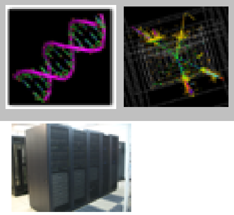
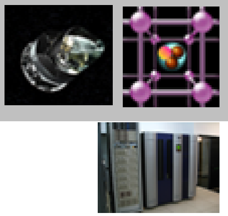




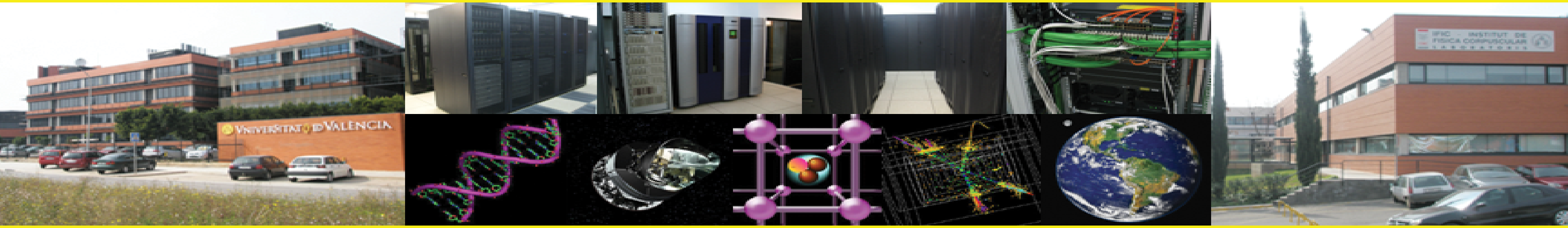
Distributed Computing - Grid Middleware

Álvaro Fernández Casaní

IFIC computing

Training Course I-COOP+2016 project: COOPB20247

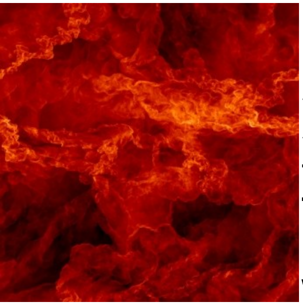
Valencia. July 2017



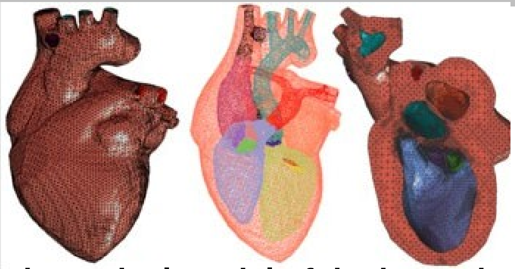
Contents of the talk

- Introduction
- Concepts and Use cases
  - isolated resources vs. shared resources
- Anatomy of the GRID
  - WLCG grid infrastructure
- Middleware layer
  - Security
  - Information System
  - Job management
  - Data management
  - Monitoring and accounting
  - Help channels
- Evolution on the WLCG grid infrastructure

Introduction



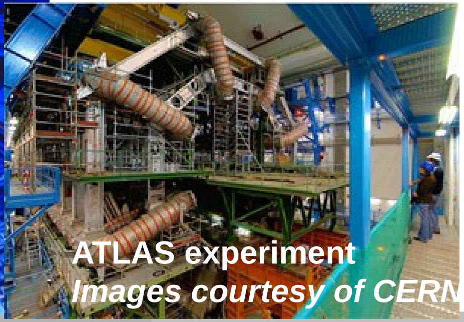
iSGTW story Credit This image, from the San Diego Supercomputer Center at UC San Diego, shows turbulent geophysical flows in the interstellar medium of galaxies. To get this one snapshot of the simulation required 4,096 processors running for two weeks, and resulted in 25 terabytes of data. Brightest regions represent highest density gas, compressed by a complex system of shocks in the turbulent flow.



