Principles & Architectures of Distributed Computation

Steven Newhouse

Contents

- Examples of Distributed (Grid) Computing
- Fallacies of Distributed (Grid) Computing
- Standards for Distributed Computing

What is Grid Computing?

Grid computing involves sharing heterogeneous resources (based on different platforms, hardware/software architectures, and computer languages), located in different places belonging to different administrative domains over a network using open standards.

Grids In A Nutshell

- Co-ordinated use of shared resources across multiple administrative domains amongst a dynamic user community.
- Evolution of various forms distributed networked computing (HPC, Data,…)
- Various resources: compute, data, storage, instruments, sensors, etc.
- Very diverse scales & skills of use.

BUT what is important… is what *you* can do with them!
e-Research/e-Science/e-Industry

Grid Computing: A broad spectrum

- Volunteer Computing
  - ClimatePrediction.net
- Co-operative Resource Pools
  - GENIE
- Large-scale Computing & Data Grids
  - EGEE
- Federated Super-Computers
  - DEISA

ClimatePredicition.net

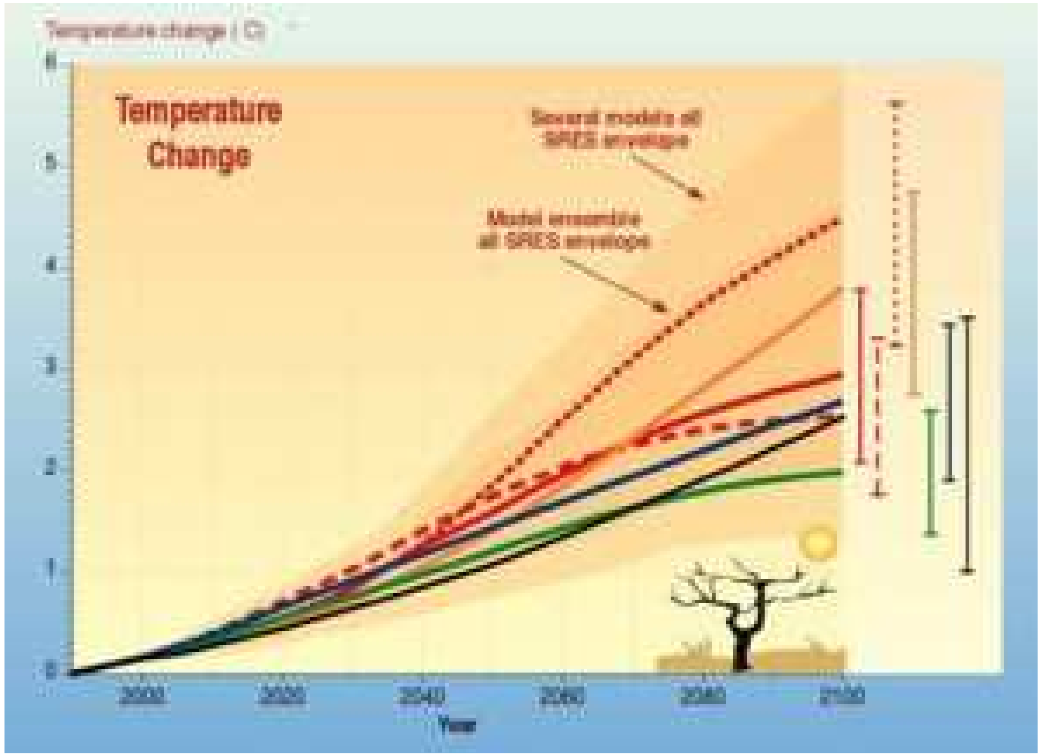
- Climate models are dependent on many parameters and assumptions
- Need to understand the sensitivity of these parameters &

