Introduction to Grid Computing

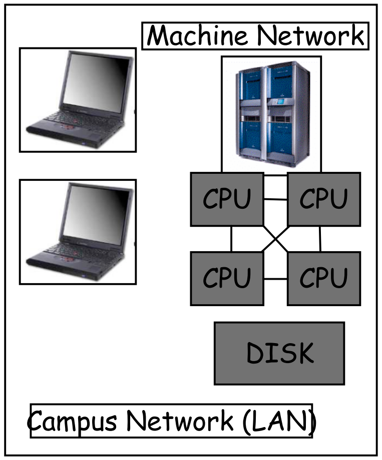
Open Science Grid

Ian Foster’s Grid Checklist (2002)

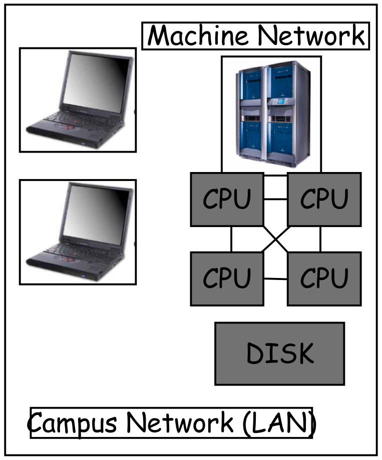
- A Grid is a system that:
  - Coordinates resources that are not subject to centralized control
  - Uses standard, open, general-purpose protocols and interfaces
  - Delivers non-trivial qualities of service

Components for Grid Computing

- Distributed People
  - Research communities who need to share data, or codes, or computers, or equipment to work on and understand common problems
  - Example: Astrophysics Network: relativists, astrophysicists, computer scientists, mathematicians, experimentalists, data analysts



- Distributed Resources
  - Computers: supercomputers, clusters, workstations
  - Storage devices, databases, networks
  - Experimental equipment: telescopes/interferometers



Components for Grid Computing

- Software infrastructure
  - Links all these together
  - *Low level:* security, information, communication, …
  - *Middleware:* data management, resource brokers, web portals, monitoring, workflow, …
- Examples
  - Globus
  - Condor

Virtual Organizations

- Groups of organizations that use the Grid to share resources for specific purposes
- Support a single community
- Deploy compatible technology and agree on working policies
  - Security policies - difficult
- Deploy different network accessible services:
  - Grid Information
  - Grid Resource Brokering
  - Grid Monitoring
  - Grid Accounting

Components for Grid Computing

- Applications
- Support for applications
  - Standard toolkits
  - SDKs/APIs
  - Libraries
  - Web portals
- Application code
  - Must be very portable
  - Must be machine independent, location independent
  - Lots of existing science code is not
- Development tools (debuggers, profilers, …)

Nature of Large Scale Distributed Applications

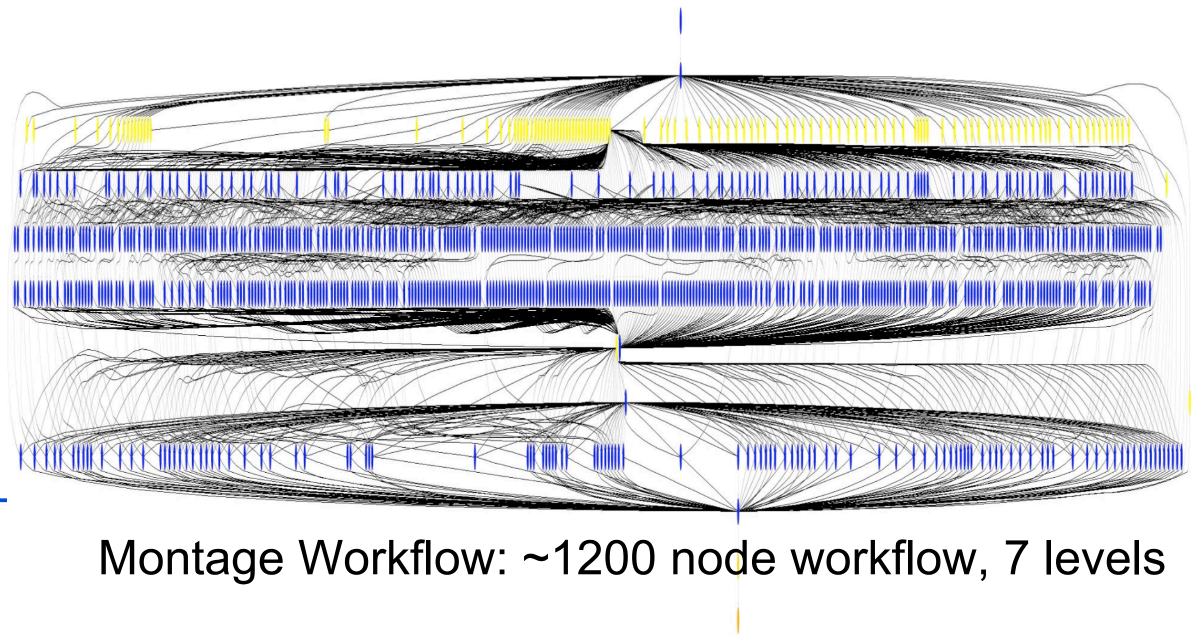
- Distributed data
  - Stored in different places. Different access policies. Expensive to move around.
- Distributed Resources
  - Resources are distributed across multiple organizations
  - Each resource looks different
- Distributed Ownership
  - Data and resources are owned by different organizations

Examples of Distributed Applications

- High Energy Physics applications
  - Monte Carlo simulations
  - CMS experiment
- Finding interesting astronomical patterns
  - Sloan Digital Sky Survey
- Coastal ocean monitoring and predicting
  - SURA Coastal Ocean Observing and Prediction (SCOOP)
- Prime number generator
  - Cracking cryptography
- Divide the application and run it on a distributed and decentralized environment

One typical application

- Many HEP and Astronomy experiments consist of:
  - Large datasets as inputs (find datasets)
  - "Transformations" which work on the input datasets (process)
  - The output datasets (store and publish)
- The emphasis is on the sharing of the large datasets
- Transformations are usually long and can be parallelized



Montage Workflow: ~1200 node workflow, 7 levels

Grid Application Types

- Minimal communication (embarrassingly parallel)
- Staged/linked/workflow
- Access to certain resources
- Fast throughput
- Large scale
- Adaptive
- Real-time on demand
- Speculative

Common Infrastructure

- Most common core Grid infrastructure deployed today is called the Globus Toolkit.
- Many higher level services are being researched and built using Globus
- www.globus.org
- Originally from Argonne National Lab and University of Southern California, in US.

The Open Grid Forum

- Standards and best practices
- Promoting Grid technologies and applications
- Modelled around bodies such as IETF (internet engineering task force)
- Working groups and research groups in many different areas
- Meet 3 times a year
- www.ogf.org

Definitions

Application Programming Interface (API) defines the interface.

