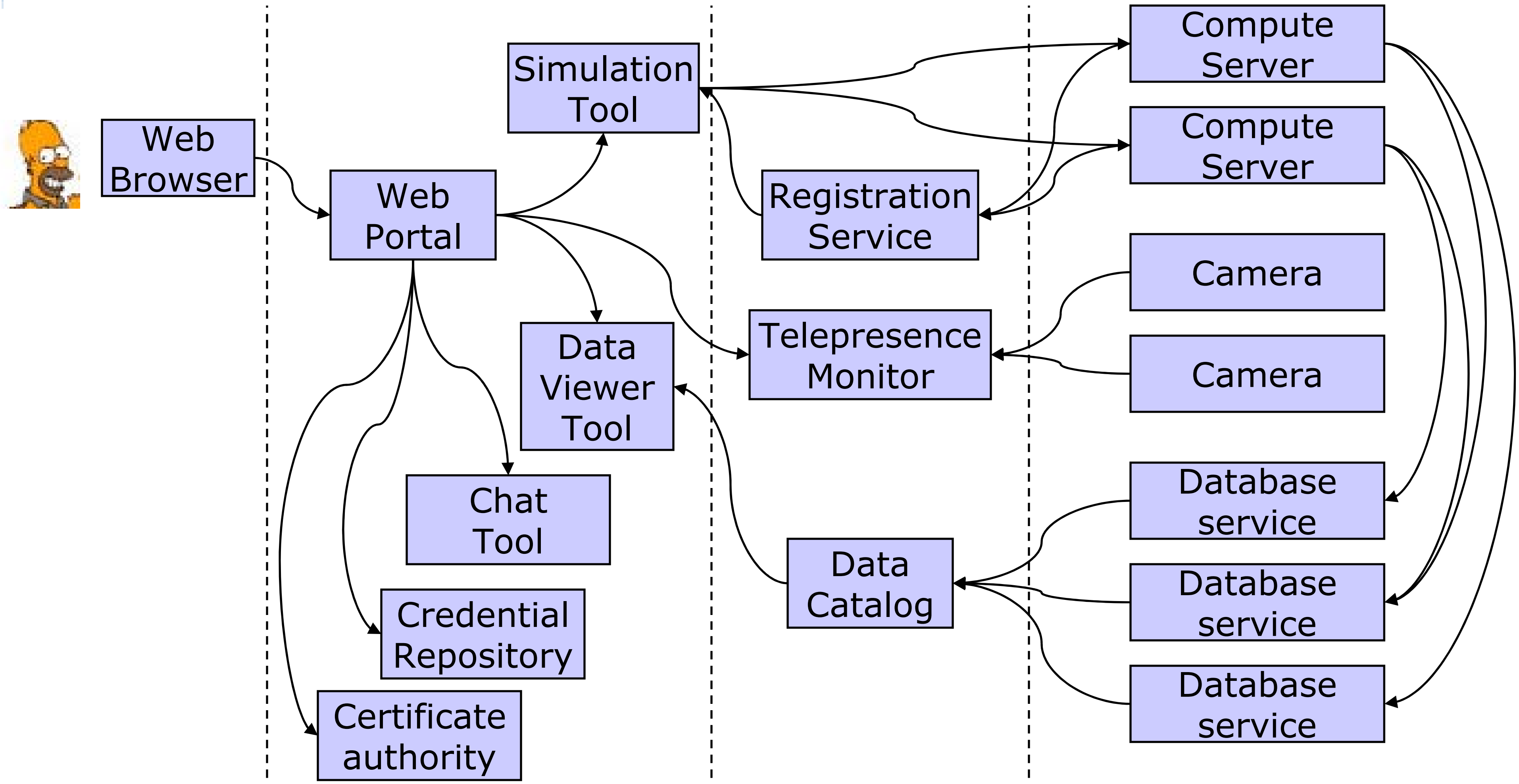
- Security
- Monitoring/Discovery
- Computing/Processing Power
- Moving and Managing Data
- Managing Systems
- System Packaging/Distribution

What End Users Need

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



How it *Really* Happens



Users work with client applications

Application services organize VOs & enable access to other services

Collective services aggregate &/or virtualize resources

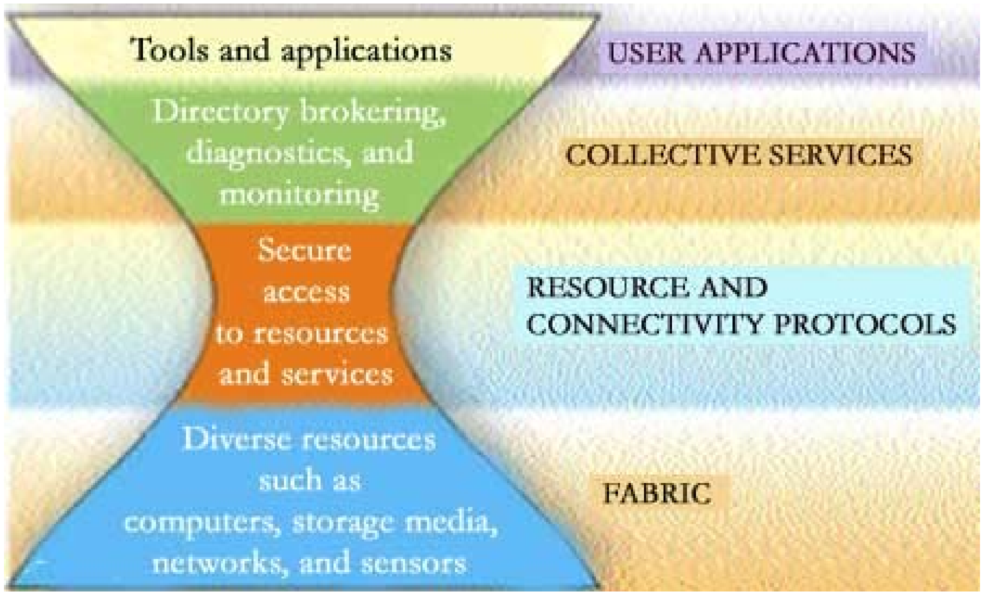
Resources implement standard access & management interfaces

How it *Really* Happens

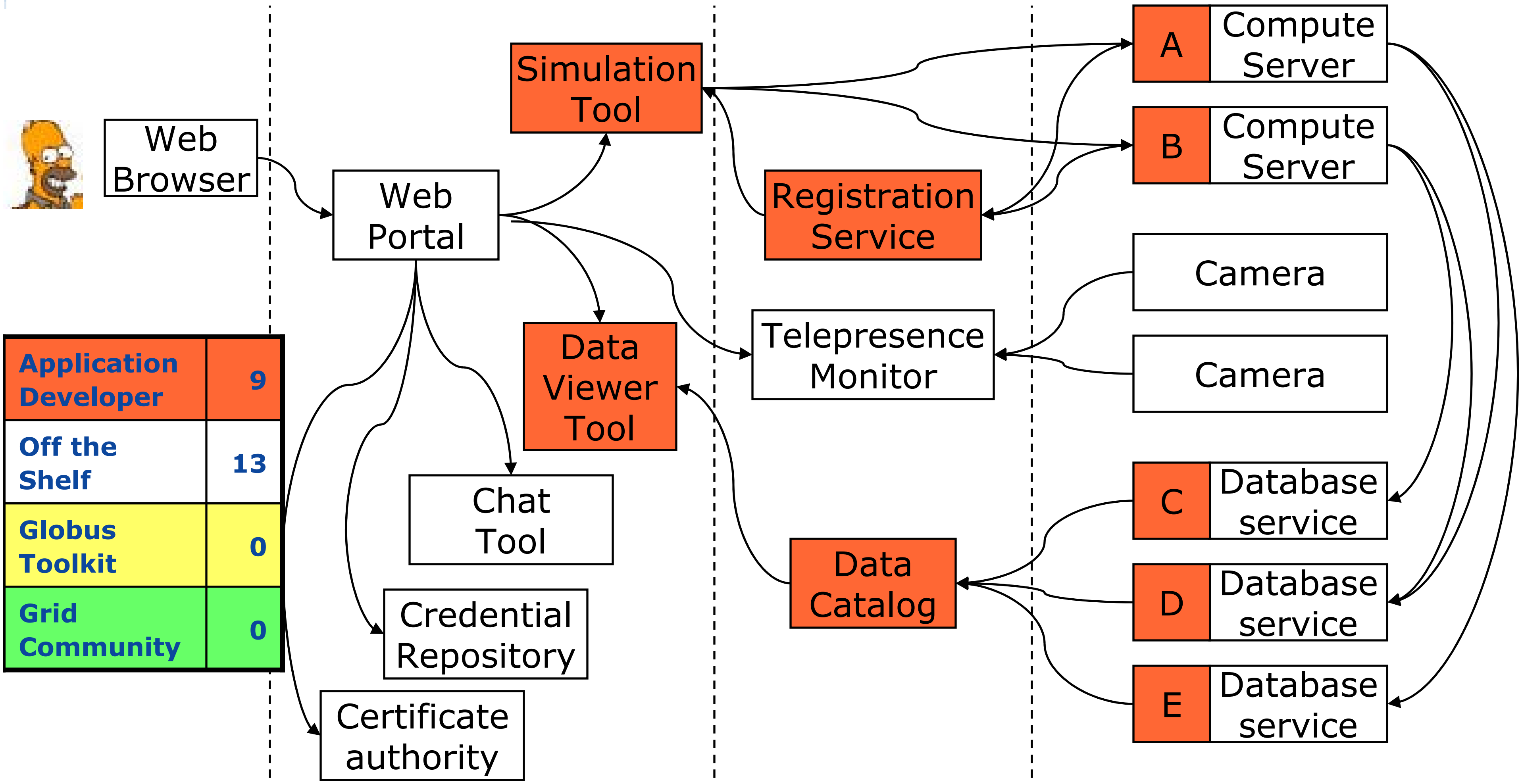
- Implementations are provided by a mix of
  - Application-specific code
  - "Off the shelf" tools and services
  - Common middleware tools and services
    - E.G. Globus Toolkit
  - Tools and services from the Grid community (compatible with GT)
- Glued together by…
  - Application development
  - System integration

Forget Homogeneity!