  assumptions
- Explore through

  repeated

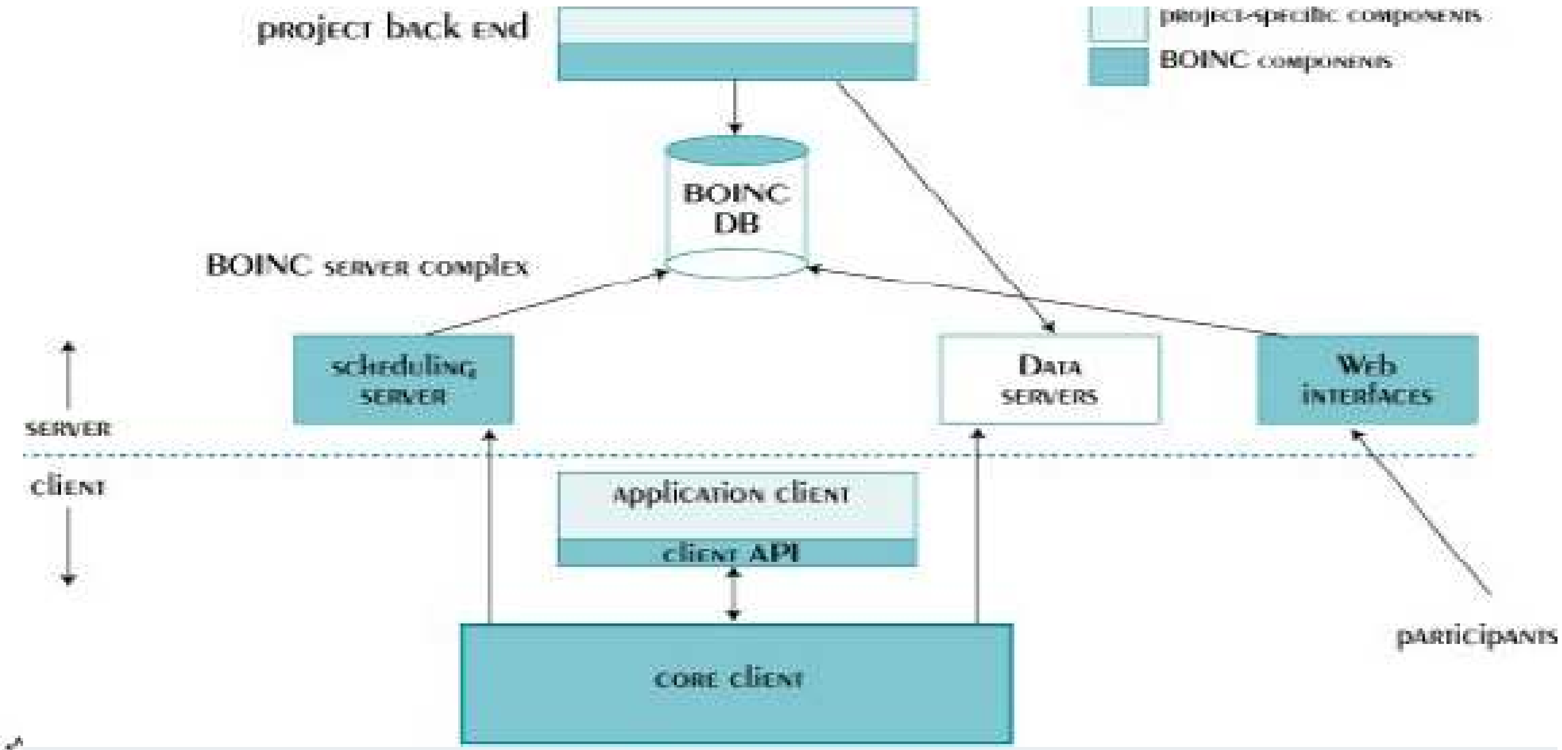
  computational runs

©



BOINC: http://boinc.berkeley.edu

Berkley Open Infrastructure for Network Computing



Security Issues: Participants

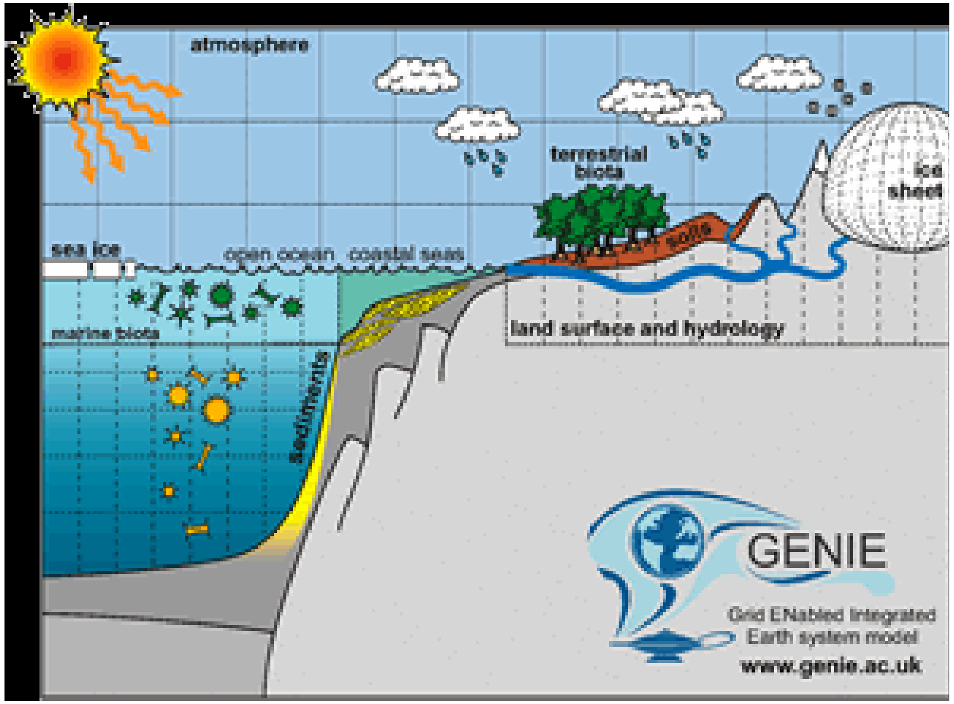
- Software package is digitally signed.
- Communications are always be initiated by the client.
- HTTP over a secure socket layer will be used where necessary to protect participant details and guarantee reliable data collection.
- Digitally signed files can be used where necessary.

Security Issues: Experiment

- Two types of run replication:
  - Small number of repeated identical runs.
  - Large numbers of initial condition ensembles.
- Checksum tracking of client package files to discourage casual tampering.
  - Opportunity to repeat runs as necessary.
  - Server security management and frequent backups.

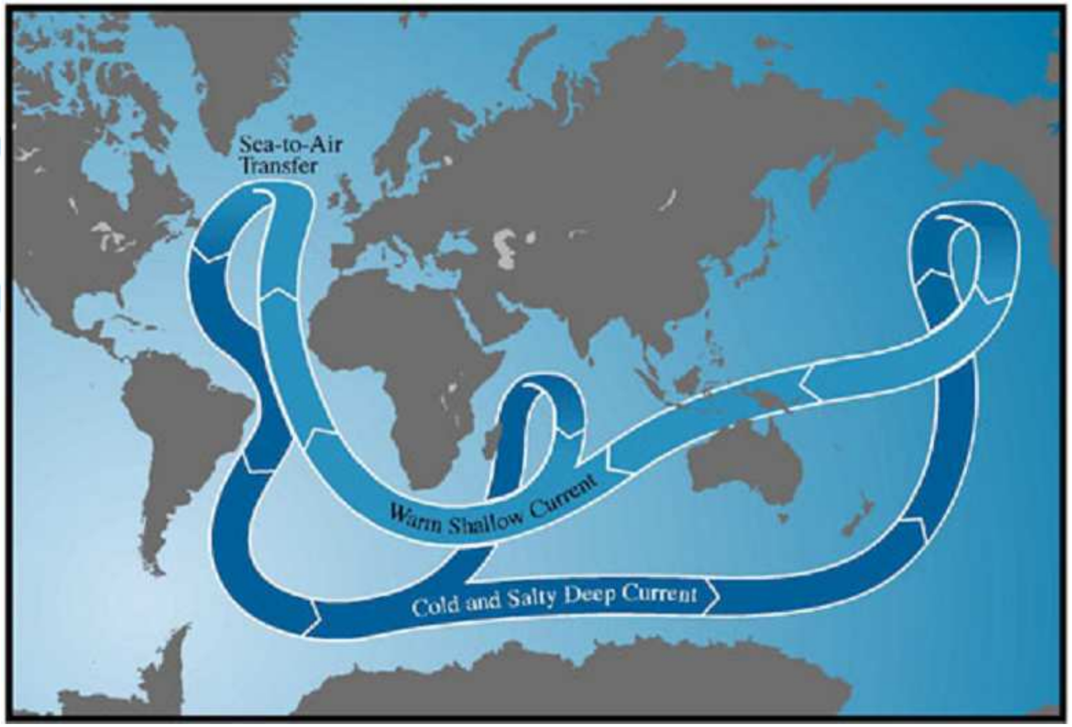
GENIE – http://www.genie.ac.uk

- Flexibly couple together state-of-the-art components to form a unified Earth System Model (ESM)
- Execute the resulting ESM across a computational Grid
- Share the distributed data produced by simulation runs
- Provide a high-level open access to the system, creating and supporting virtual organisations of Earth System modellers



The Problem: Thermohaline circulation

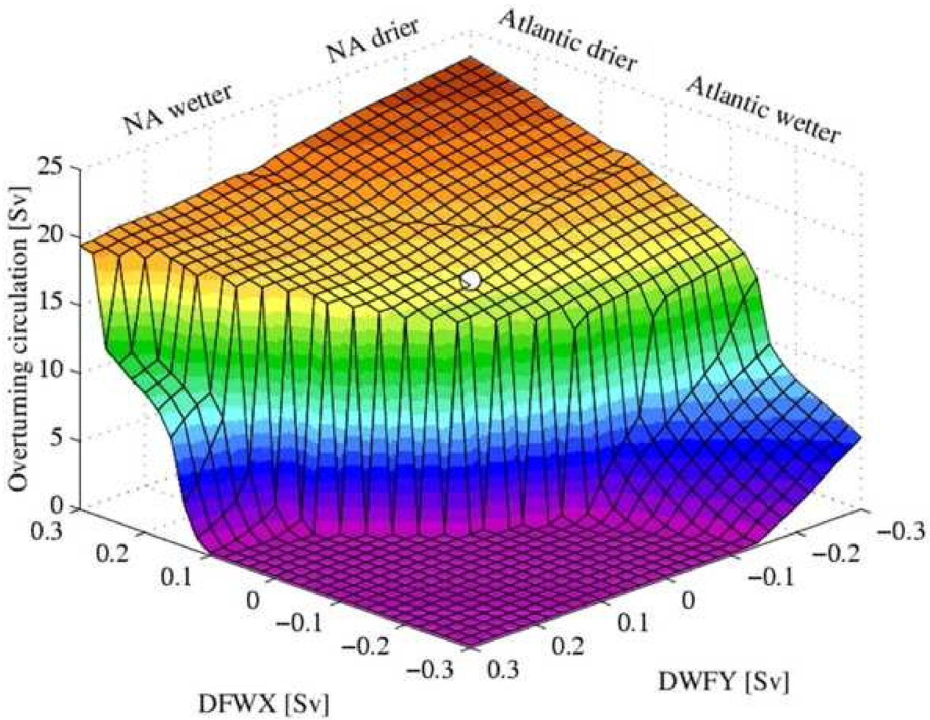
- Ocean transports heat through the "global conveyor belt."
- Heat transport controls global climate.
- Wish to investigate strength of model ocean circulation as a function of two external parameters.
- Use GENIE-Trainer.