- Refers to definition, not implementation
  - For example, there are many implementations of MPI
- Specification 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 (security), MPI (message passing)

A Software Development Kit (SDK) is a particular instantiation of an API

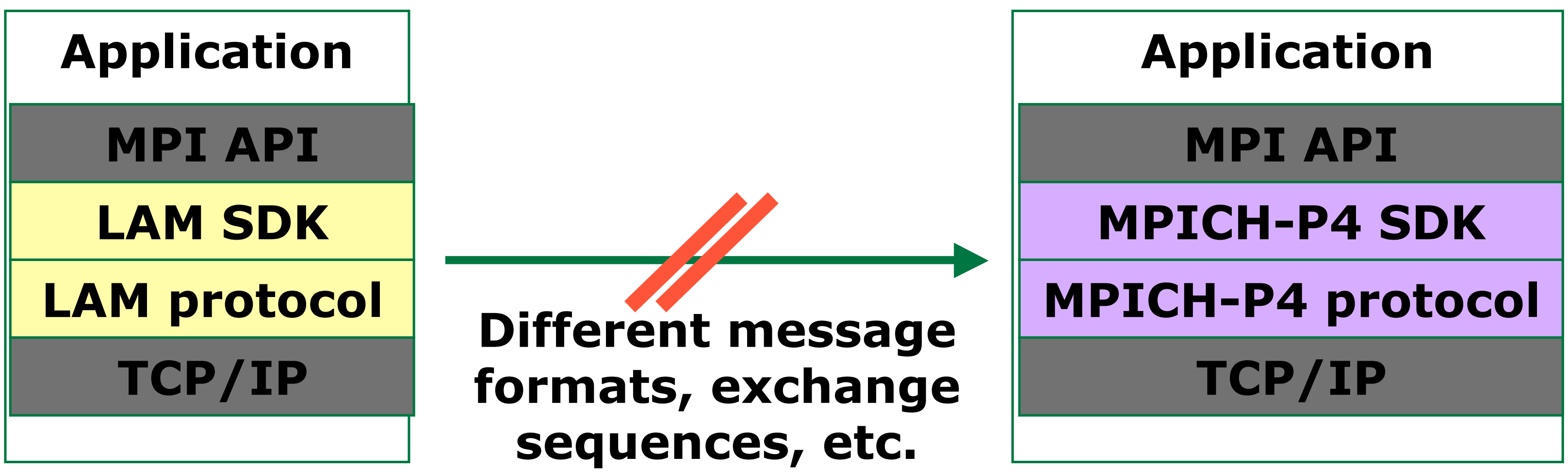
- An SDK consists of libraries and tools
  - Provides implementation of API specification
- One API can have multiple SDKs
- Examples of SDKs
  - MPICH

Protocols 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



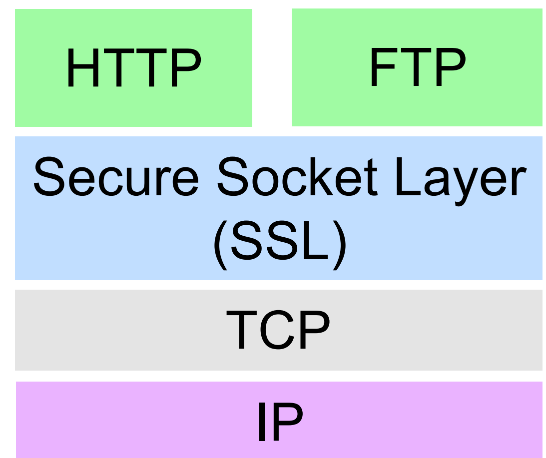
APIs and protocols are both important

- Standard APIs/SDKs are important
  - They enable application *portability*
    - Can move application to different places
  - But w/o standard protocols, interoperability is hard
  - Example: MPI
- Standard protocols are important
  - Between computers
  - Enable *interoperability*
    - Applications can talk to each other
  - Enable shared infrastructure – example: the internet
  - But w/o standard APIs/SDKs, application portability is hard (different platforms access protocols in different ways)

Security

Secure Sockets Layer SSL (TLS)

- Encrypted communications over Internet
  - Ensures that the information is sent unchanged, and only to the server you intended
- SSL uses a private key to encrypt data
  - Netscape and Internet Explorer support SSL
  - Web sites use SSL to obtain confidential user information, such as credit card numbers.
  - URLs that require an SSL connection start with https: instead of http:.
- Also known as Transport Level Security



OpenSSL is an open source implementation of SSL and TLS

- OpenSSL is used by:
  - Apache HTTP Server for https support
  - MySQL to provide secure database access.

OpenSSH is an implementation of the SSH protocol suite of tools

- Encrypts all traffic, including passwords
- Provides a variety of authentication methods.
- *Includes:*
  - `ssh` program - logins,
  - `scp` – copy files
  - `sftp` – general file transfer
- Also other basic utilities: ssh-add, ssh-agent, ssh-keygen

Security: Terminology

- Authentication: Establishing identity
- Authorization: Establishing rights
- Message protection
  - Message integrity
  - Message confidentiality
- Non-repudiation
- Digital signature
- Accounting
- Delegation

Authentication means identifying that you are whom you claim to be

- Authentication stops imposters
- *Examples of authentication:*
  - Username and password
  - Passport
  - ID card
  - Public keys or certificates
  - Fingerprint

Authorization is what you are allowed to do

- Is this device allowed to access to this service?
- Read, write, execute permissions in Unix
- ACLs provide more flexible control

Digital Signature

- An electronic signature that authenticates the identity of the sender of a message, the signer of a document, or ensures that the contents of a message are intact.
- Digital signatures are easily transportable, cannot be imitated by someone else, and can be automatically time-stamped.
- The ability to ensure that the original signed message arrived means that the sender cannot easily repudiate it later.