- Trying to force homogeneity on users is futile. Everyone has their own preferences, sometimes even *dogma*.
- The Internet provides the model…



How it *Really* Happens (without the Grid)



Application Developer	**9**
Off the Shelf	**13**
Globus Toolkit	**0**
Grid Community	**0**

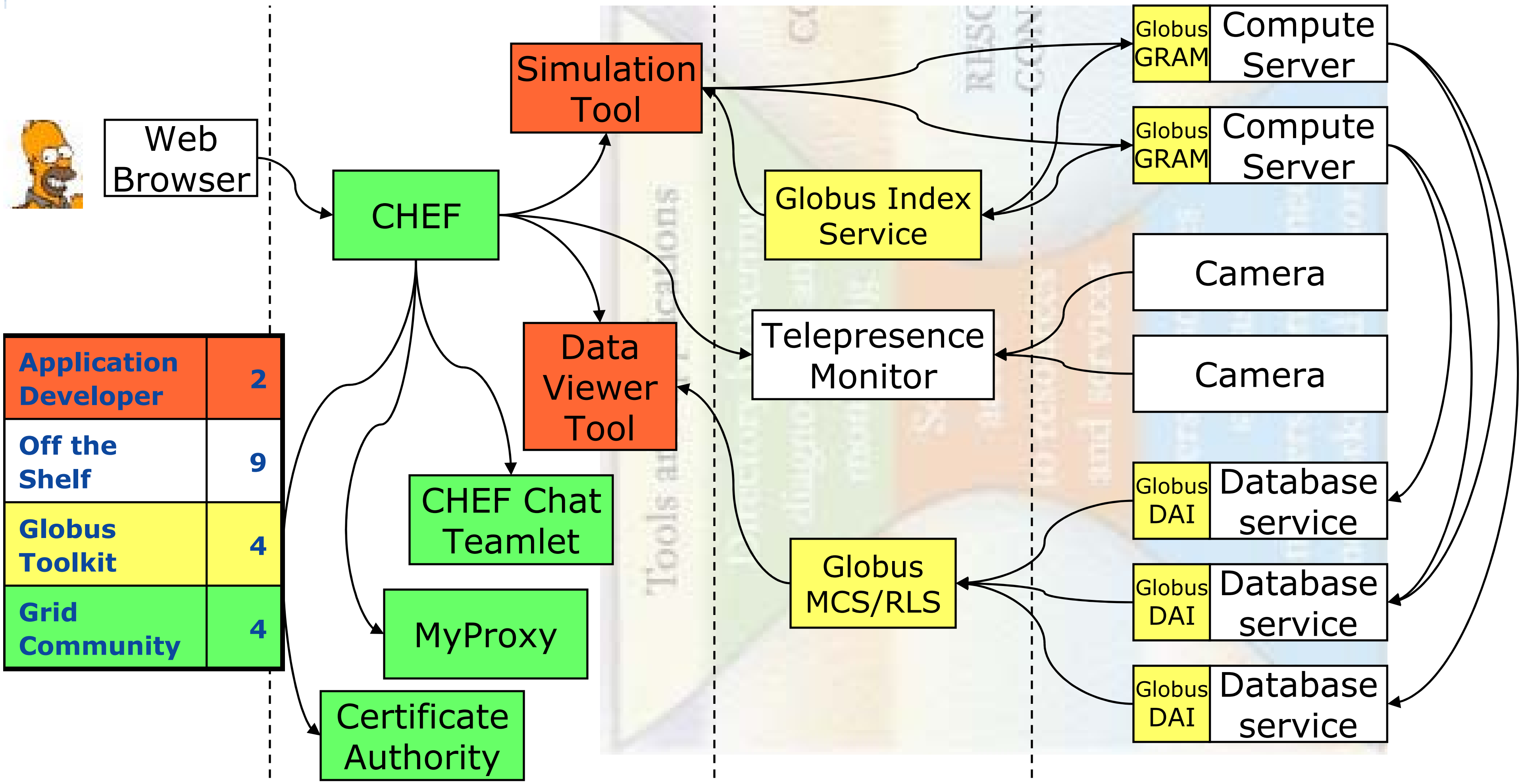
Users work with client applications

Application services organize VOs & enable access to other services

Collective services aggregate &/or virtualize resources

Resources implement standard access & management interfaces

How it *Really* Happens (with the Grid)



Application Developer	**2**
Off the Shelf	**9**
Globus Toolkit	**4**
Grid Community	**4**

Users work with client applications

Application services organize VOs & enable access to other services

Collective services aggregate &/or virtualize resources

Resources implement standard access & management interfaces

Why Standardize An Approach?

- Building large-scale systems by composition of many heterogeneous components demands that we extract and standardize common patterns
  - Approach to resource identification
  - Resource lifetime management interfaces
  - Resource inspection and monitoring interfaces
  - Base fault representation
  - Service and resource groups
  - Notification
  - And many more…
- Standardization encourages tooling & code re-use
  - Support to build services more quickly & reliably

Putting it All Together: 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, etc.
- …building on Web service standards.
  - extending those standards when needed

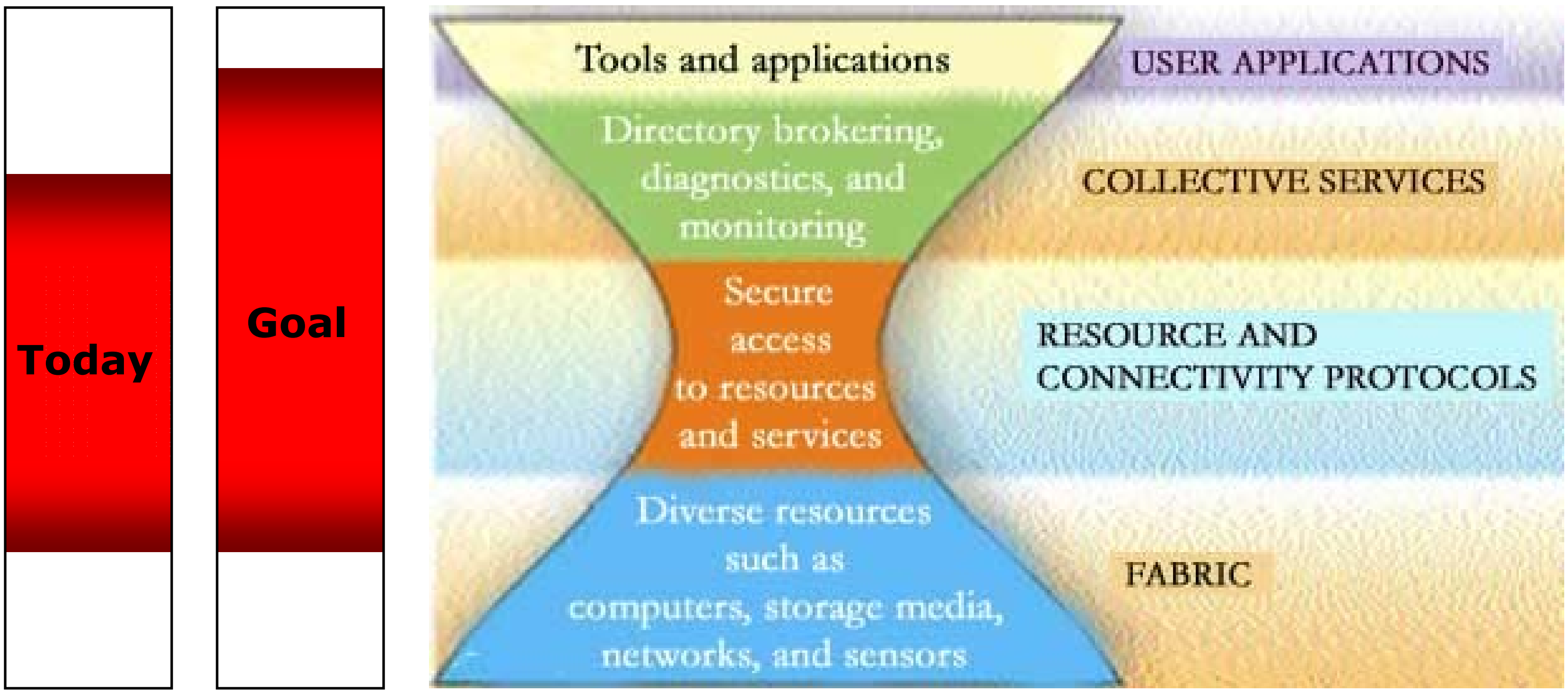
What Is the Globus Toolkit?

- The Globus Toolkit is a collection of solutions to problems that frequently come up when trying to build collaborative distributed applications.
- Heterogeneity
  - To date (v1.0 - v4.0), the Toolkit has focused on *simplifying heterogenity* for application developers.
  - We aspire to include more "vertical solutions" in future versions.
- Standards
  - Our goal has been to capitalize on and encourage use of existing standards (IETF, W3C, OASIS, GGF).
  - The Toolkit also includes reference implementations of new/proposed standards in these organizations.