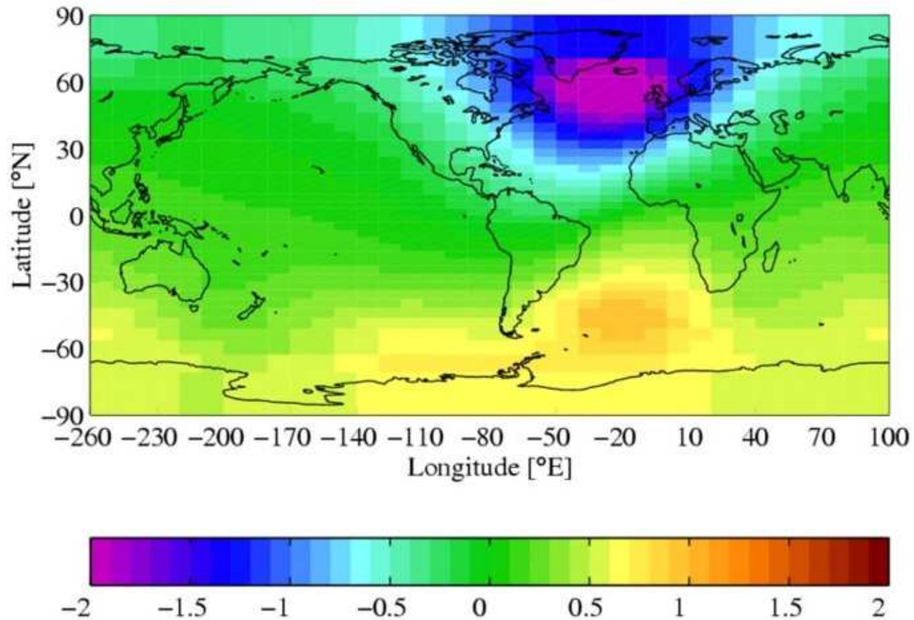
- Wish to perform $31 \times 31 = 961$ individual simulations.
- Each simulation takes ~4 hours to execute on typical Intel P3/1GHz, 256MB RAM, machine $\Rightarrow$

time taken for 961 sequential runs ≈ 163 days!!!

The Results:
Scientific Achievements



Intensity of the thermohaline circulation as a function of freshwater flux between Atlantic and Pacific oceans (DFWX), and mid-Atlantic and North Atlantic (DFWY).



Surface air temperature difference between extreme states (off - on) of the thermohaline circulation.

North Atlantic 2°C colder when the circulation is off.

time taken for 961 runs over ~200 machines ≈ 3 days

Extensive use of Condor

- Condor pools at:
  - Imperial College London
  - Southampton
- Evolving Infrastructure
  - Initially very simple Condor job
  - Portal interface to define & start Condor job
  - Improved data retrieval and visualisation interfaces

What Condor provides…

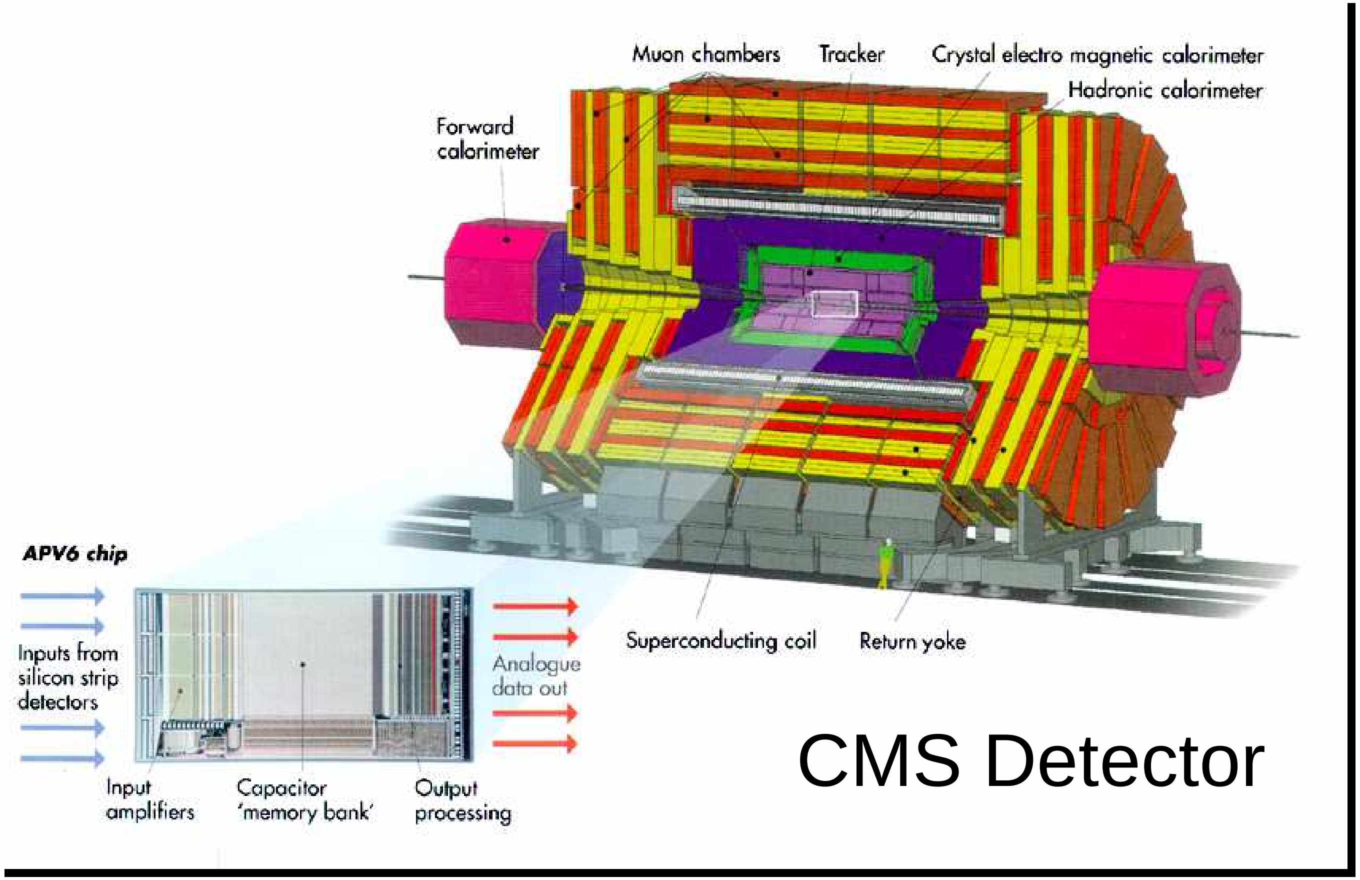
- Transparent execution environment
- Sandboxing to support remote execution
- Scheduling within & between pools
- Information on available resources

Commodity Production Computing & Data Grid

- A production grid is (inevitably) a collection of solutions customised for a particular scenario:
  - Scalability
  - Reliability
  - Authentication, Authorisation & Accounting
  - Portability
  - Customer satisfaction

CERN: Large Hadron Collider (LHC)

Raw Data: 1 Petabyte / sec
Filtered 100Mbyte / sec = 1 Petabyte / year = 1 Million CD ROMs