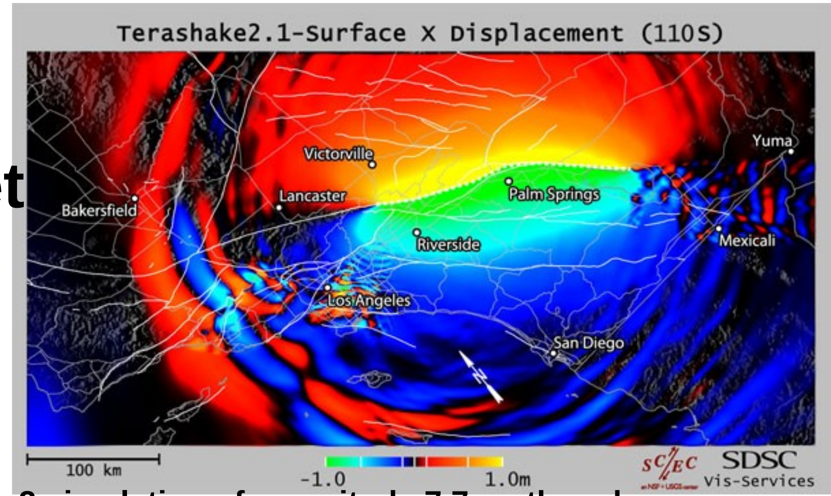
SGTW story | Credit A mathematical model of the heart that simulates blood flow using high-performance parallel computers. Image courtesy of the TACC Visualization Laboratory, the University of Texas at Austin.

- **You as a user (scientist, developer, sysadmin) want to get you job done**

The Time Projection Chamber of ALICE (A Large Ion Collider Experiment).