Digital Certificate

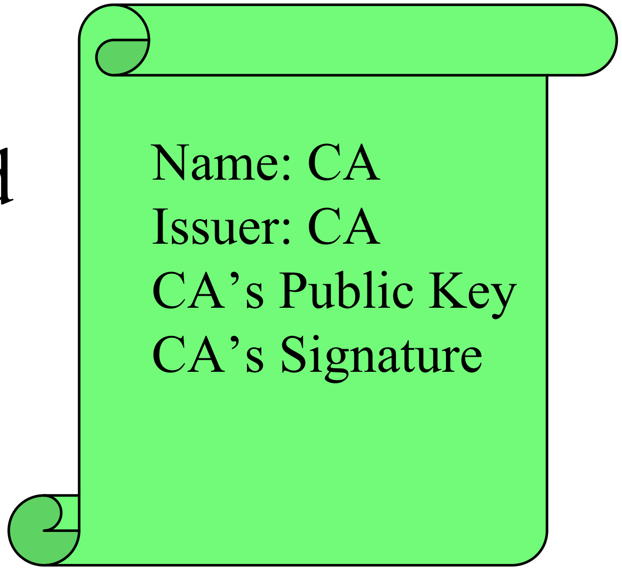
- The signer of a digital certificate says something like "I attached G.Allen's public key to this digital certificate and then signed it with my private key"
- Any user of G.Allen's digital certificate must completely trust the competency and honesty of the person/organization who signed the certificate
- For anyone to confidently use G.Allen's digital certificate they must also trust that they have a validated copy of the signers public key
- There is nothing secret about the contents of a digital certificate
- Has expiration date
- Analogy: Driving License issued by DMV (+ other countries)

Managing Digital Certificates

- Digital certificate administrative frameworks are called "public key infrastructures" (PKIs).
- Two major ones
  - X.509 (standardized by IETF)
  - Pretty Good Privacy (PGP)
- Centrally controlled system for managing digital certificates in X.509 talk is a "certificate authority"

Certificate Authorities (CAs) exist only to sign user certificates

- A small set of trusted entities
- A CA signs its own certificate
- The CA's certificate is distributed in a trusted manner



Hardware Components & Grids

Basic Elements

- Distributed systems built from
  - Computing elements (processors)
  - Communication elements (networks)
  - Storage elements (disk, attached or networked)
- New elements
  - Visualization/interactive devices
  - Experimental and operational devices

Distributed Resources

- Local workstations
- Site Resources
- Campus Resources
- State Resources
- National Centers
- National Grids (OSG, TeraGrid)
- International Grids (GGTC)

Compute Elements

- Clock speed
- Cache hierarchy
- Floating point registers
- Main memory
- Internal bandwidths
- ...
- Need powerful operating systems, compilers, applications to use all this

Supercomputers

- Definition of supercomputer
  - Machine on Top500.org?
  - Machine costing over $1M?
  - Most powerful machines
  - One-of-a-kind
- Top 3 (November 2006)
  1. IBM Blue Gene/L (US) 131k procs, 280 TF
  2. Cray Red Storm (US) 26k procs, 101 TF
  3. IBM BGW (US) 40k procs, 91 TF
- Top 3 (2003)
  - Earth Simulator (JAPAN) 5K procs/36 TF (6)
  - ASCI Q (USA) 8K procs/14 TF (12)
  - G5 Cluster (USA) 2k procs/12 TF (14)

Communication Elements

- Links, routers, switches, name servers, protocols
- Infrastructure evolves slowly (politics, large scale changes, money)
- Gilder's Law: total bandwidth of communication systems doubles every six months
- Change in LAN to desktops
  - 100 mbps shared
  - 100 mbps switched
  - 1 gbps
  - 10 gbps
- Clusters: Gigabit ethernet (TCP/IP and MPICH/LAM) standard, Myrinet (own MPI drivers) better performance

Network Speeds

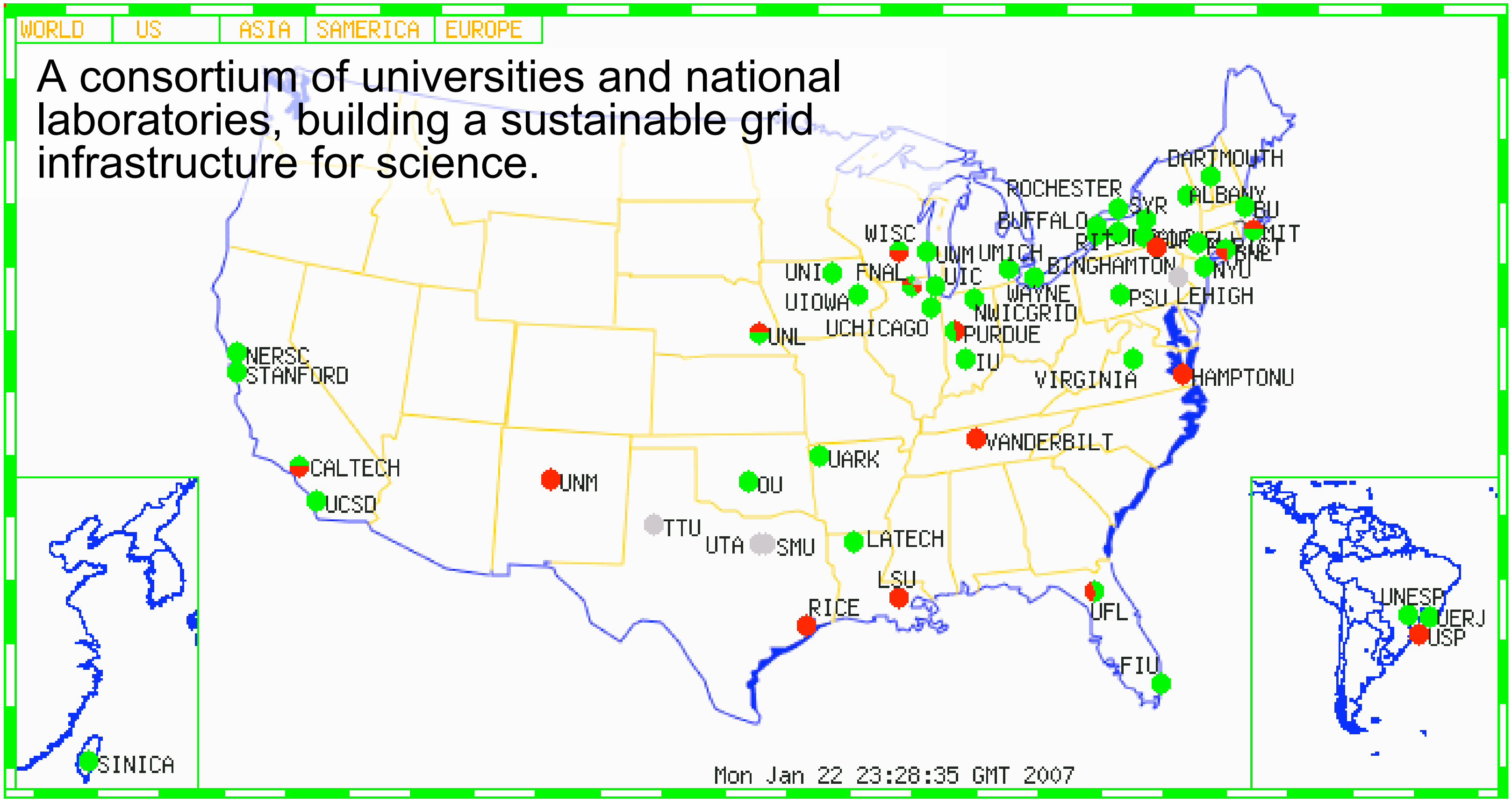
- Analog modem: 57 kbps
- GPRS: 114 kbps
- Bluetooth: 723 kbps
- T-1: 1.5 Mbps
- Eth 10Base-X: 10Mbps
- 802.11b (WiFi) 11 Mbps
- T-3: 45 Mbps
- OC-1: 52 Mbps
- Fast Eth 100Base-X: 100 Mbps
- OC-12: 622 Mbps
- GigEth 1000Base-X: 1 Gbps
- OC-24: 1.2 Gbps
- OC-48: 2.5 Gbps
- OC-192: 10 Gbps
- 10 GigEth: 10 Gbps
- OC-3072: 160 Gbps