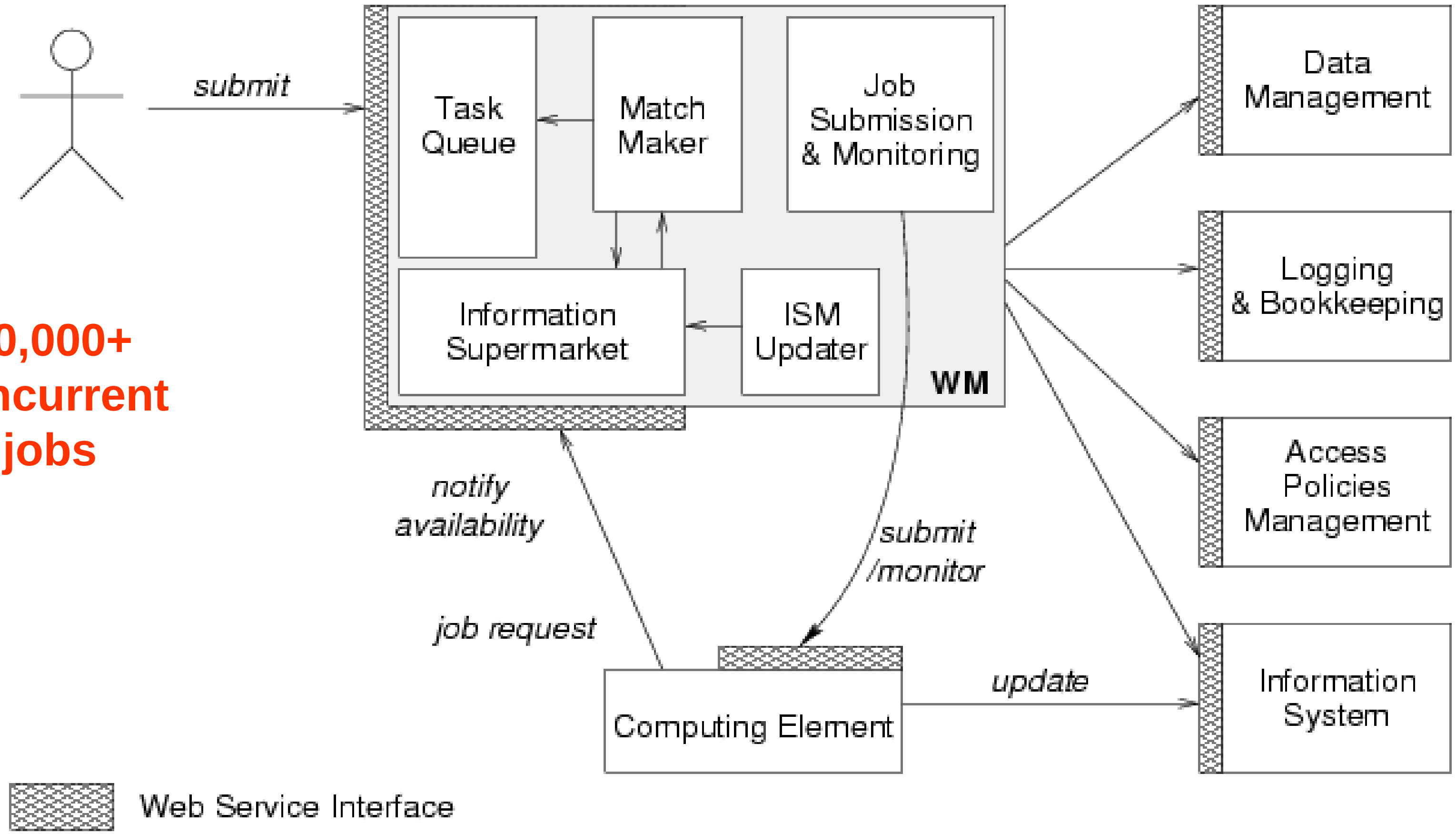


CMS Detector

EGEE: A Production Grid

- Enabling Grids for E-SciencE
- EU project – Funded until March 2008
  - EDG (European Data Grid): 2001-2004
  - EGEE-I: 2004-2006
- Focus on production infrastructure
  - Multiple VO's – large & small
  - Grid Operational Centres – to monitor infrastructure
- Its big & growing
  - 100+ sites, 10K+nodes, 6PB+ storage
- http://www.eu-egee.org

Workload Management System



10,000+ concurrent jobs

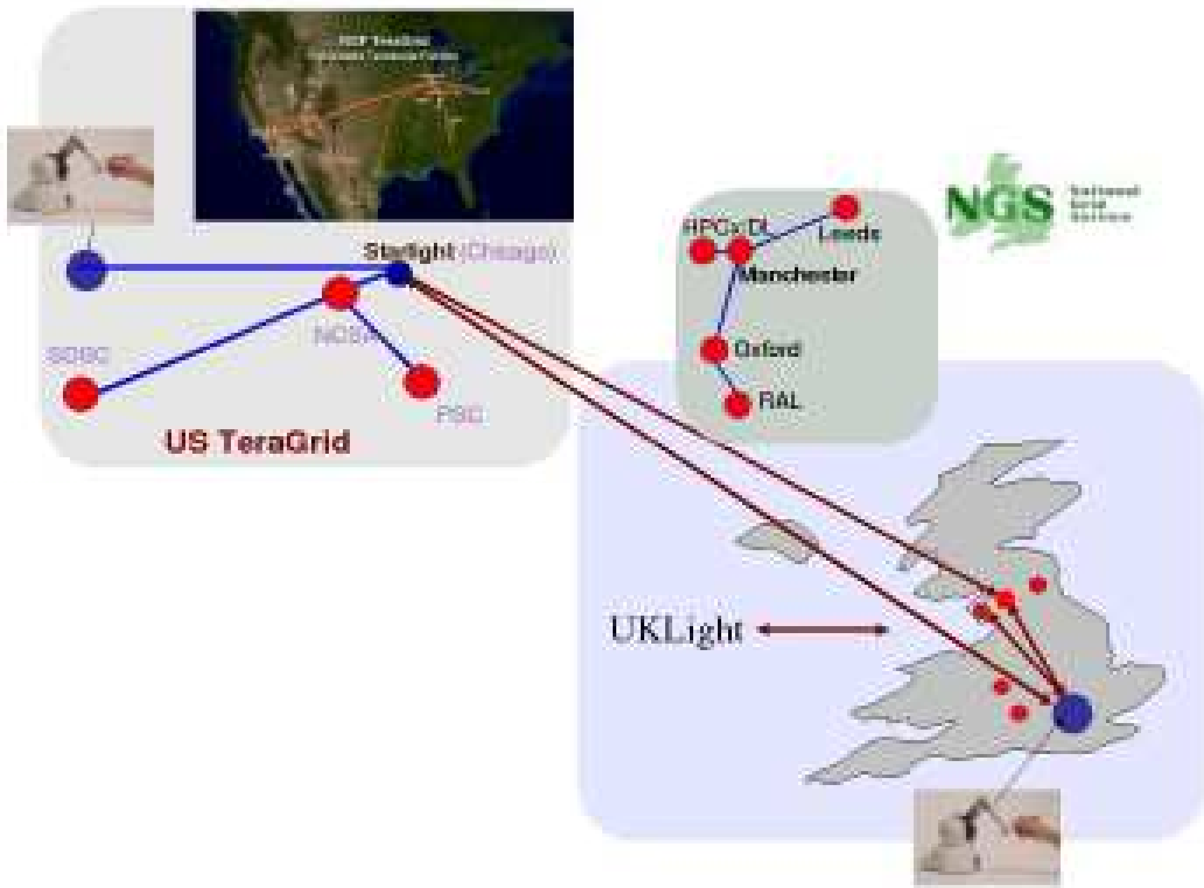
EGEE Integrated Software (gLite): Condor, Globus + EGEE

Super-computing Grids

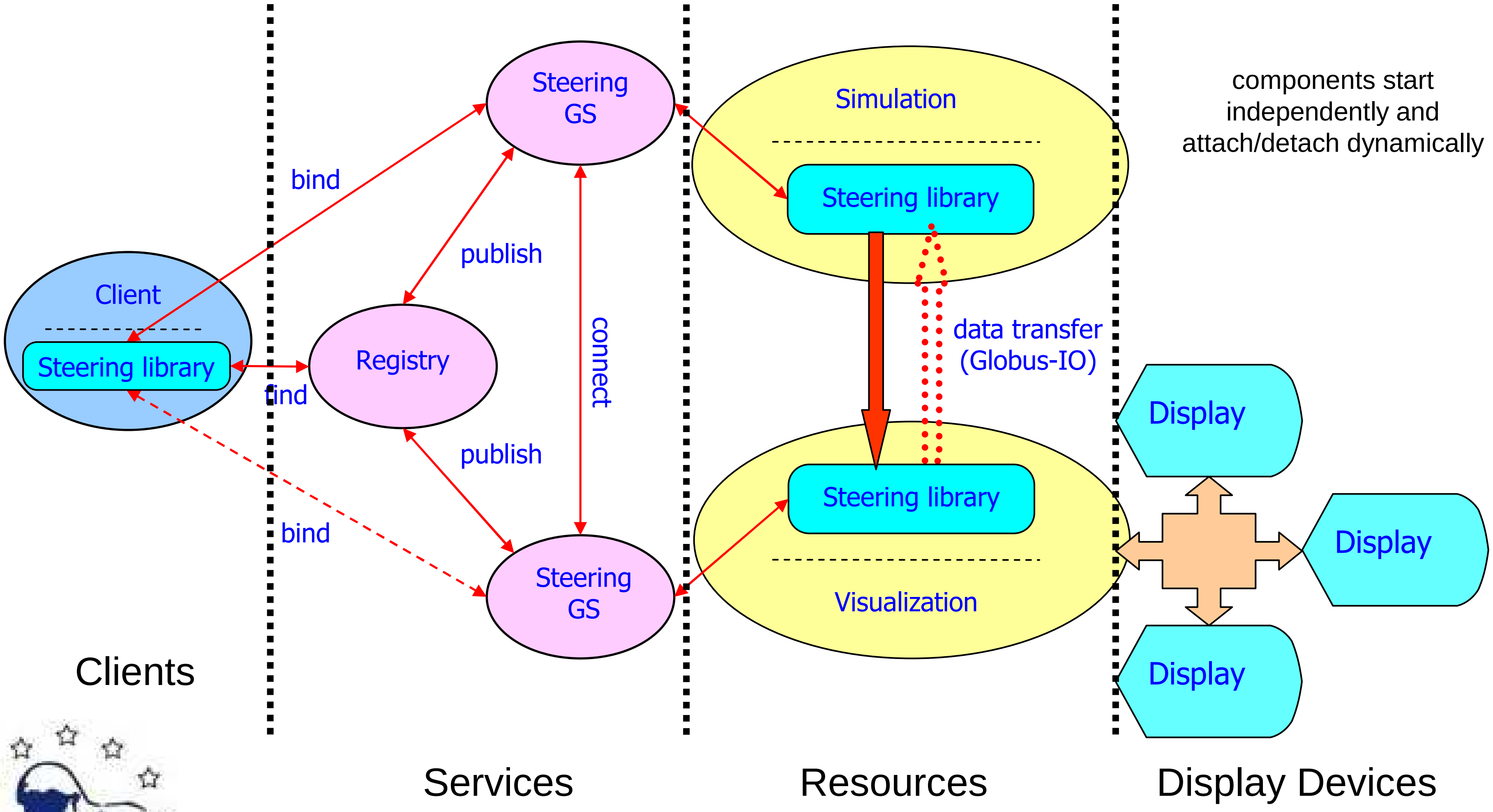
- DEISA – Distributed European Infrastructure for Super-computing Applications
  - Linked IBM (AIX) machines across Europe
    - Expansion to other systems during the project
  - Standard submission interface from Unicore
  - Global Parallel File System (GPFS)
  - Dedicated networking infrastrucure
- USA TeraGrid & UK NGS
  - Pre-web service GT4 components