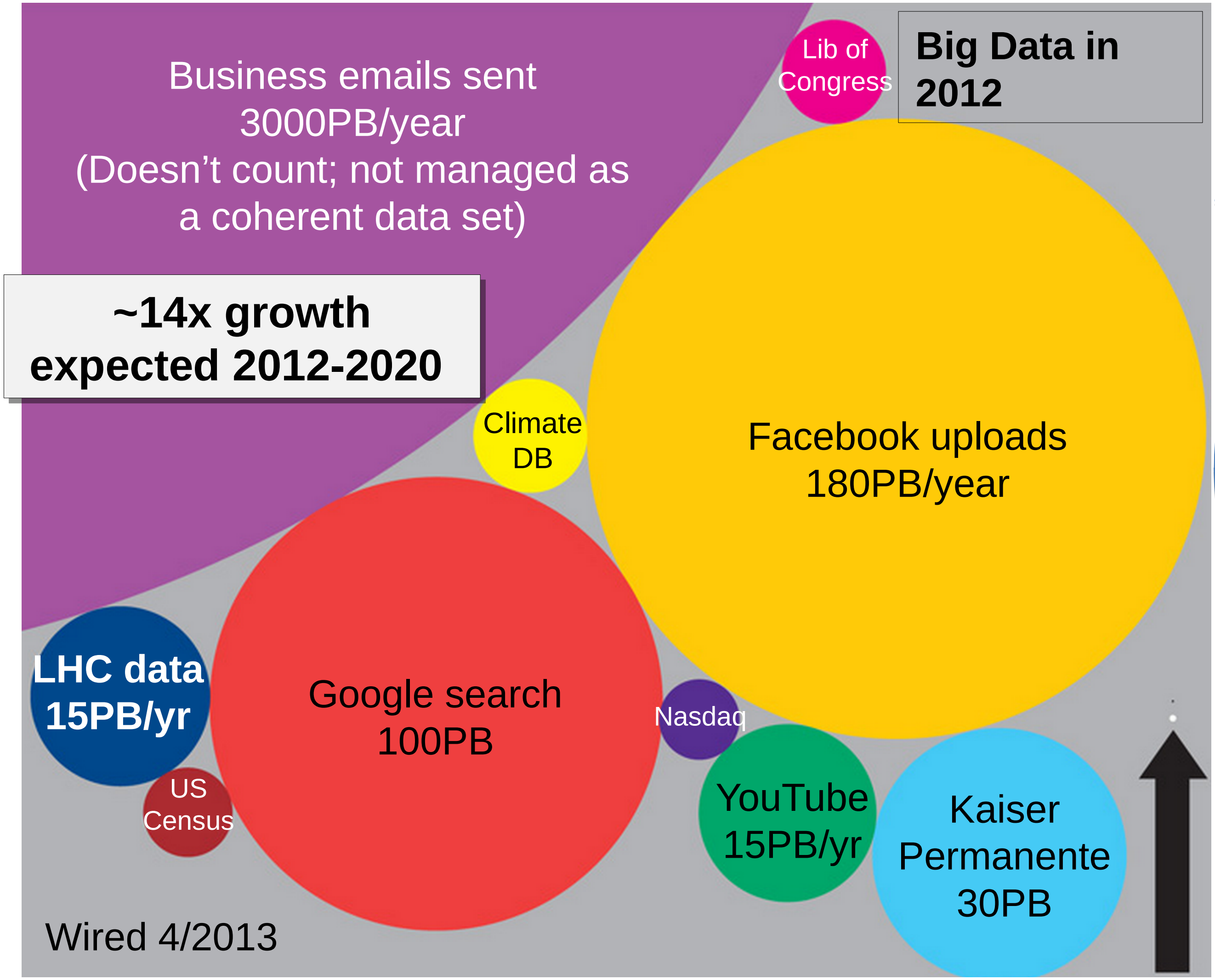


ATLAS experiment
Images courtesy of CERN

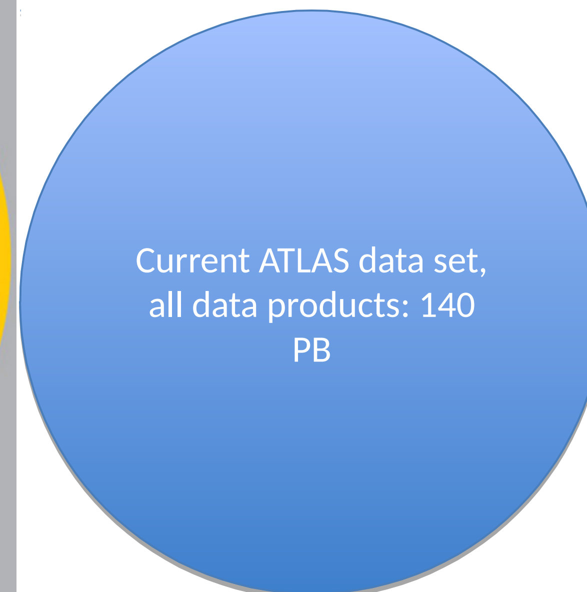


TeraShake 2 simulation of magnitude 7.7 earthquake, created by scientists at the Southern California Earthquake Center and the San Diego Supercomputer Center. *Simulation: SCEC scientists Kim Olsen, Steven Day, SDSU et al; Yifeng Cui et al, SDSC/UCSDVisualization: Amit Chourasia, SDSC/UCSD*

Big Data?



Source: Wired magazine



4

Computational problems

- **Sequential Calculations**: jobs are executed secuentially in 1 cpu
- **Parallel calculations:**many sub-calculations can be worked on "in parallel". This allows you to speed up your computation.
  - Coarse Grain vs. Fine Grain: depending on the number of computations vs. communications
  - Embarasingly parallel: every computation is independent of every other (very coarse grain)
- **High Performance Computing (HPC):** problems that require of high-end resources, tightly-coupled networks with lots of processors and high-speed communication networks.
- **High Throughput Computing (HTC):** values the number of finished computations instead of the computing power. Loosely coupled networks

- **Use of isolated resources**
  - You want to use computational power and storage
  - Don’t want/need to share your resources
  - Don’t want/need to share results
- – This is the traditional cluster’s user case
- – It does not need grid technology, but still can use its methods
- – Disadvantages: can under use resource, depending on computer/data necessities cannot afford costs