“Standard Plumbing” for the Grid

- *Not* turnkey solutions, but *building blocks* and *tools* for application developers and system integrators.
  - Some components (e.g., file transfer) go farther than others (e.g., remote job submission) toward end-user relevance.
- Since these solutions exist and others are already using them (and they’re free), it’s easier to reuse than to reinvent.
  - And compatibility with other Grid systems comes for free!

How Far Does the Globus Toolkit Go?

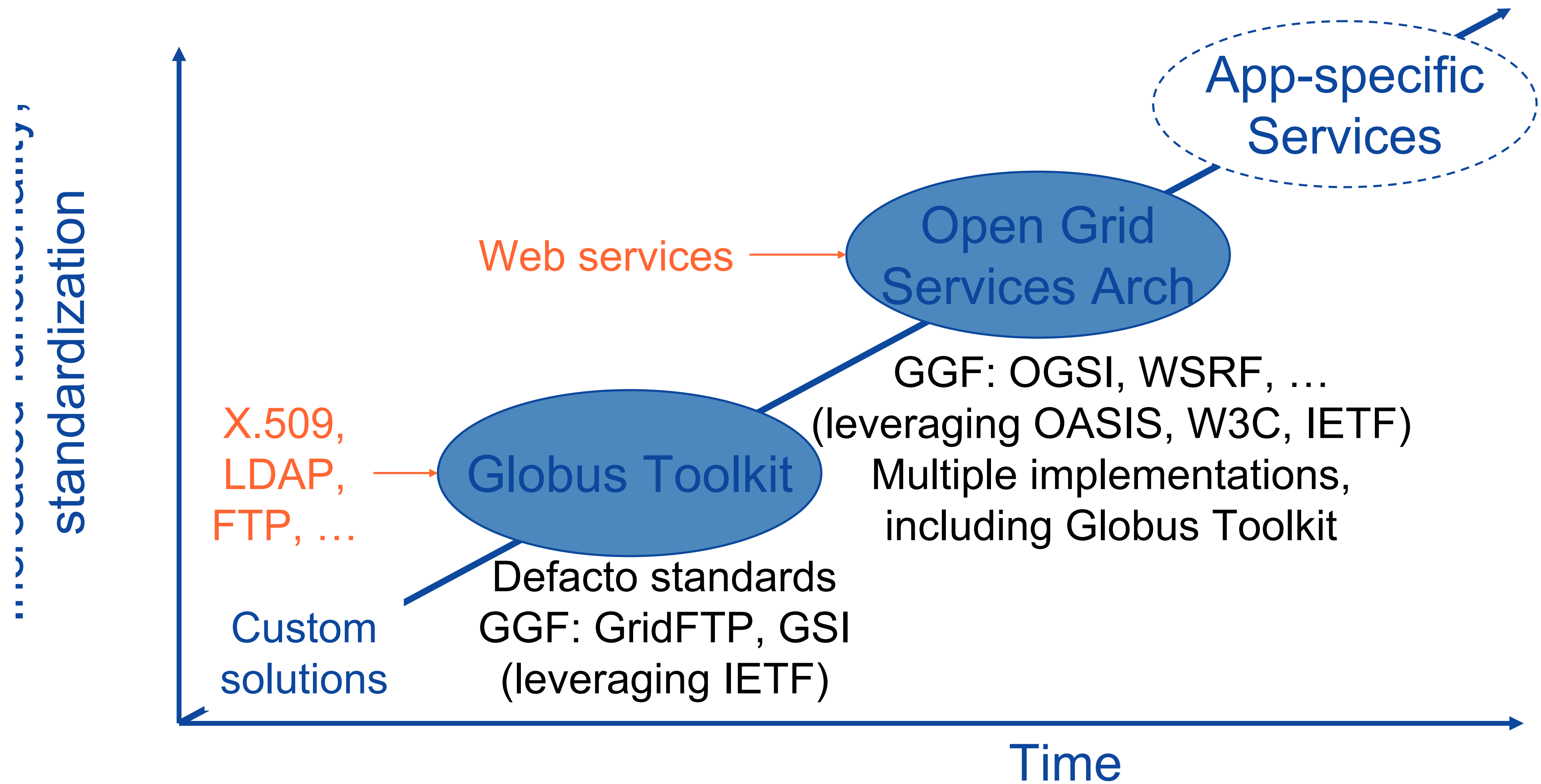


Leveraging Existing and Proposed Standards

- SSL/TLS v1 (from OpenSSL) (IETF)
- LDAP v3 (from OpenLDAP) (IETF)
- X.509 Proxy Certificates (IETF)
- GridFTP v1.0 (GGF)
- OGSI v1.0 (GGF)
- And others on the road to standardization:

  WSRF (GGF, OASIS), DAI, WS-Agreement, WSDL 2.0, WSDM, SAML, XACML

The "Grid Ecosystem"



What You Get in the Globus Toolkit

- OGSI(3.x)/WSRF(4.x) Core Implementation
  - Used to develop and run OGSA-compliant Grid Services (Java, C/C++)
- Basic Grid Services
  - Popular among current Grid users, common interfaces to the most typical services; includes both OGSA and non-OGSA implementations
- Developer APIs
  - C/C++ libraries and Java classes for building Grid-aware applications and tools
- Tools and Examples
  - Useful tools and examples based on the developer APIs

What Have You Got Now?

- A Grid development environment
  - Develop new OGSI-compliant Web Services
  - Develop applications using Grid APIs
- A set of basic Grid services
  - Job submission/management
  - File transfer (individual, queued)
  - Database access
  - Data management (replication, metadata)
  - Monitoring/Indexing system information
- Entry into Grid community software
  - Still more useful stuff!

How To Use the Globus Toolkit

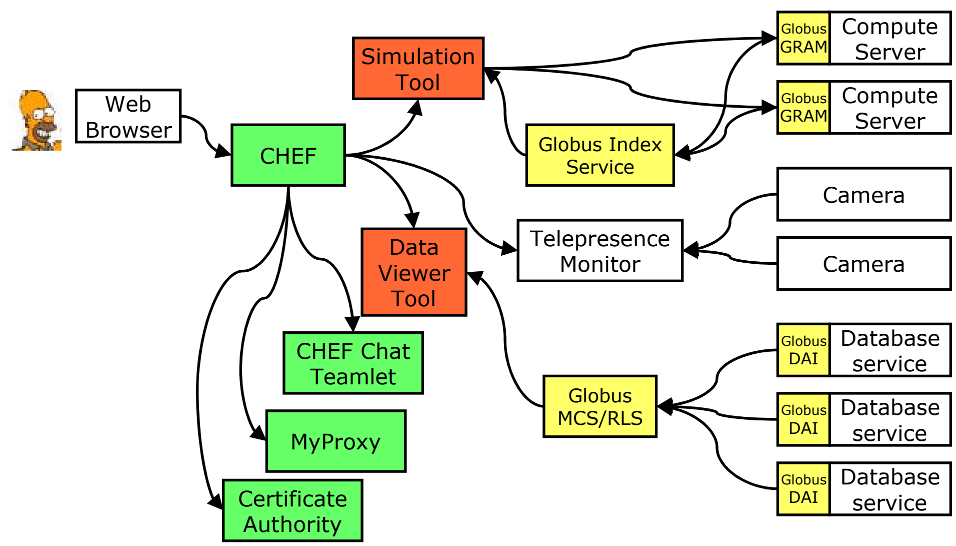
- By itself, the Toolkit has surprisingly limited *end user* value.
  - There's very little user interface material there.
  - You can't just give it to end users (scientists, engineers, marketing specialists) and tell them to do something useful!
- The Globus Toolkit is useful to *application developers* and *system integrators*.
  - You'll need to have a specific *application or system* in mind.
  - You'll need to have the right *expertise*.
  - You'll need to set up prerequisite *hardware/software*.
  - You'll need to have a *plan*.

Easy to Use – But Few Applications are "Easy"

- The uses that the Toolkit has been aimed at are *not* easy challenges!
- The Globus Toolkit makes them easier.
  - Providing solutions to the most common problems and promoting standard solutions
  - A well-designed implementation that allows many things to be built on it (lots of happy developers!)
  - 6+ years of providing support to Grid builders
  - Ever-improving documentation, installation, configuration, training

Architecture

- Once you have some decent requirements and some understanding of use cases…
  - Draw the system design.
  - Describe how the design will meet the needs of typical use cases.
  - Consider deployment and M&O requirements for the design.
  - Get feedback!
- You will start getting a sense of what components will be needed.



Select Components

- Within the system design, components will have functional requirements, too.
  - Capabilities ("features")
  - Interfaces (protocols, APIs, schema)
  - Performance/scalability metrics
- Ideally, much of it already exists.
  - Leverage what's already out there (Web, Grid, fabric technologies, off-the-shelf products, etc.).
  - Decompose into smaller bits if necessary.
  - If too much is unique to this application, you're probably doing something wrong.
  - If a candidate component is almost--but not quite--perfect, it can probably be extended (or used in conjunction with something else) to meet requirements.

Integration Plan

- Existing components must be integrated.
  - Identify “integration points”
  - Define interfaces
  - Develop “glue” if necessary
- New components must be developed.
  - Identify requirements (features+interfaces+performance)
  - Plan development

Application Development

- Phased "top-down" development
  - Focus on satisfying individual project goals or requirements in turn, or
  - Focus on widening deployment in turn.
  - Danger of "muddying" the architecture (inefficiencies creep in, especially regarding reusability).
- "Bottom-up" development
  - Focus first on components, then move to "system integration".
  - Danger of missing the "big picture" (missing unstated requirements).