SPICE: Simulated Pore Interactive Computing Environment (Coveney *et al*)

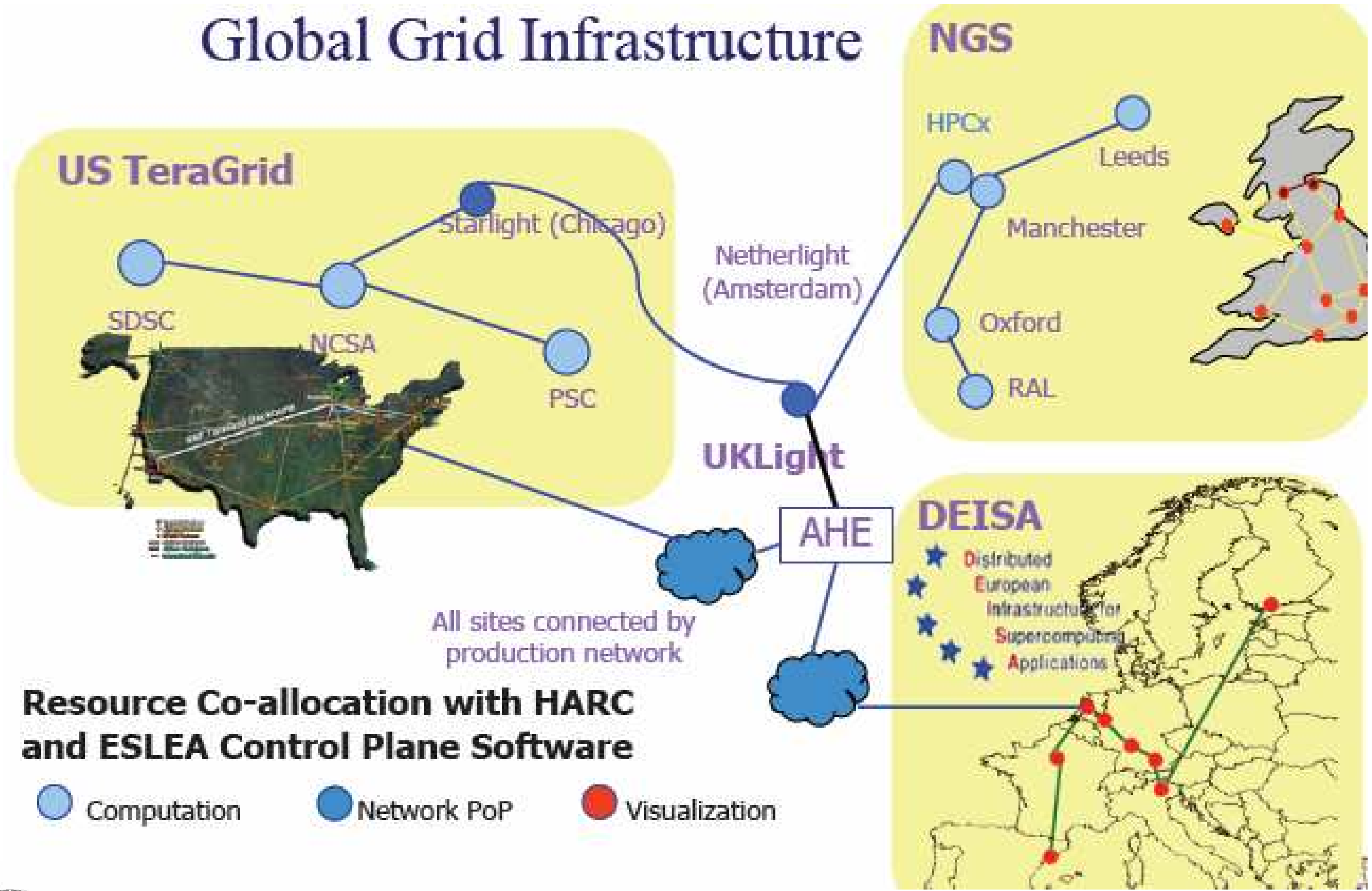
- NGS & TeraGrid during SuperComputing 05



SPICE Architecture



The Future for HPC Grids



So where does that get us…?

- Discovery:
  - Condor Collector, SPICE Registry
  - glite Information Index, BOINC Task Distribution
- Selection:
  - Condor Matchmaker
  - gLite Workload Management System
  - Unicore Abstract Job Objects
- Execution:
  - Condor, pre-WS GRAM, GridSAM, Unicore

The Eight Fallacies of Distributed Computing (and the Grid)