Use case: share computation power

- **You want to use/share computation resources**
  - **Origins example: Seti@Home**
  - **Example:BOINC, LHC@HOME, ...**
  - **Not all applications are object of "boincfication"**
- **But resources in general not for free, benefits come from sharing costs, and in general from Distributed Computing:**
  - **High Availability**
  - **Reduce Performance Bottlenecks**
  - **Redundancy (services)**

A solution: Access remote resources when available, and share yours with common members (Virtual Organization)
  - **Need methods to identify users**
  - **Need methods to allow/ban users**
  - **Technology to share computations, best use of resources, etc**
- **Units are computational *jobs***

What about Data?

For example:the LHC produces 40 million collisions per second

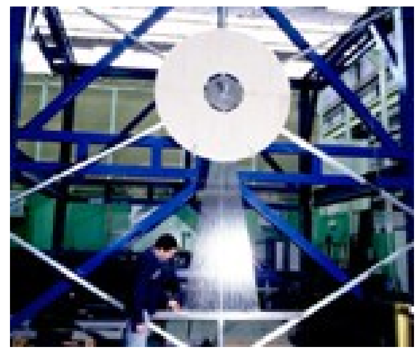
This is filtered down to 100 interesting collisions per second

Each collision produces about one Megabyte of data = recording rate of 0.1 Gigabytes/sec

10^{10} collisions recorded each year

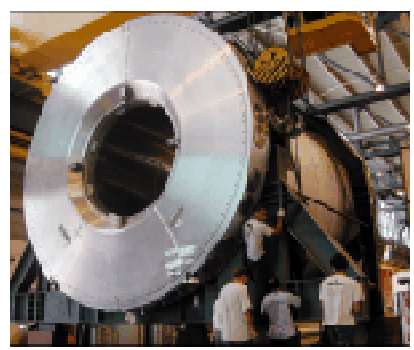
= 10 Petabytes/year of data, plus analysis data