- Home internet
  - Upload: 35 KB/s
  - Download 250 KB/s

Storage Elements

- Magnetic tape/Magnetic disk
- Magnetic disk
  - Properties: density/rotation/cost
  - 1970-1988 density improvements 29% per year
  - 1988-now density improvements 60% per year
  - Standard in PCs: 500mb (1995), 2gb(1997), 100gb (2002)
  - Performance not increasing so fast
    - Peak transfer (~100mbs)
    - Seek times (3-5ms) [bottleneck]
- Grids: cost of storage negligible, high speed networks make large data libraries attractive

Open Science Grid (OSG) provides shared computing resources, benefiting a broad set of disciplines



- OSG incorporates advanced networking and focuses on general services, operations, end-to-end performance

http://www.opensciencegrid.org/

Introduction to Grid Middleware

Globus Toolkit and Condor

- We will focus on Globus components as well as Condor in this workshop
- Globus tools can be used in different ways:
  - Client tools which you can use from a command line
  - APIs (scripting languages, C, C++, Java, …) to build your own tools, or use direct from applications
  - Web service interfaces
  - Higher level tools built from these basic components, e.g. RFT

Grid Security is hard but crucial

- Resources might be valuable
- Problems being solved might be sensitive
- Resources are located in distinct administrative domains
  - Each resource has own policies, procedures, security mechanisms, etc.
- Implementation must be broadly available & applicable
  - Standard, well-tested, well-understood protocols; integrated with wide variety of tools

Security Services

- Forms the underlying communication medium for all the services
- Secure Authentication and Authorization
- Single Sign-on
  - User explicitly authenticates only once – then single sign-on works for all service requests
- Uniform Credentials
- Ex: GSI (Grid Security Infrastructure)

Grid Security Infrastructure (GSI)

- Users:
  - Easy to use
  - Single sign-on: only type your password once
  - Delegate proxies
- Administrators:
  - Can specify local access controls
  - Have accounting

GSI builds on X.509 PKI

- PKI allows you to know that a given key belongs to a given user
- PKI builds off of asymmetric encryption:
  - Each entity has two keys: public and private
  - Data encrypted with one key can only be decrypted with other
  - The public key is public
  - The private key is known only to the entity
- The public key is given to the world encapsulated in a X.509 certificate (Grid Certificate)

A GSI certificate includes four pieces of information:

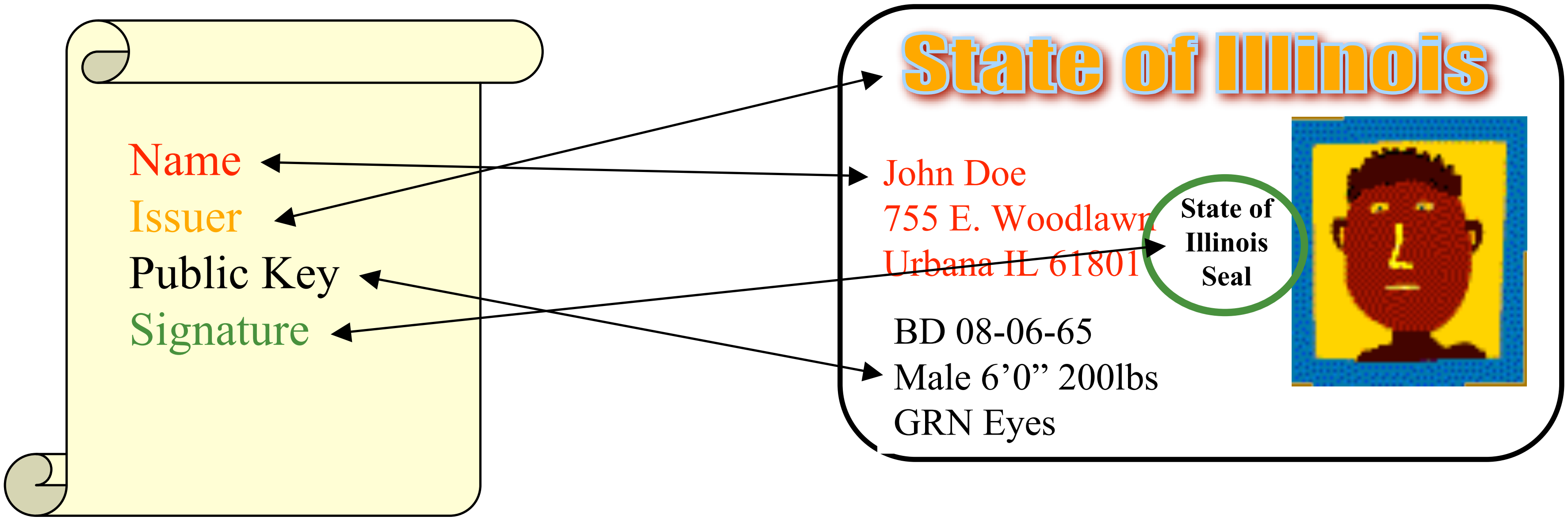
- Subject name
  - Identifies the person or object that the certificate represents
- The subject's *Public Key*
- Identity of the CA that signed the certificate
  - Certifies that the public key and the identity belong to the subject
  - Uses the digital signature of the named CA

Another CA certifies the link between the public key and the subject.