1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

I also interpret the ‘network’ as the ‘things’ I want to connect to

From Deutsch & Gosling

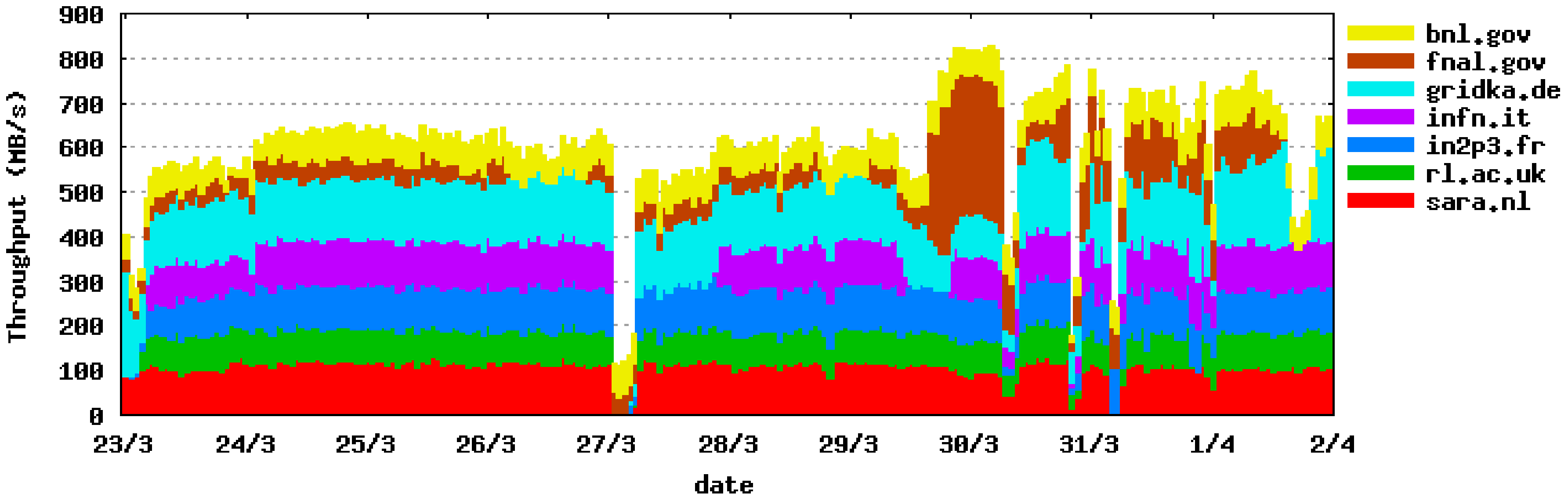
The network is reliable

- Services are here today & gone tomorrow
  - DNS provides a level of abstraction above IPs
  - Defensive programming – test failure & success
- 'Well known' registries collect information
  - GT4 Index Service, Condor Collector, gLite II
- Good Naming schemes protect against failure
  - Human → Abstract → Address (e.g. Services)
  - Services can migrate to new locations
- Condor uses stateless UTP protocol
  - Very good recovery behaviour following failure

Latency is zero

- Services calls can take a long time to complete
  - Run an HPC climate simulation (~ several days)
- Build in asynchronous communication
  - Start an activity
  - Subscribe to a notification interface
  - Messages delivered to you following ‘events’
  - Firewalls mean that ‘polling’ for state changes is still an option

Bandwidth is infinite



- LHC Service Challenges (2005)
  - Demonstrate sustained transfers from Tier 0 → 1
  - 600MB/s daily average over 10 days

The Network is Secure

- Need to enable sharing through trust
  - Need to know who is accessing the resource
  - Need to control who can access the resource
  - Need to know how much of the resource they used
  - Need to know what they did and when
- Avoid tampering
  - Digital signatures (also WS-Security)
- Provide confidentiality
  - https & WS-SecureConversation

Topology does not change

- Abstracts layers provide some protection
  - Naming
- Some simple autonomic behaviour
  - GridFTP configurable TCP parameters & channels
- Don’t make assumptions on topology in the design!

There is one administrator

- Clearly many, many, many administrators
  - Networks are partitioned & owned
  - Local Resources are ‘owned’
- Virtual Organisation models critical
  - Provide a virtual administration point
  - Used by local administrators to configure resources

Transport cost is zero

- Cost has many interpretations…
  - For reliable data transfers need dedicated networks
    - UK Light – 10GB optical network
  - More flexibility by negotiating quality of services
    - Some protocol support within networks… between difficult
  - Heavy users will have to pay
    - Dedicated infrastructure → $$$

The network is homogeneous

- Heterogeneity assumed in most middleware
  - Low/High encoded Endian-ness for data
  - Growing use of Web Services
  - XML encoding for messages
  - Java a common development environment
- Use of open standards for virtualisation
  - Well defined ‘application’ service interfaces
    - Standards Organisations: Open Grid Forum
  - Widely adopted Web Service plumbing
    - OASIS (Organisation for the Advancement of Structured Information Standards): WS-ResourceFramework

Defining Open Standards

- Bespoke proprietary protocols do not lead to interoperability but wrapping
  - Condor: Native communication library
  - GT4: WS-ResourceFramework
    - WS-GRAM for job submission
  - UNICORE: ****
  - gLite: Own protocol & interfaces but use others:
    - Globus (pre WS-GRAM) & Condor-C
- There has to be a better way….

Three Generations of Grid

1

- Local “metacomputers“
  – Distributed file systems
  – Site-wide single sign-on
- "Metacenters" explore inter-organizational integration
- Totally custom-made, top-to-bottom: proofs-of-concept

Source: Charlie Catlett

Three Generations of Grid

1

- Local “metacomputers“
  - 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:
  - *Condor, Globus, UNICORE, Legion, etc.*
- Need significant customization to deliver complete solution
- Interoperability is still very difficult!

Source: Charlie Catlett

Three Generations of Grid

1

- Local “metacomputers“
  - 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:
  - *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**

Source: Charlie Catlett

Three Generations of Grid

1
- Local “metacomputers“
  - Distributed file systems
  - Site-wide single sign-on
- "Metacenters" explore inter-org nal integration
- Totally custom-made, top-to ofs-of-concept

We are here!

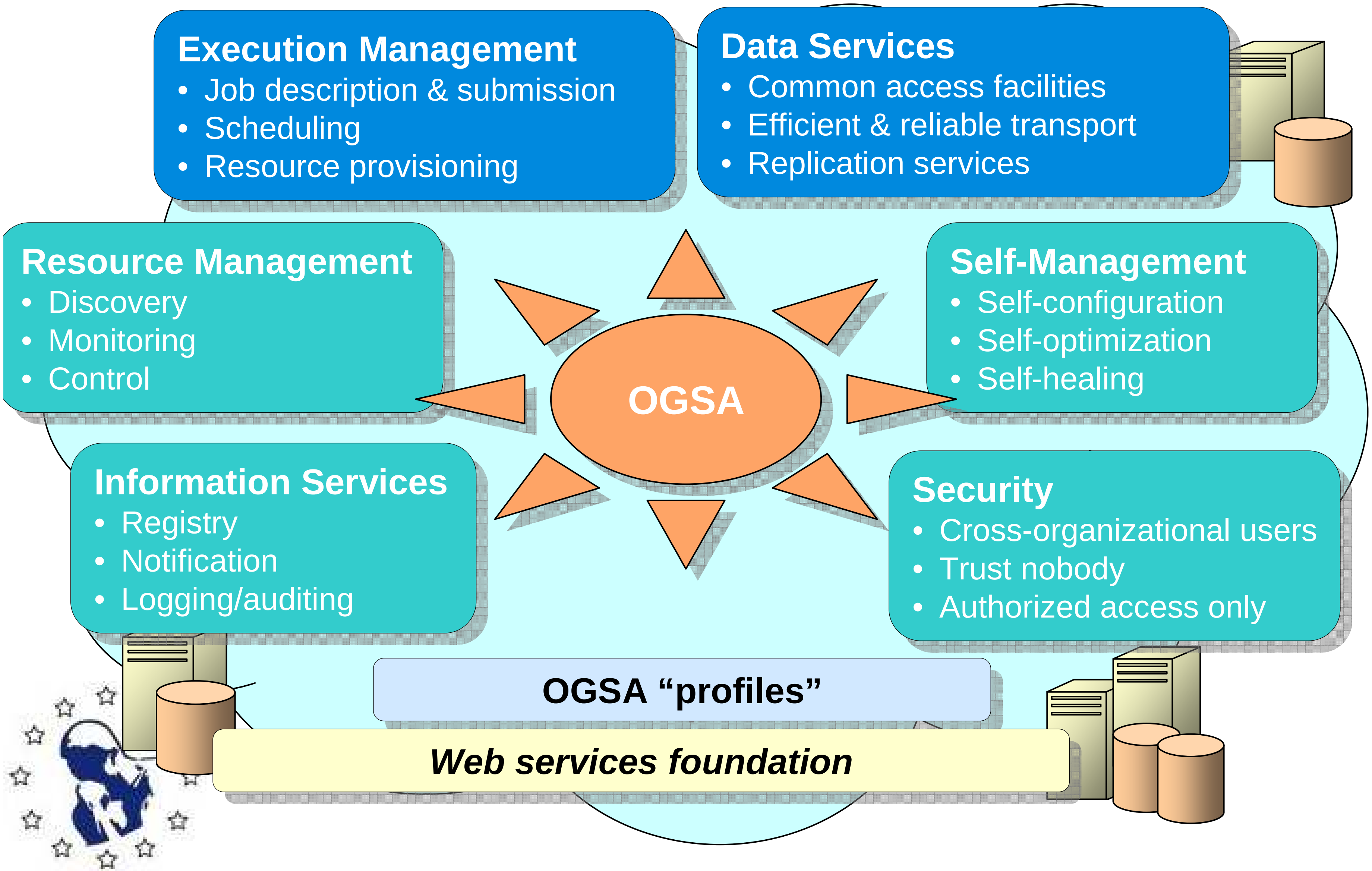
2
- Utilize softwar ommunications protocols developed
  - *Condor, G E, Legion, etc.*
- Need sig tomization to deliver complete solution
- Inte ility is still very difficult!