ALICE



CMS

LHCb



ATLAS

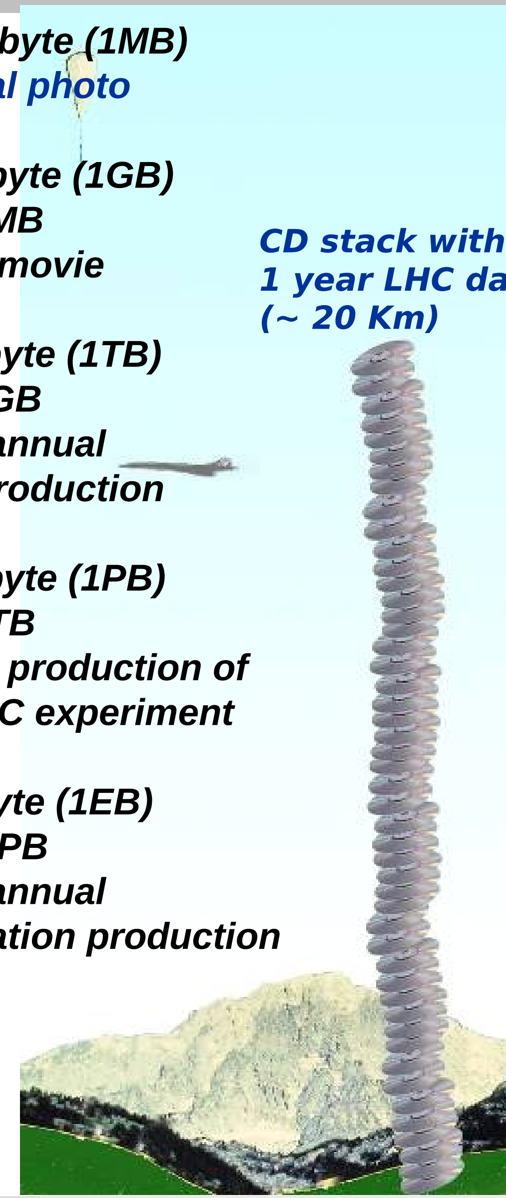
1 Megabyte (1MB)
A digital photo

1 Gigabyte (1GB)
= 1000MB
A DVD movie

1 Terabyte (1TB)
= 1000GB
World annual book production

1 Petabyte (1PB)
= 1000TB
Annual production of one LHC experiment

1 Exabyte (1EB)
= 1000 PB
World annual information production



From gridcafe.org

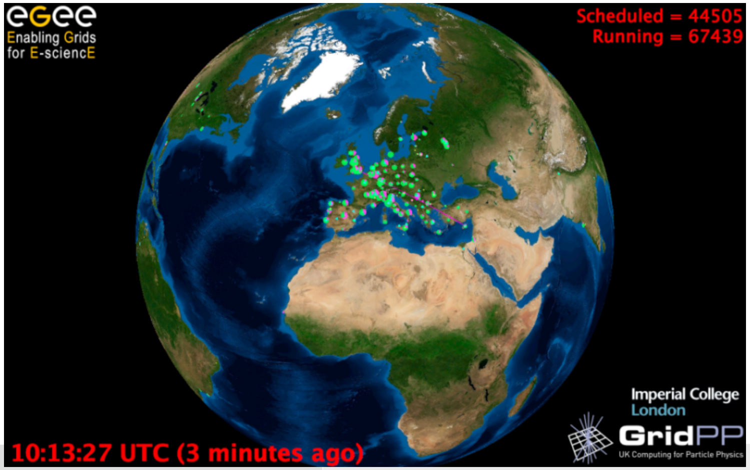
Grid Technology and Virtual Organizations

- **CERN** started to see the high amount of data and computing power they need to process it
- Not feasible to store at a central point
- **Distribute resources among participant centers**
  - Centre puts its computing and storage resources (helps to share costs)
  - Data is distributed among centers
  - Everybody can access remote resources

– **Need technology to access these resources in a coherent manner: Grid Technology**
  - Users belong to Virtual Organizations
  - Secure access and trustworthy relations

Why GRID

- **Great quantity of data with unabordable cost to store it centrally at CERN. Cost**
- **More that 2000 scientists and research centers around the world accessing this data. Performance**
- **Need to have it always available. Availability**
- **A solution is to use distributed technologies-> GRID COMPUTING**



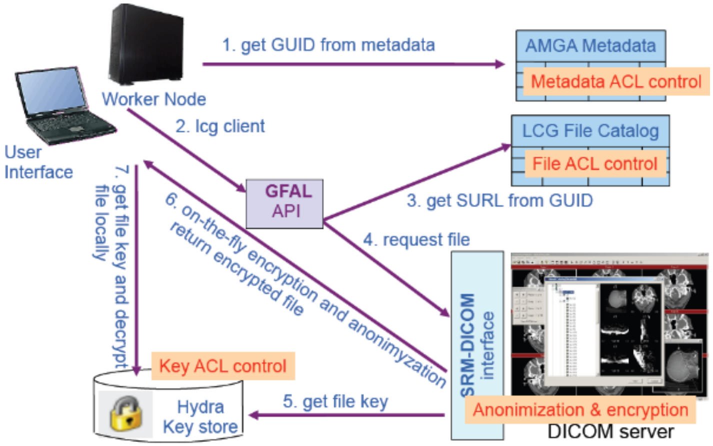
Use case: share data and results

- **Sharing of data** is crucial for some applications
  - You produce data at one site that is consumed at some other place
  - Reduce access bottlenecks (Replication)
  - Data always available (High Availability)
  - Privacy data issues
- Need the appropriate technology:
  - Methods for storing, locating data
  - Methods for Replicate of data
  - Methods for guarantee privacy of data