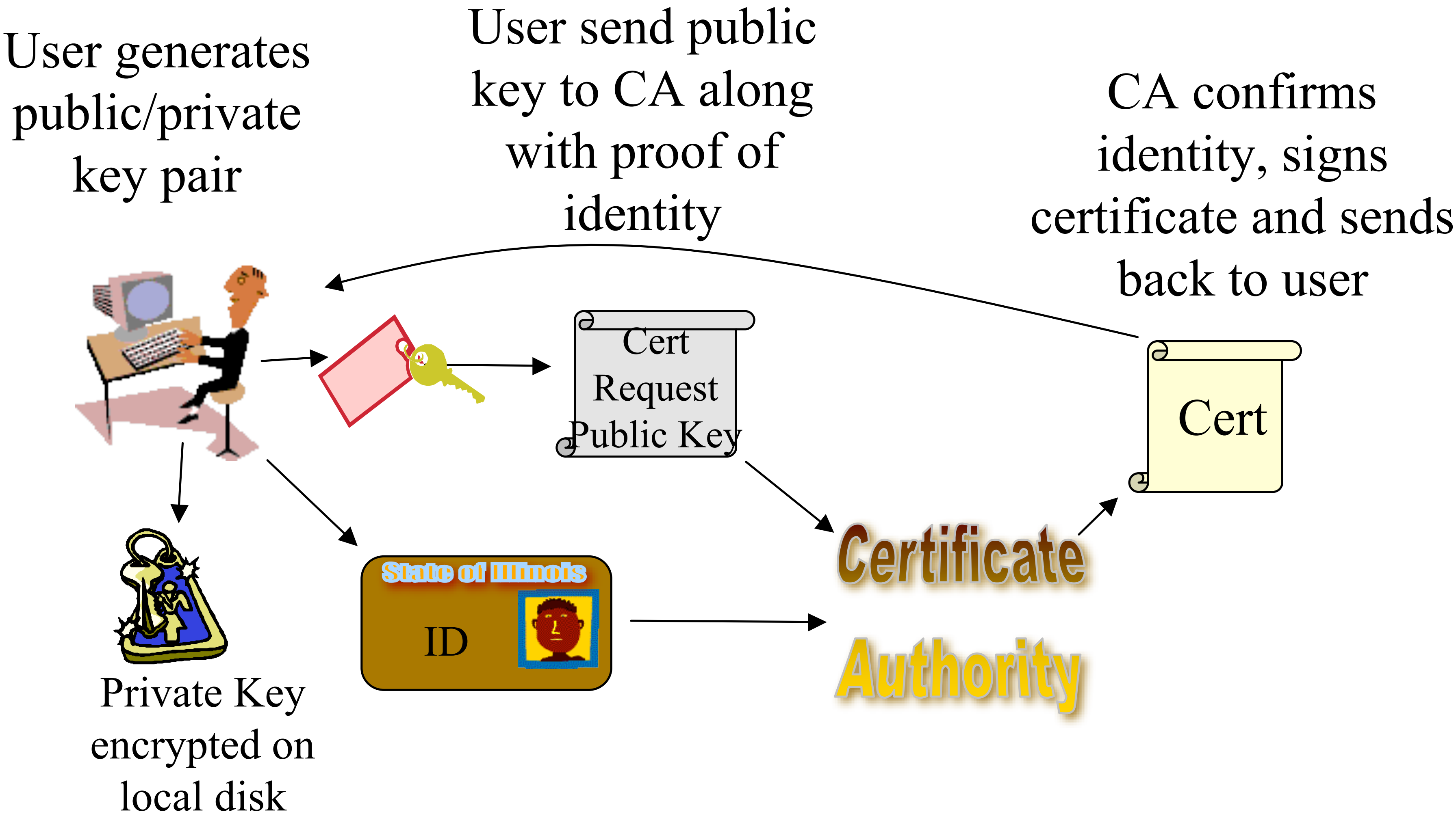
- To trust the certificate and its contents, the CA's certificate must be trusted.
- The link between the CA and its certificate must be established via a non-cryptographic method.

A Certificate is similar to a passport of driver's license

- Identity is signed by a trusted party



How Do You Get a Certificate?



Applications that use GSI

- Use GSI in Globus for:
  - Submitting jobs
  - Transferring data
  - Querying information services (often turned off)
- Other software using GSI:
  - Condor
  - GSI OpenSSH
  - MyProxy

Grid Monitoring & Information Services

To efficiently use a Grid, you must monitor its resources.

- Check the availability of different grid sites
- Discover different grid services
- Check the status of “jobs”
- Make better scheduling decisions with information maintained on the “health” of sites

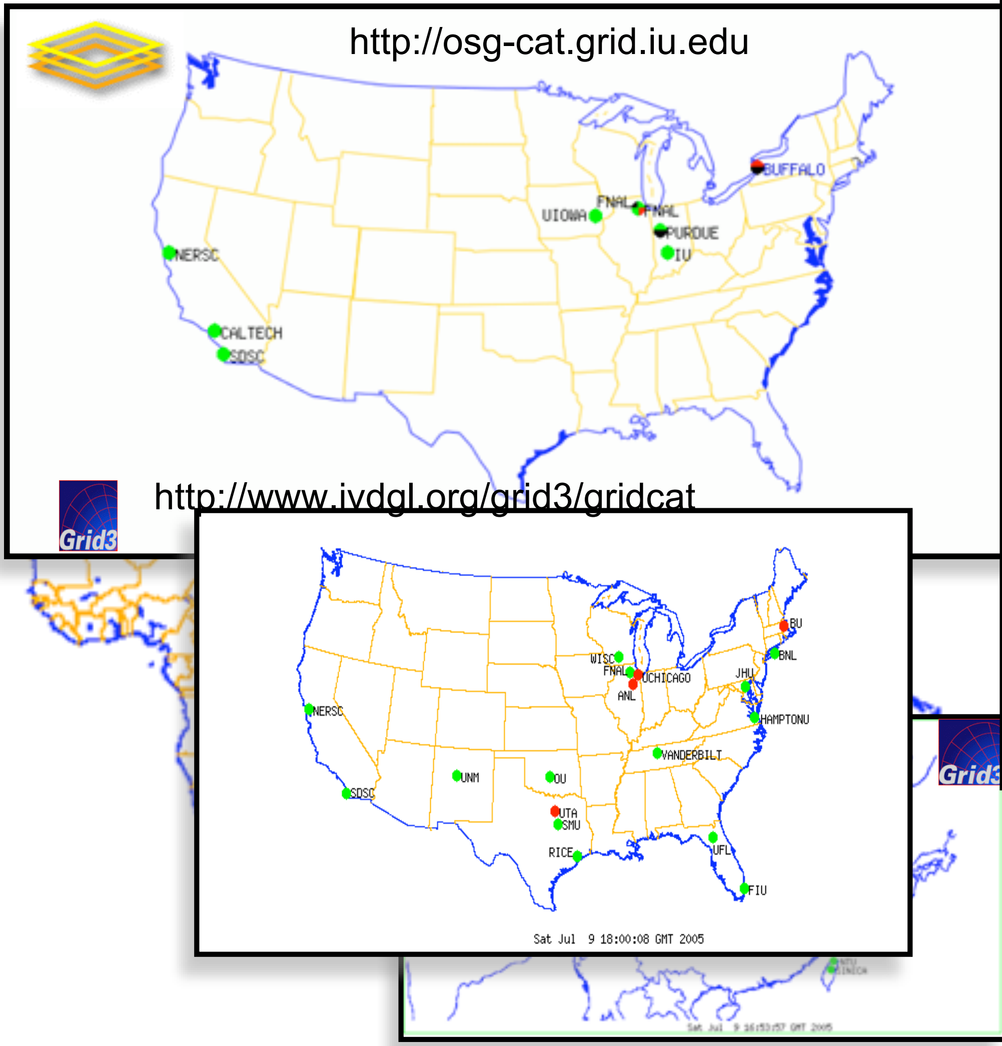
Monitoring provides information for several purposes