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

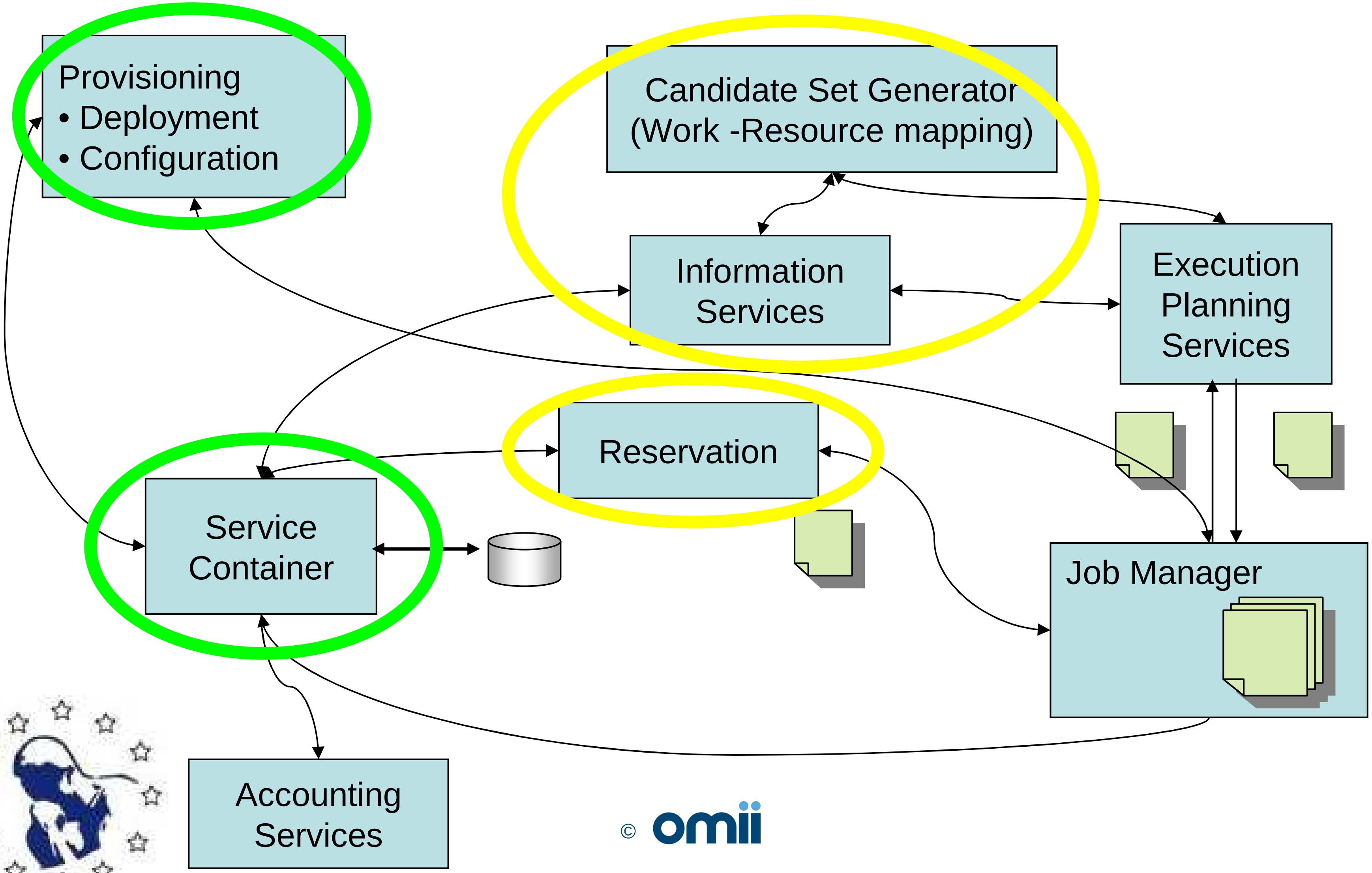
***Standardization** is key for third-generation grids!*