Deployment

- Involve "real users" as early as possible.
  - You'll learn a lot and be able to "course correct."
  - You'll establish "happy users" to help in later stages.
- Pick early adopters carefully.
  - Aggressive users, technologically skilled, representative of the target user base.
  - Set expectations carefully.
  - Be wary of overinvestment.
- Deployment is a significant chunk of your effort.
  - Separate team?
  - Make sure it's linked to the development activity.

Computation-Intensive Science: Grid2003

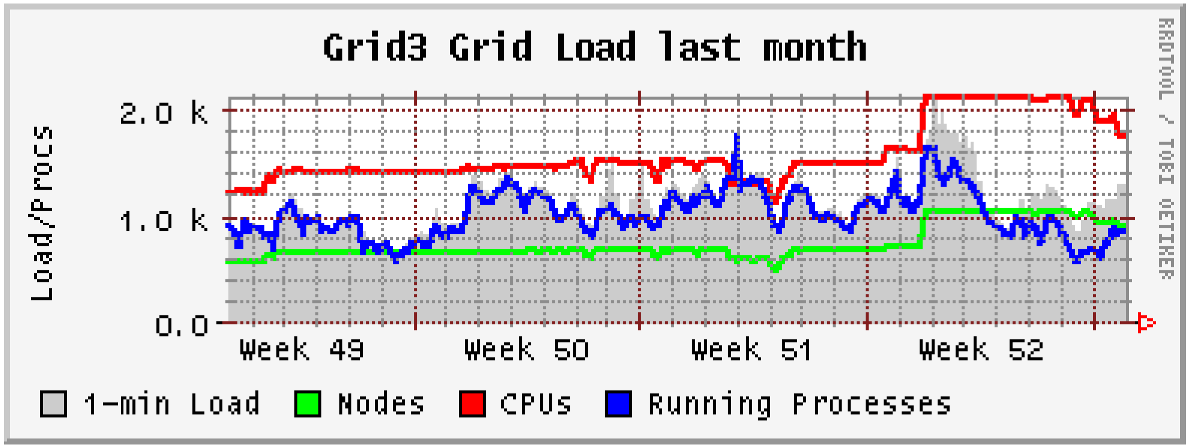
- GriPhyN - Grid Physics Network (NSF)
- iVDGL - International Virtual Data Grid Laboratory (NSF)
- LCG - LHC Computing Grid (EU)
- PPDG - Particle Physics Data Grid (DOE)

Grid2003 Project Goals

- Ramp up U.S. Grid capabilities in anticipation of LHC experiment needs in 2005.
  - Build, deploy, and operate a working Grid.
  - Include all U.S. LHC institutions.
  - Run real scientific applications on the Grid.
  - Provide state-of-the-art monitoring services.
  - Cover non-technical issues (e.g., SLAs) as well as technical ones.
- Unite the U.S. CS and Physics projects that are aimed at support for LHC.
  - Common infrastructure
  - Joint (collaborative) work

Grid2003 Requirements

- General Infrastructure
- Support Multiple Virtual Organizations
- Production Infrastructure
- Standard Grid Services
- Interoperability with European LHC Sites
- Easily Deployable
- Meaningful Performance Measurements



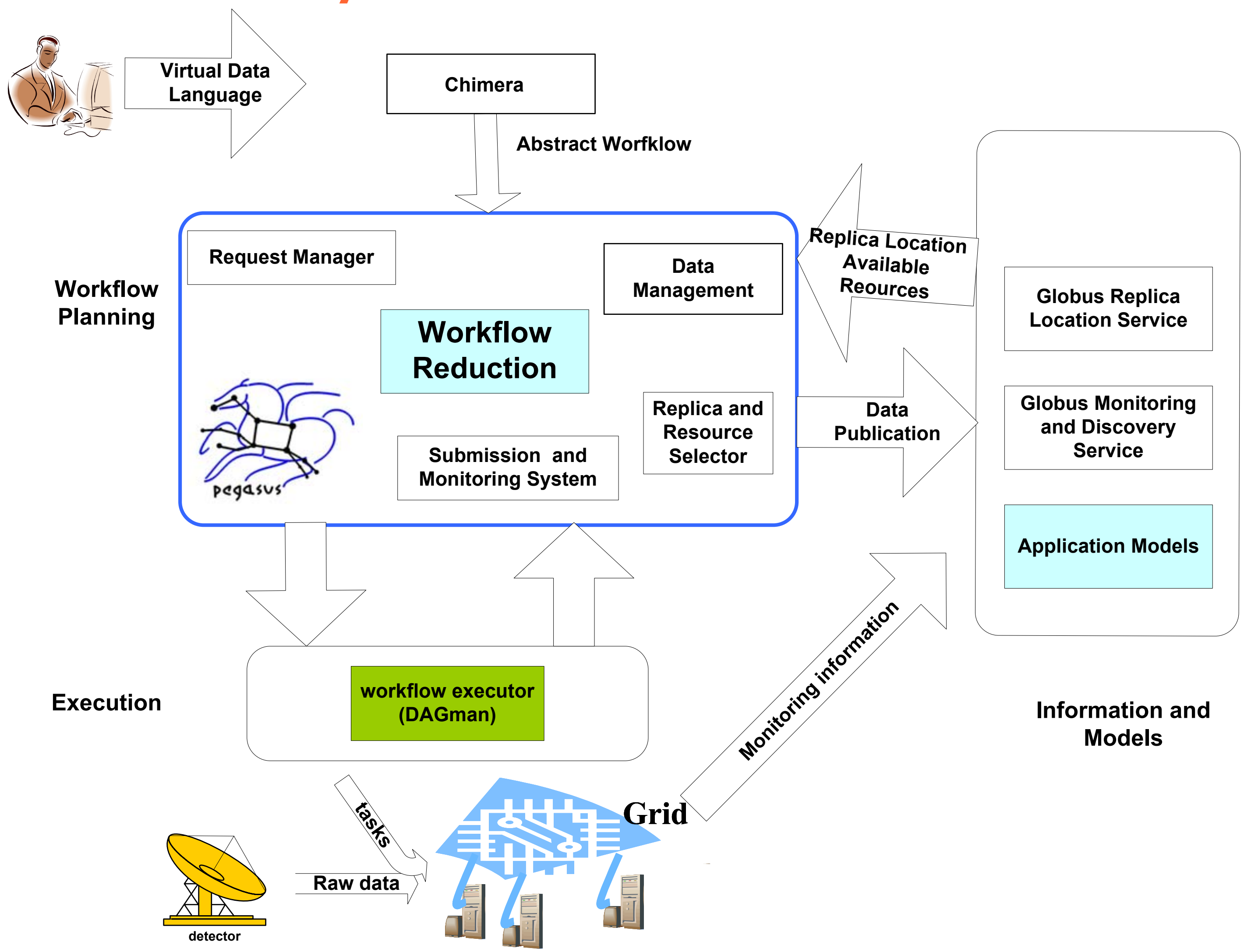
Grid2003 Components

- GT GRAM
- GT MDS
- GT GridFTP
- GT RLS
- GT MCS
- Condor-G
- DAGman
- Chimera & Pegasus
- GSI-OpenSSH
- MonALISA
- Ganglia
- VOMS
- PACMAN

Grid2003 Components

- **Computers & storage** at 28 sites (to date)
  - 2800+ CPUs
- **Uniform service environment** at each site
  - Globus Toolkit provides basic authentication, execution management, data movement
  - Pacman installation system enables installation of numerous other VDT and application services
- **Global & virtual organization services**
  - Certification & registration authorities, VO membership services, monitoring services
- **Client-side tools** for data access & analysis
  - Virtual data, execution planning, DAG management, execution management, monitoring
- **IGOC**: iVDGL Grid Operations Center

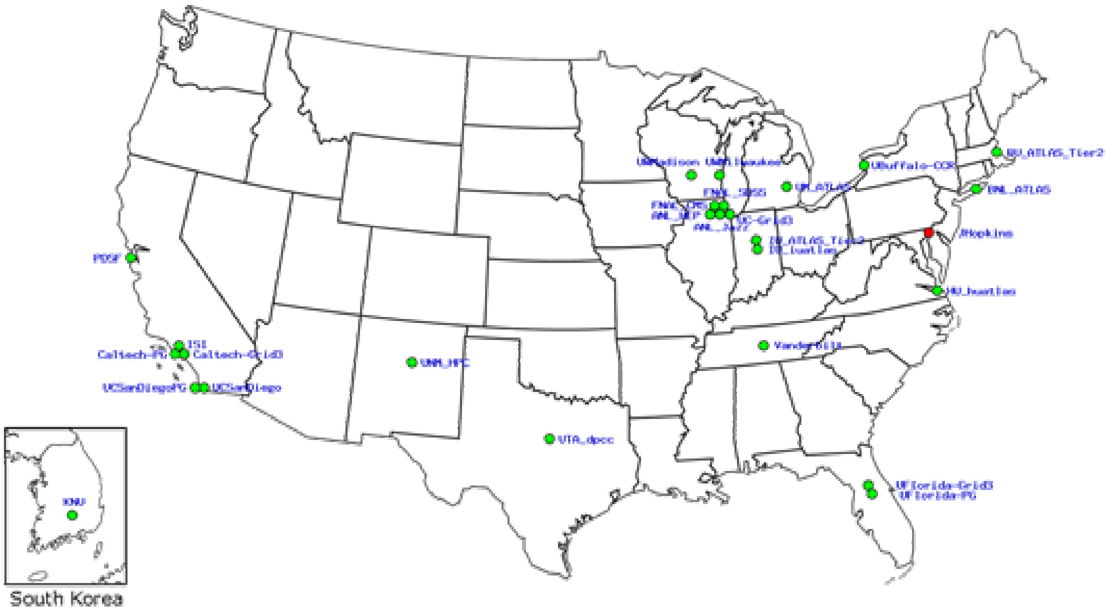
System Overview



Grid2003 Operation

- All software to be deployed is integrated in the Virtual Data Toolkit (VDT) distribution.
  - The VDT uses PACMAN to ease deployment and configuration.
  - Each participating institution deploys the VDT on their systems, which provides a standard set of software and configuration.
  - A core software team (GriPhyN, iVDGL) is responsible for VDT integration and development.
- A set of centralized services (e.g., directory services) is maintained Grid-wide.
- Applications are developed with VDT capabilities, architecture, and services directly in mind.