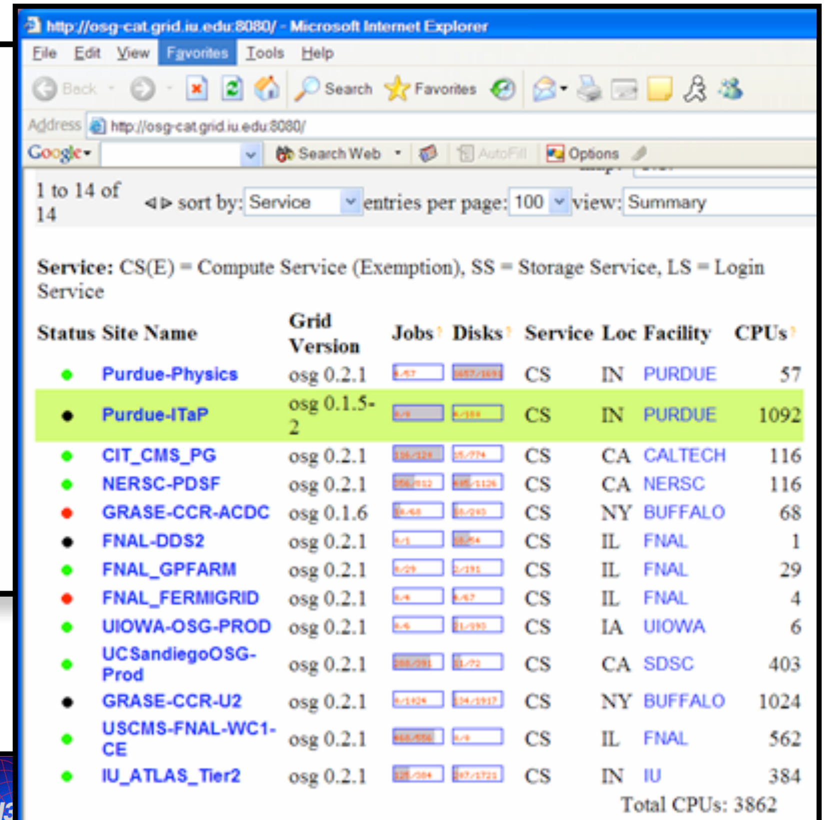
- Operation of Grid
  - Monitoring and testing Grid
- Deployment of applications
  - What resources are available to me? (Resource discovery)
  - What is the state of the grid? (Resource selection)
  - How to optimize resource use? (Application configuration and adaptation)
- Information for other Grid Services to use

Monitoring information is either static or dynamic, broadly.

- Static information about a site:
  - Number of worker nodes, processors
  - Storage capacities
  - Architecture and Operating systems
- Dynamic information about a site
  - Number of jobs running on each site
  - CPU utilization of different worker nodes
  - Overall site "availability"
- Time-varying information is critical for scheduling of grid jobs
- More accurate info costs more: it's a tradeoff.