Source: Charlie Catlett

Open Grid Services Architecture

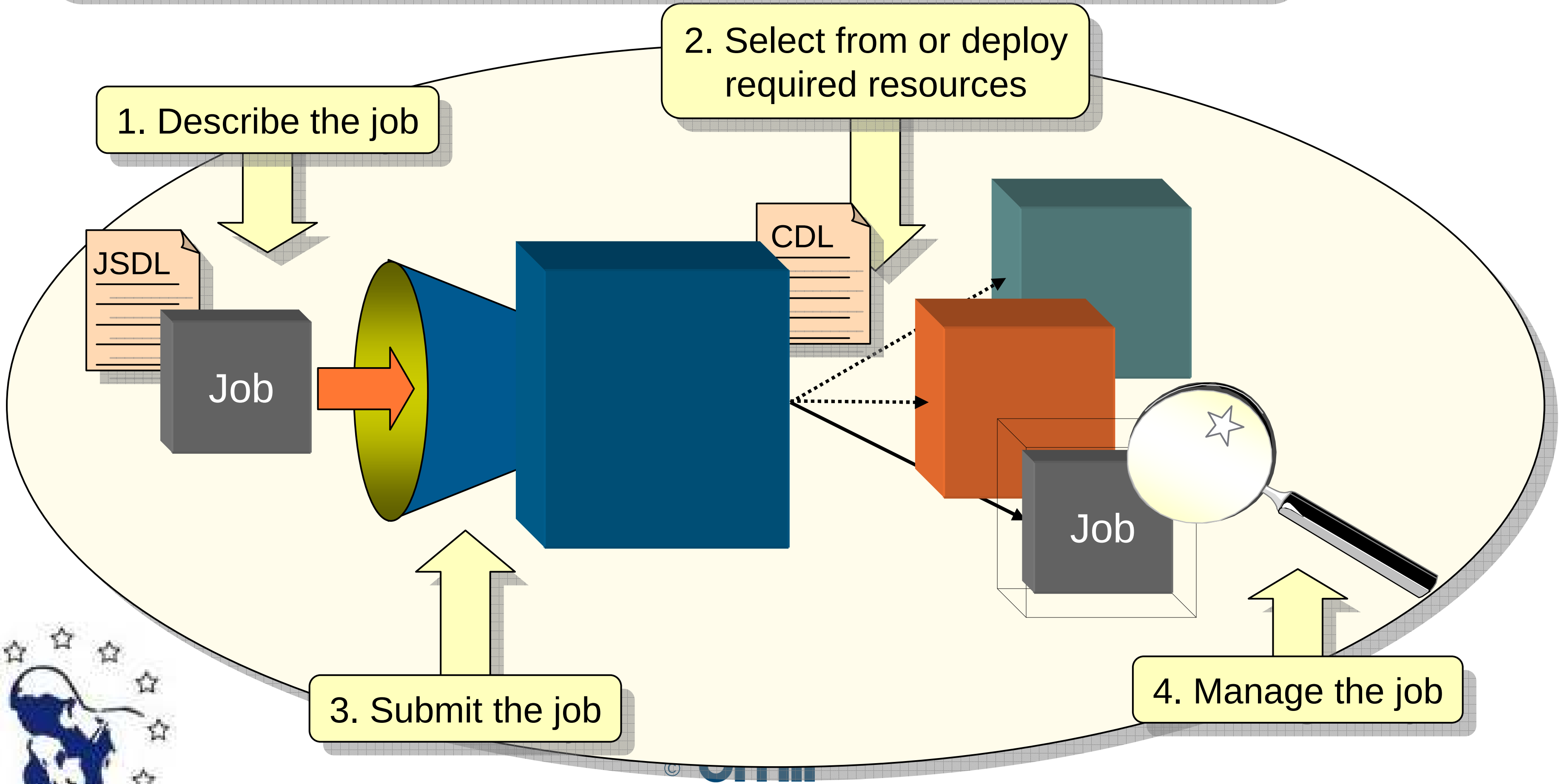


Execution Management Services (EMS) within OGSA



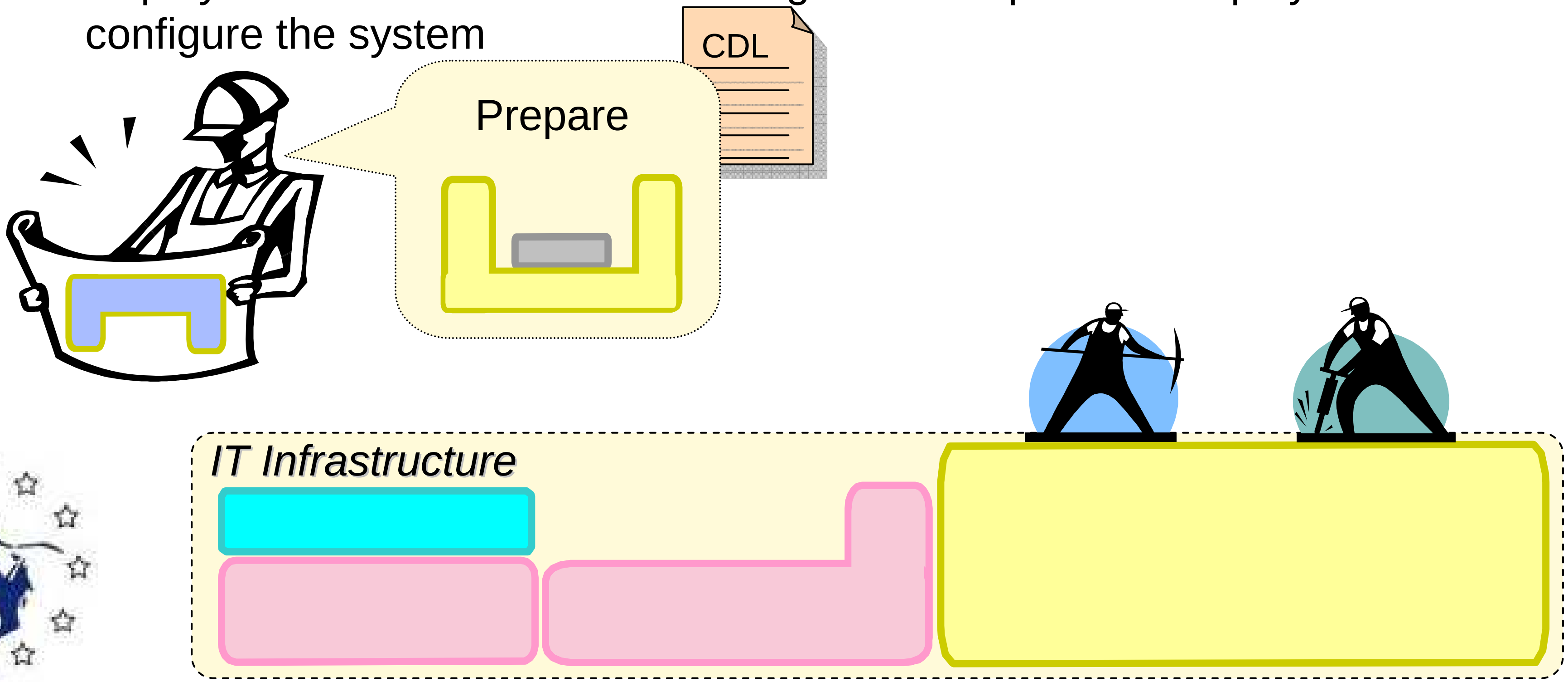
Basic Execution Management

- The basic problem
  - Provision, execute and manage services/resources in a grid
  - Allow for legacy applications



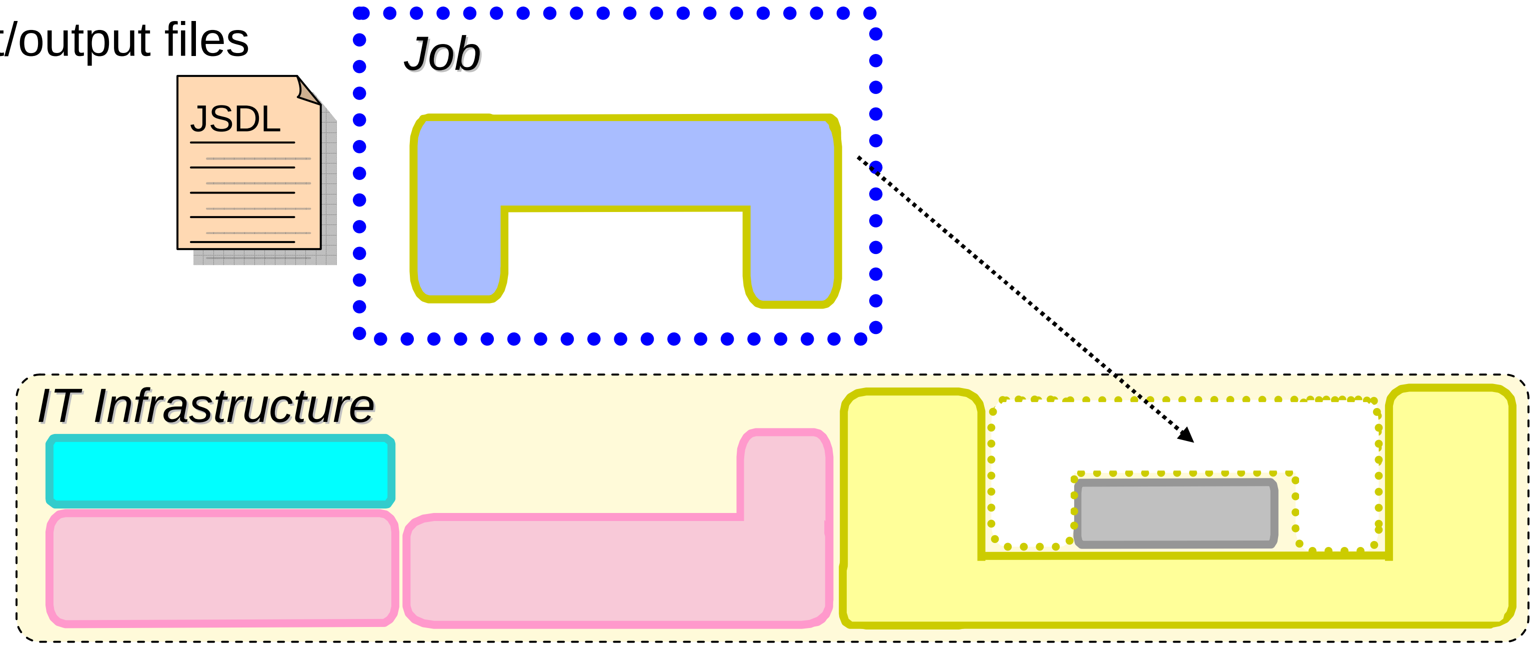
Configuration & Deployment: CDL

- Prepare the infrastructure so that the job can execute
  - Provide a *right-shaped* slot to fit the job
- Main parts:
  - *Configuration Description Language* (CDL) provides declarative definition of system configuration
  - Deployment service carries out configuration requests to deploy and configure the system



Describing a Job: JSDL

- Job Submission Description Language (JSDL)
  - A language for *describing the requirements of jobs for submission*
  - Declarative description
- A JSDL *document* describes the job requirements
  - Job identification information
  - Application (e.g., executable, arguments)
  - Required resources (e.g., CPUs, memory)
  - Input/output files



OGSA
Basic Execution Service

- BES_Factory
  - CreateActivityFromJSDL
- BES_Activity_Management
  - GetActivityStatus
  - RequestActivityStateChanges
  - GetActivityJSDLDocuments
- BES_Container_Management
  - StopAcceptingNewActivities
  - StartAcceptingNewActivities
  - IsAcceptingNewActivities

Summary

- Flexibility & composability in software design
  - You cannot predict how your software will be used
- Grids are (by definition) very diverse
  - But what you do with them (e-Science/e-Industry) is important
- There will NOT be a single grid middleware
  - Open standards & interoperability key

Thank you…

- Acknowledgements:
  - OGSA-WG
  - Listed organisations
- Questions?