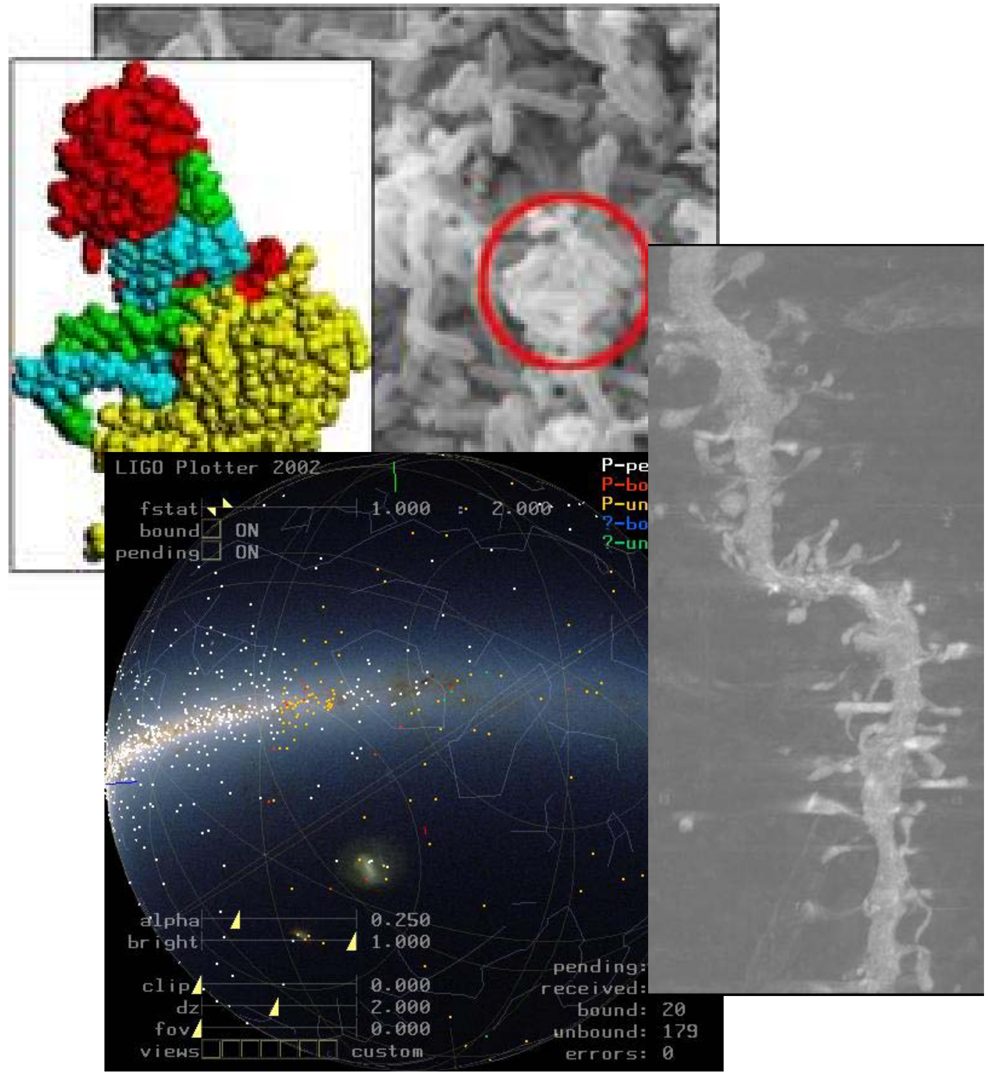
Grid2003 Deployment



- VDT installed at more than 25 U.S. LHC institutions, plus one Korean site.
- More than 2000 CPUs in total.
- More than 100 individuals authorized to use the Grid.
- Peak throughput of 500-900 jobs running concurrently, completion efficiency of 75%.

Grid2003 Applications

- 6 VOs, 11 Apps
- High-energy physics simulation and data analysis
- Cosmology based on analysis of astronomical survey data
- Molecular crystalography from analysis of X-ray diffraction data
- Genome analysis
- System “exercising” applications



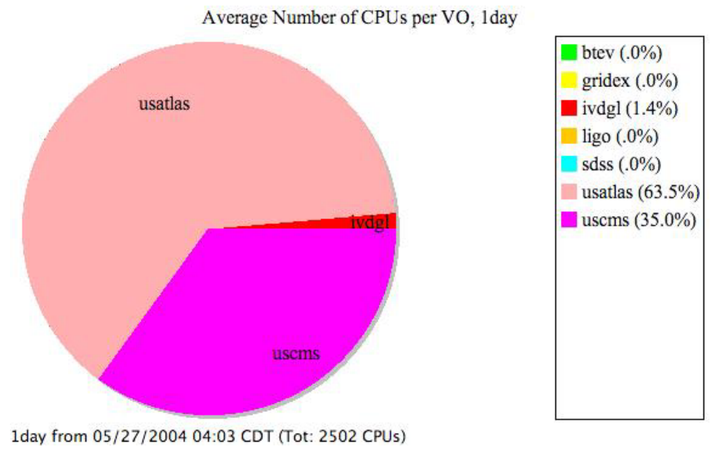
Grid2003 Applications To Date

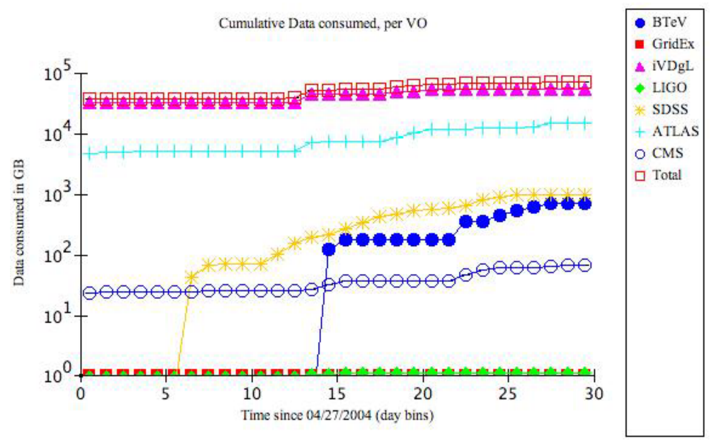
- CMS proton-proton collision simulation
- ATLAS proton-proton collision simulation
- LIGO gravitational wave search
- SDSS galaxy cluster detection
- ATLAS interactive analysis
- BTeV proton-antiproton collision simulation
- SnB biomolecular analysis
- GADU/Gnare genone analysis
- Various computer science experiments

www.ivdgl.org/grid2003/applications

Grid2003 Interesting Points

- Each virtual organization includes its own set of system resources (compute nodes, storage, etc.) and people. VO membership info is managed system-wide, but policies are enforced at each site.
- *Throughput* is a key metric for success, and monitoring tools are used to measure it and generate reports for each VO.