GridCat



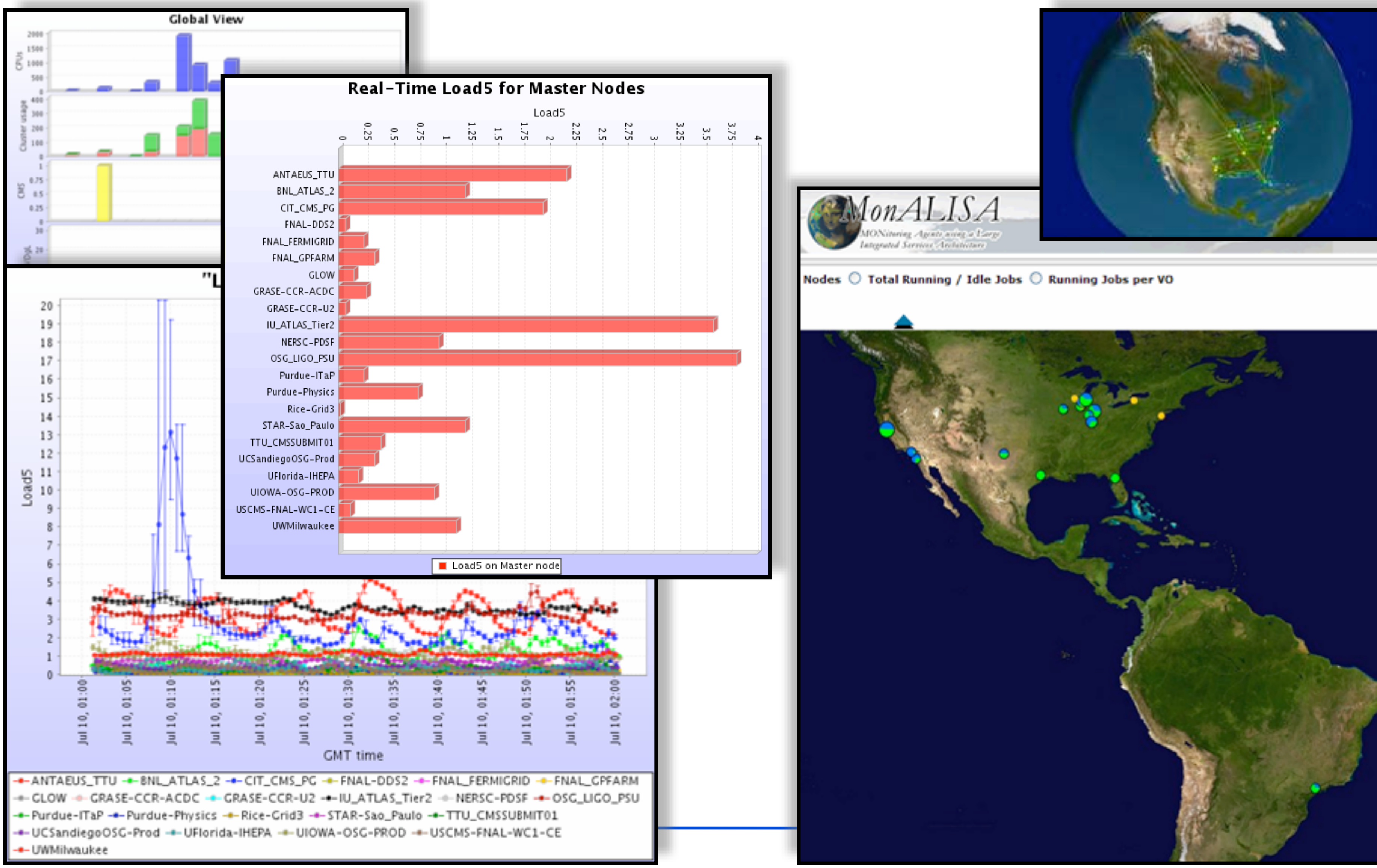


Status	Site Name	Grid Version	Jobs	Disks	Service	Loc	Facility	CPUs
●	Purdue-Physics	osg 0.2.1			CS	IN	PURDUE	57
●	Purdue-ITaP	osg 0.1.5-2			CS	IN	PURDUE	1092
●	CIT_CMS_PG	osg 0.2.1			CS	CA	CALTECH	116
●	NERSC-PDSF	osg 0.2.1			CS	CA	NERSC	116
●	GRASE-CCR-ACDC	osg 0.1.6			CS	NY	BUFFALO	68
●	FNAL-DDS2	osg 0.2.1			CS	IL	FNAL	1
●	FNAL_GPFARM	osg 0.2.1			CS	IL	FNAL	29
●	FNAL_FERMIGRID	osg 0.2.1			CS	IL	FNAL	4
●	UIOWA-OSG-PROD	osg 0.2.1			CS	IA	UIOWA	6
●	UCSandiegoOSG-Prod	osg 0.2.1			CS	CA	SDSC	403
●	GRASE-CCR-U2	osg 0.2.1			CS	NY	BUFFALO	1024
●	USCMS-FNAL-WC1-CE	osg 0.2.1			CS	IL	FNAL	562
●	IU_ATLAS_Tier2	osg 0.2.1			CS	IN	IU	384

Total CPUs: 3862

http://www.ivdgl.org/gridcat/home/

MonALISA



http://monalisa.caltech.edu/

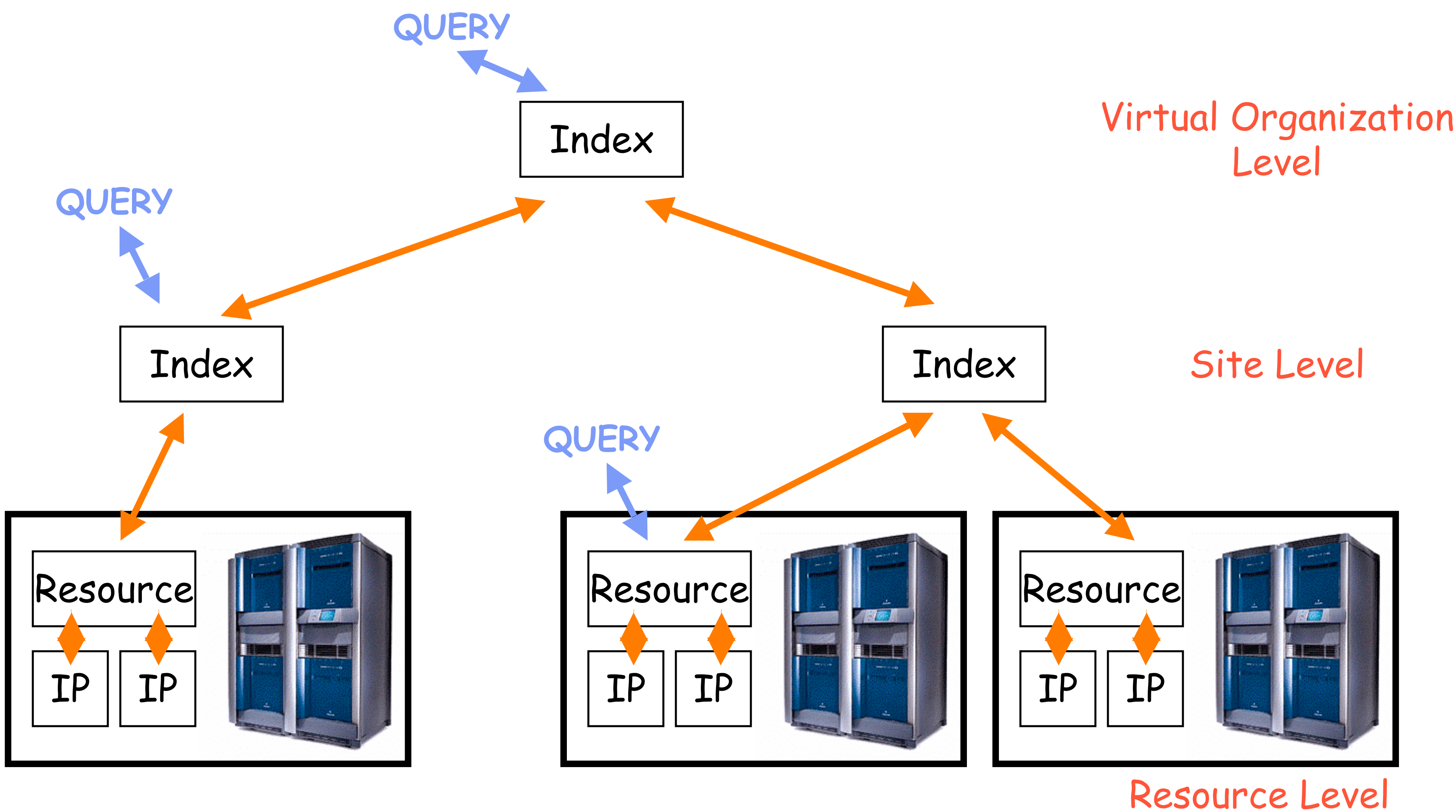
Globus Monitoring and Discovery System

- MDS is a grid information service
- It provides:
  - Uniform, flexible access to information
  - Scalable, efficient access to dynamic data
  - Access to multiple information sources
  - Decentralized maintenance

Globus MDS

- Handles static (e.g OS type) and dynamic (e.g current load) data
- Access to data can be restricted with GSI (Grid Security Infrastructure) credentials and authorization features

MDS Hierachy



Data Management

Data management services provide a flexible mechanism to move and share data

- Grids are used for analyzing and manipulating large amounts of data
  - Metadata (data about data): *What is the data?*
  - Data location: *Where is the data?*
  - Data transport: *How to move the data?*

Data Movement

- Issues
  - How to move data
    - Robustly
    - Securely
    - Faster
- Solutions
  - scp, globus-url-copy, wget
  - GridFTP

GridFTP is a secure, efficient and standards-based data transfer protocol

- Robust, fast and widely accepted
- Globus GridFTP server
- Globus *globus-url-copy* GridFTP client
- Other clients exist (e.g., *uberftp*)

GridFTP is secure, reliable and fast

- Security through GSI
  - Authentication and authorization
  - Can also provide encryption
- Reliability by restarting failed transfers
- Fast
  - Can set TCP buffers for optimal performance
  - Parallel transfers
  - Striping (multiple endpoints)
- Not all features easily accessible from basic client

File catalogues tell you where the data is

- Replica Location Service (RLS)
- Phedex
- RefDB / PupDB

Requirements from a File Catalogue

- Abstract out the logical file name (LFN) for a physical file
  - maintain the mappings between the LFNs and the PFNs *(physical file names)*
- Maintain the location information of a file

In order to avoid "*hotspots*", replicate data files in more than one location

- Effective use of the grid resources
- Each LFN can have more than 1 PFN
- Avoids single point of failure
- Manual or automatic replication
  - Automatic replication considers the demand for a file, transfer bandwidth, etc.