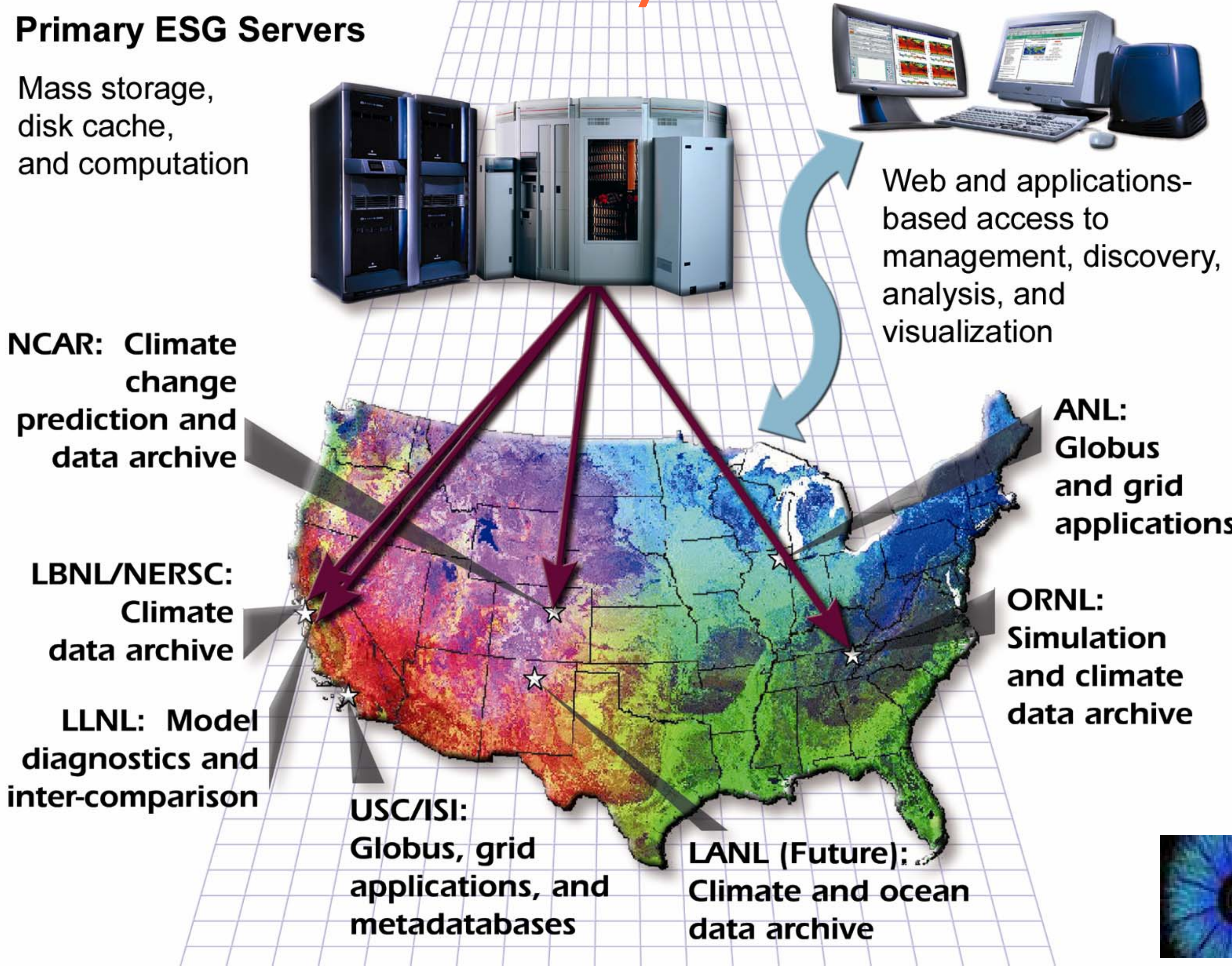


Grid2003 Metrics

Metric	Target	Achieved
Number of CPUs	400	2762 (28 sites)
Number of users	> 10	102 (16)
Number of applications	> 4	10 (+CS)
Number of sites running concurrent apps	> 10	17
Peak number of concurrent jobs	1000	1100
Data transfer per day	> 2-3 TB	4.4 TB max

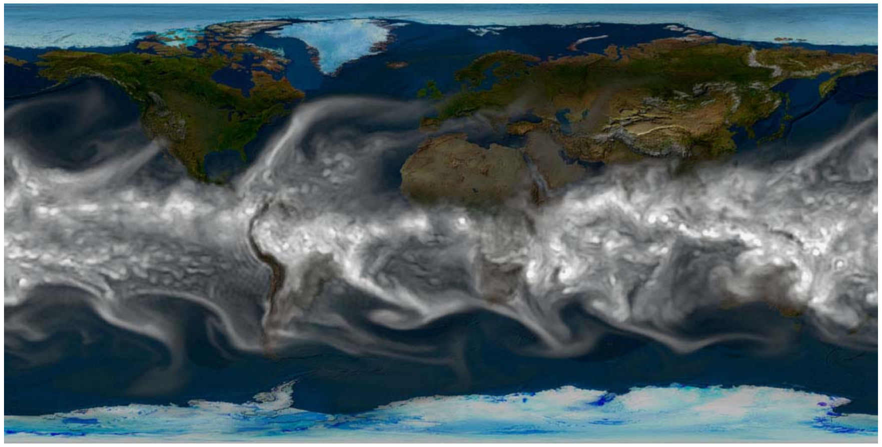
Data-Intensive Science: the Earth System Grid

ESG



Scientific Discovery through Advanced Computing

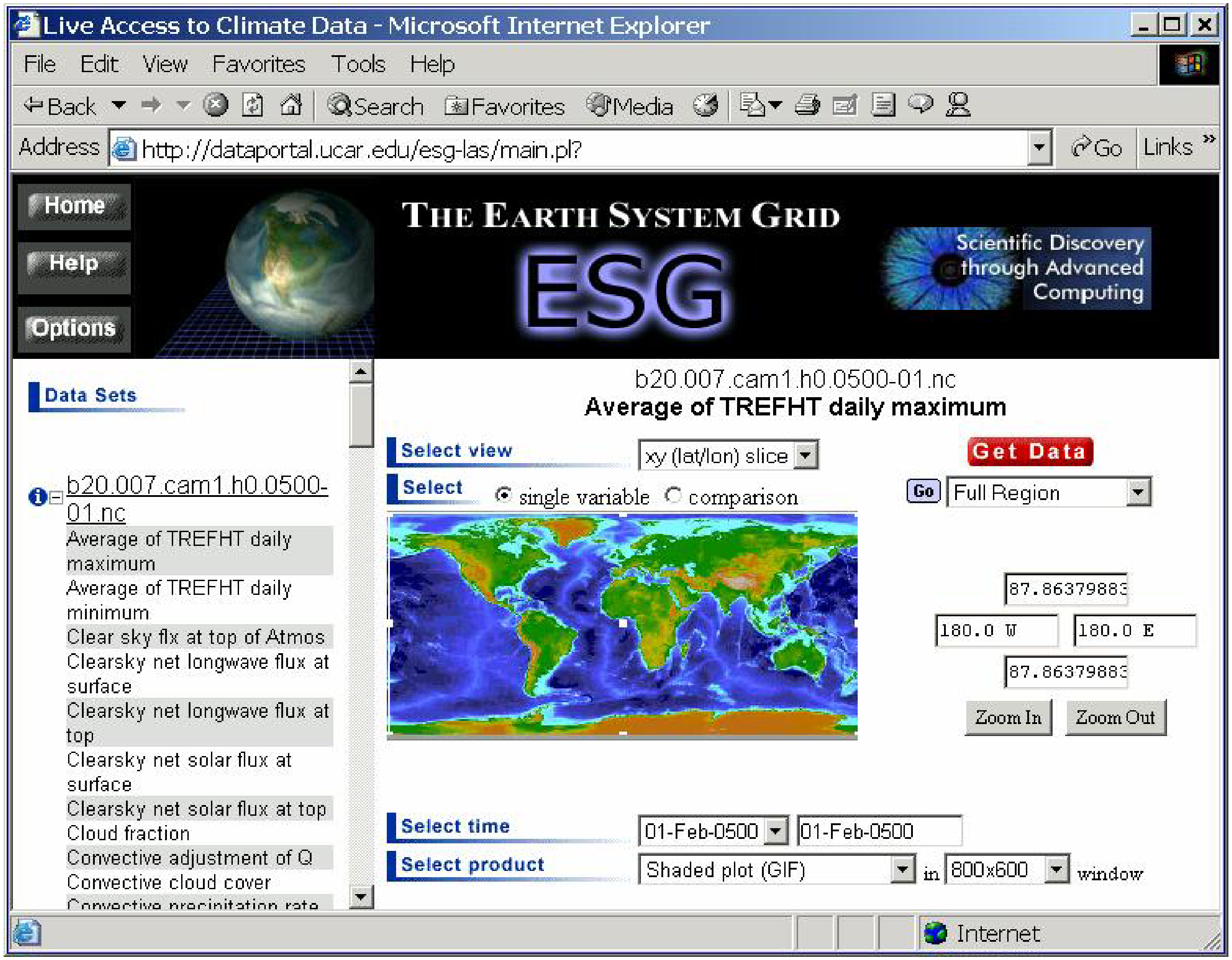
ESG Project Goals



- Improve productivity/capability for the simulation and data management team (data producers).
- Improve productivity/capability for the research community in analyzing and visualizing results (data consumers).
- Enable broad multidisciplinary communities to access simulation results (end users).
- The community needs an integrated “cyberinfrastructure” to enable smooth *workflow* for *knowledge development*: compute platforms, collaboration & collaboratories, data management, access, distribution, and analysis.