The Globus Replica Location Service (RLS)

- Each RLS server usually runs
  - *Local Replica Catalog (LRC)*
    - What files do you have (directly know physical location), mapped to URLs or physical file names (*PFN*)

  and/or

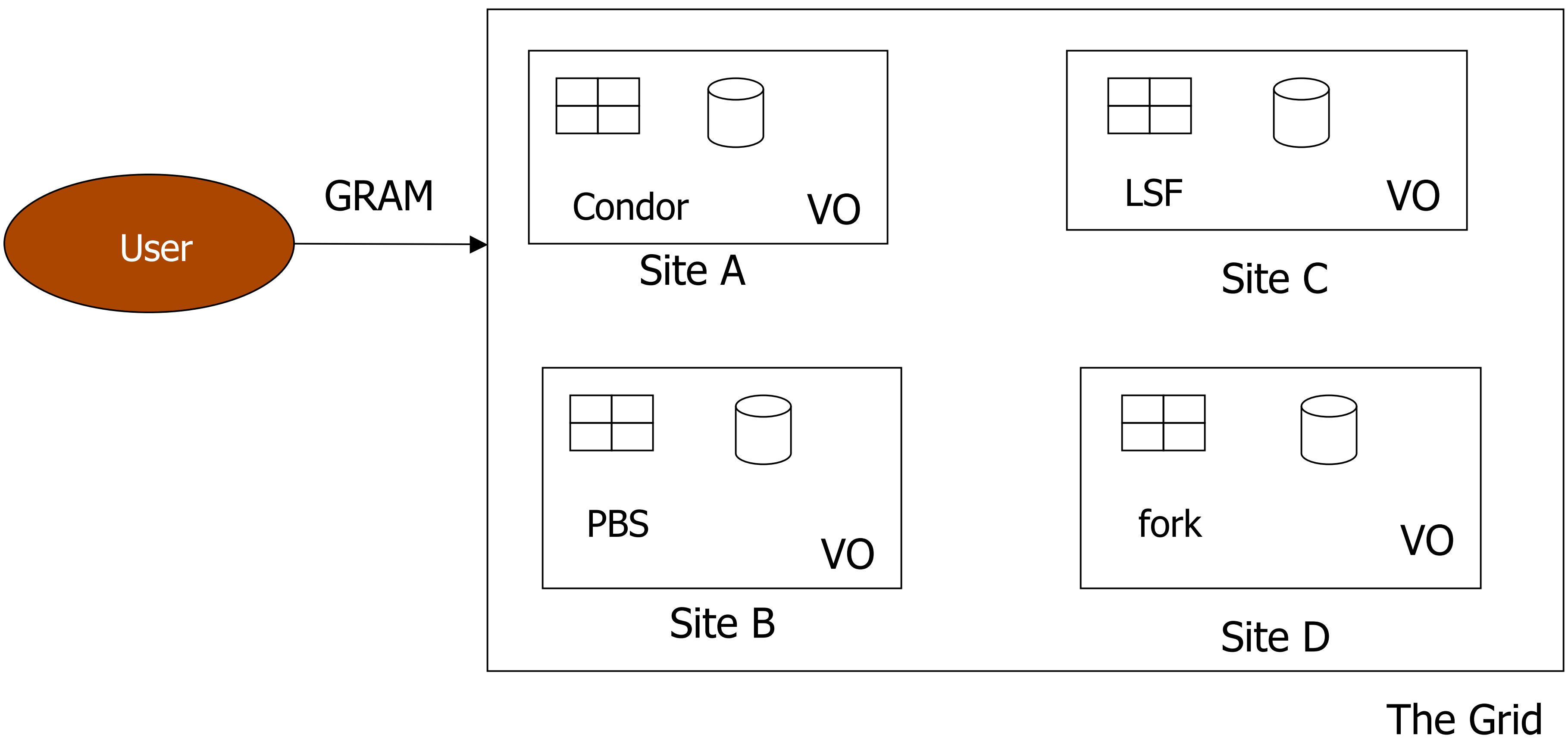
  - *Replica location index (RLI)*
    - Catalog of what LFNs other LRCs know about
- Similar hierarchical structure to MDS.

Job Management

Job Management Services provide a standard interface to remote resources

- Includes CPU, Storage and Bandwidth
- Main component is the remote job manager
  - *Globus Resource Allocation Manager (GRAM)*
- Other needs:
  - scheduling
  - monitoring
  - job migration
  - notification

Job Management on a Grid



GRAM: What is it?

- *Globus Resource Allocation Manager*
- Given a job specification:
  - Create an environment for a job
  - Stage files to and from the environment
  - Submit a job to a local resource manager
  - Monitor a job
  - Send notifications of the job state change
  - Stream a job’s stdout/err during execution

A "Local Resource Manager" is a batch system for running jobs across a computing cluster

- In GRAM
- *Examples:*
  - Condor
  - PBS
  - LSF
  - Sun Grid Engine
- Most systems allow you to access "fork"
  - Default behavior
  - It runs on the gatekeeper:
    - A bad idea in general, but okay for testing

The client describes the job in with GRAM's Resource Specification Language (RSL)

- Example:

```
& (executable = a.out)
  (directory = /home/nobody )
  (arguments = arg1 "arg 2")
```

- Use higher level tools (such as portals) to construct anything but simple RSL
- See http://www.globus.org/gram/rsl_spec1.html

Managing your jobs

- We need something more than just the basic functionality of the globus job submission commands
- Some desired features
  - Job tracking
  - Submission of a set of inter-dependant jobs
  - Check-pointing and Job resubmission capability
  - Matchmaking for selecting appropriate resource for executing the job
- *Options:* Condor, PBS, LSF, …

The Problem of Grid Scheduling

- Decentralised ownership
  - No one controls the grid
- Heterogeneous composition
  - Difficult to guarantee execution environments
- Dynamic availability of resources
  - Ubiquitous monitoring infrastructure needed
- Complex policies
  - Issues of trust
  - Lack of accounting infrastructure
  - May change with time

Based on:

Grid Intro and Fundamentals Review

Dr Gabrielle Allen

Center for Computation & Technology

Department of Computer Science

Louisiana State University

gallen@cct.lsu.edu

Grid Summer Workshop

June 26-30, 2006