Earth System Grid

Goal: address technical obstacles to the sharing & analysis of high-volume data from advanced earth system models



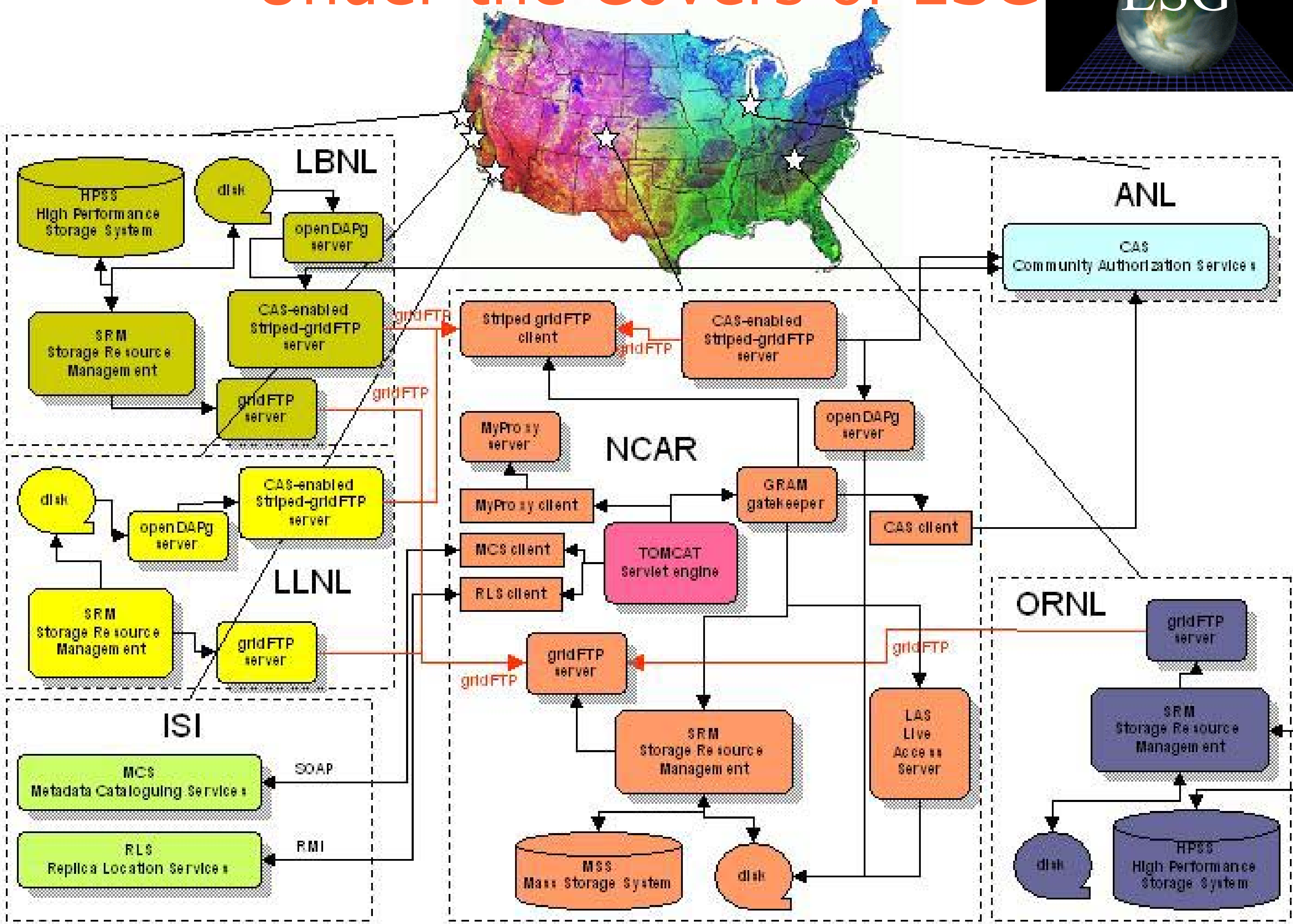
ESG Requirements

- Move data a minimal amount, keep it close to computational point of origin when possible.
- When we must move data, do it fast and with a minimum amount of human intervention.
- Keep track of what we have, particularly what's on deep storage.
- Make use of the facilities available at a number of sites. (Centralization is not an option.)
- Data must be easy to find and access using standard Web browsers.

Major ESG Components

- Grid Services
  - GRAM
  - GridFTP (+striped GridFTP server)
  - MDS (+WebSDV, +Trigger Service, +Archiver)
  - MyProxy
  - SimpleCA
  - RLS
  - MCS
- Other Services
  - OpenDAPg
  - HPSS
  - SRM
  - Apache, Tomcat
- ESG-specific services
  - Workflow Manager
  - Registration Service

Under the Covers of ESG



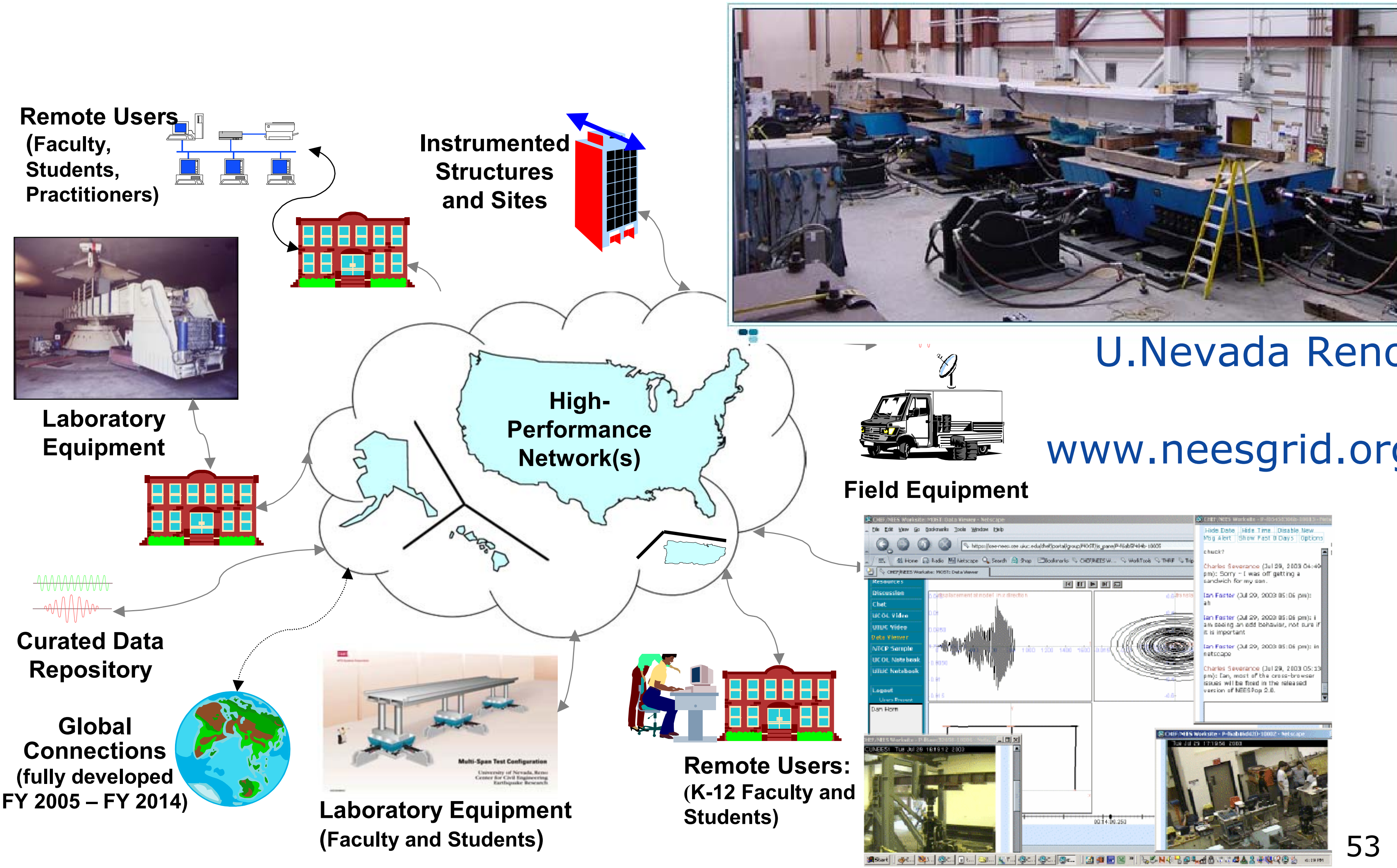
ESG Deployment

- Four data centers (LBNL, LLNL, NCAR, ORNL)
- User registration and authorization established
- Two major datasets are available, with associated metadata
- Work underway to add IPCC datasets as they are produced

ESG Interesting Points

- A lot of effort has been needed to build acceptable metadata models.
- Ease of use (simple interfaces, like registration service) is critical!
  - Users shouldn’t have to see anything other than web interface and the data they ask for.
  - Don’t bother giving certificates to users as long as they’re using the portal for everything.
- Specific goals (e.g., providing access to specific datasets) will dramatically focus work.

Collaborative Engineering: NEESgrid



U.Nevada Reno

www.neesgrid.org

NEESgrid System Integrators

- National Center for Supercomputing Applications (NCSA)
- Argonne National Laboratory
- USC-Information Sciences Institute
- University of Michigan
- Stanford University
- UC-Berkeley
- Pacific Northwest National Laboratory

NSF's Goals for NEESgrid

- Encourage collaboration among earthquake engineering researchers and practitioners.
  - Provide remote access to large-scale NSF earthquake engineering facilities.
  - Provide distributed collaboration tools.
  - Provide easy-to-use simulation capabilities.
  - Allow integration of physical and simulation capabilities.
  - Provide a community data repository for sharing data generated by use of the system.
- Create a *cyberinfrastructure* for earthquake engineering.
  - Define and implement Grid-based integration points for system components.

NEESgrid Core Capabilities

- Tele-control and tele-observation of experiments
- Data cataloging and sharing
- Remote collaboration and visualization tools and services
- Simulation execution and integration

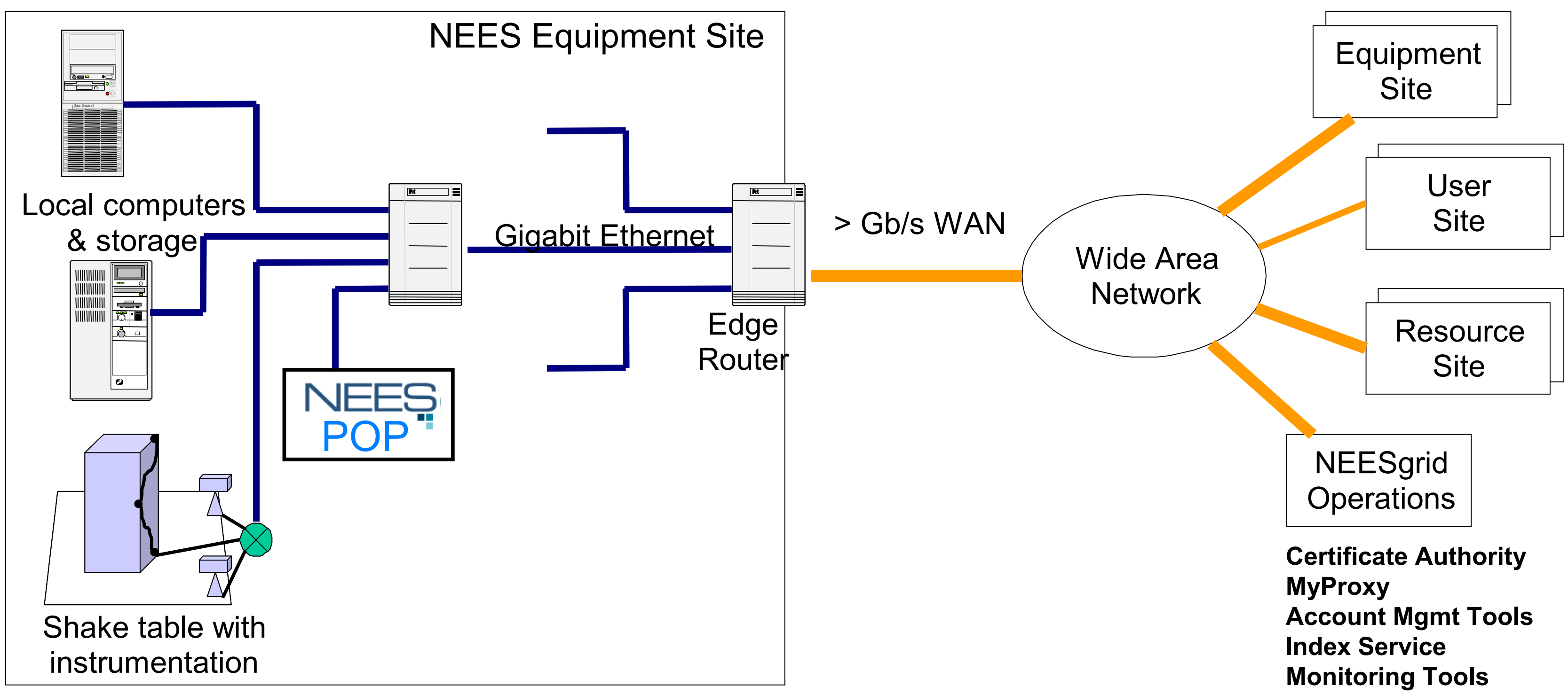
NEESgrid Requirements

- Single sign-on with Grid credentials
- Web interfaces for end users
  - Collaboration services (chat, video, documents, calendars, notebooks, etc.)
  - Telepresence services (video feeds)
  - Telecontrol (in limited instances)
  - Data viewing, data browsing and searching
  - Simulation capabilities
- Uniform interfaces for major system capabilities
  - Control
  - Data acquisition
  - Data streams
  - Data repository services

More NEESgrid Requirements

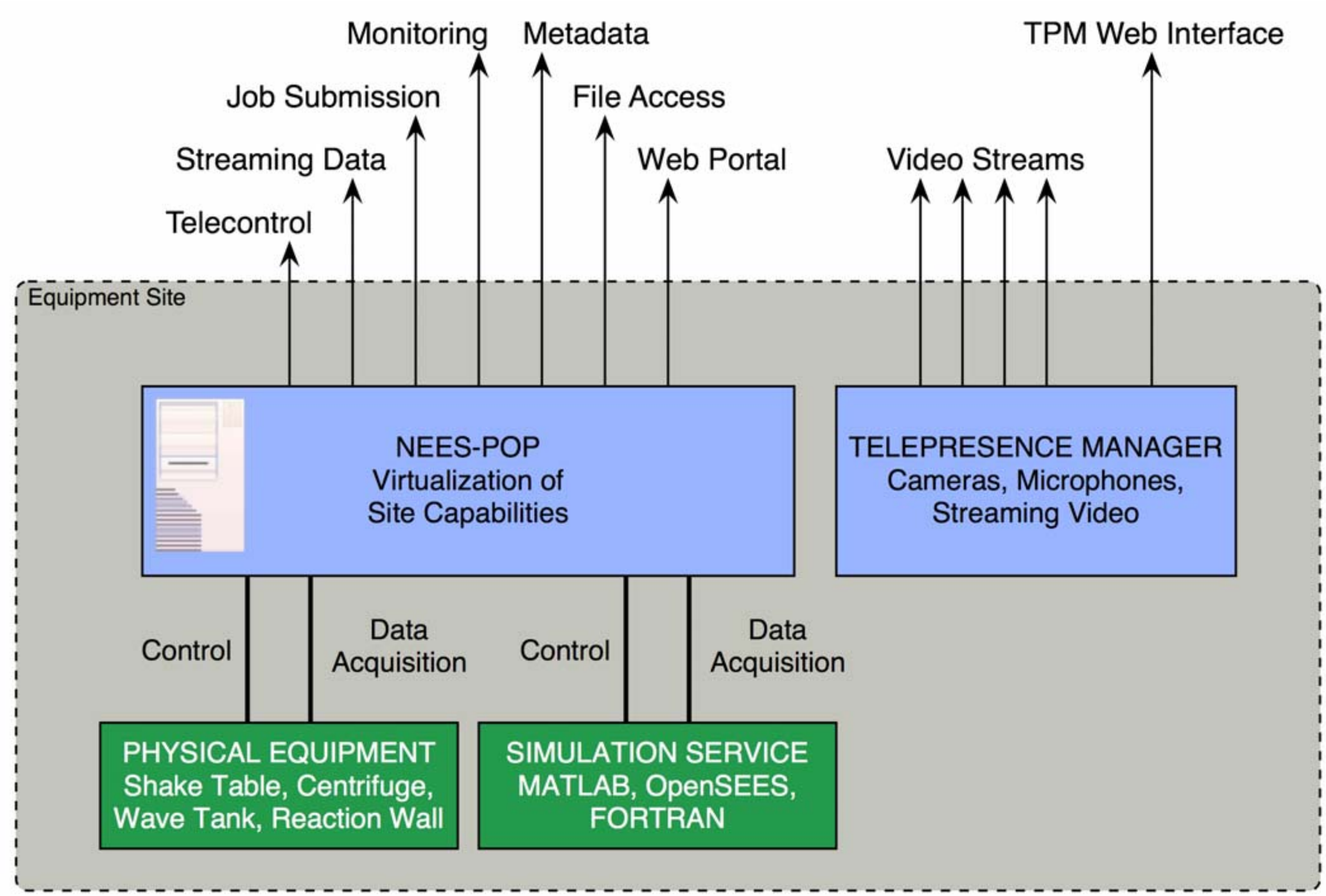
- System security
  - Protect facilities from misuse
  - Physical safety!
- Distributed collaboration during realtime experiments
- Automated (pre-programmed) control of distributed experiments (physical *and* simulation)
- Simplify effects of heterogeneity at facilities

NEESgrid High-level Structure



Certificate Authority
MyProxy
Account Mgmt Tools
Index Service
Monitoring Tools
NEESgrid Website
Bugzilla
Mailing Lists

Architecture of NEESgrid Equipment Site



Major NEESgrid Components

- OGSA Services
  - NTCP - Uniform Telecontrol Interface
  - NMDS - Metadata Repository Management
  - NFMS - File Repository Management
- Creare Data Turbine - Data & Video
- CHEF - Web Portal, Collaboration Tools
- NEESgrid Simulation Portal - Simulation Tools
- OpenSEES, FedeasLab - Simulation Frameworks
- Other Grid Services
  - MyProxy - Authentication
  - GridFTP - File Movement
  - GRAM - Job Submission/Management
  - MDS, Big Brother - System Monitoring
  - GSI-OpenSSH - Administrative Logins
  - GPT - Software Packaging

NEESgrid Deployment

- NEES-POPs installed at 16 facilities
- Experiment-based Deployment (EBD)
  - Sites propose experiments
  - SI and sites cooperatively run experiment using NEESgrid (deployment)
  - Tests architecture and components, identifying new requirements
- October 2004 transition to M&O team (SDSC)
- First round of research proposals also begin in October 2004
- Grand Opening in November 2004

NEESgrid Interesting Points

- Requirements are hard to define when a community is unused to collaboration.
  - Early deployment and genuine use is critical for focusing work.
  - Iterative design is useful in this situation.
- Considerable effort has been needed for data modeling (still unproven).
- Plug-in interfaces (“drivers”) are much more useful than originally imagined.
- Real users don’t want to deal with WSDL. They need user-level APIs.

the globus alliance
www.globus.org

Lessons Learned

- The Globus Toolkit has useful stuff in it.
- To do anything significant, a lot more is needed.
  - The Grid community (collectively) has many useful tools that can be reused!
  - System integration expertise is mandatory.
- OGSA and community standards (GGF, OASIS, W3C, IETF) are extremely important in getting all of this to work together.
- There’s much more to be done!

Continue Learning

- Visit the Globus Alliance website at: www.globus.org
- Read the book: *The Grid: Blueprint for a New Computing Infrastructure (2nd edition)*
- Talk to others who are using the Toolkit: discuss@globus.org (subscribe first)
- Participate in standards organizations: GGF, OASIS, W3C, IETF