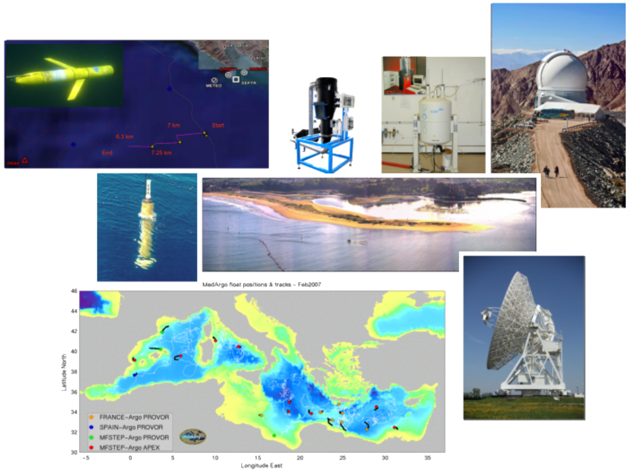
Use case: Medical Data

- **Another area is medical imaging and medical data:**
  - **Privacy of data**
  - **Data cannot leave physically centres (no replication, accept jobs from VO**
- **Example of medical data application**

http://www.gridtalk.org/Documents/ehealth.pdf



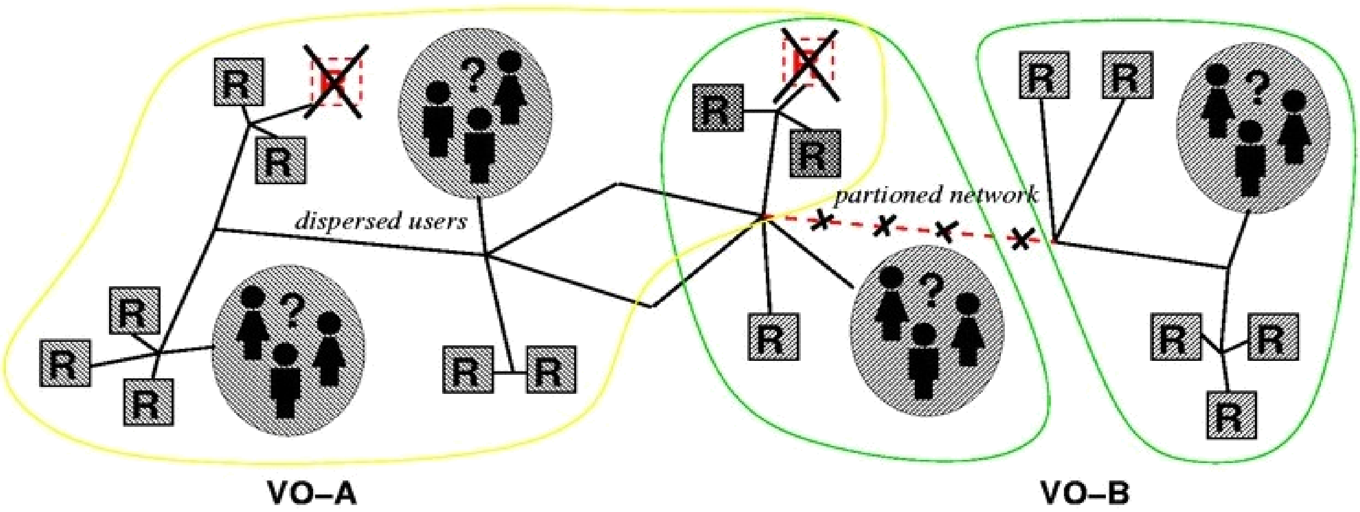
cases: Use of Remote Instrumentation



- Use of expensive remote instrumentation (astronomic instruments, spectrometers, …), that can be exploited by higher community
- Improves scheduling of usage to all users
- Remote Users benefit from expensive or even unique instruments.
- Need strong authentication and security

Virtual Organizations

- A VO is a temporary alliance of stakeholders
  - **Users**
  - **Service providers**



A set of individuals or organisations, not under single hierarchical control, (temporarily) joining forces to solve a particular problem at hand, bringing to the collaboration a subset of their resources, sharing those at their discretion and each under their own conditions.

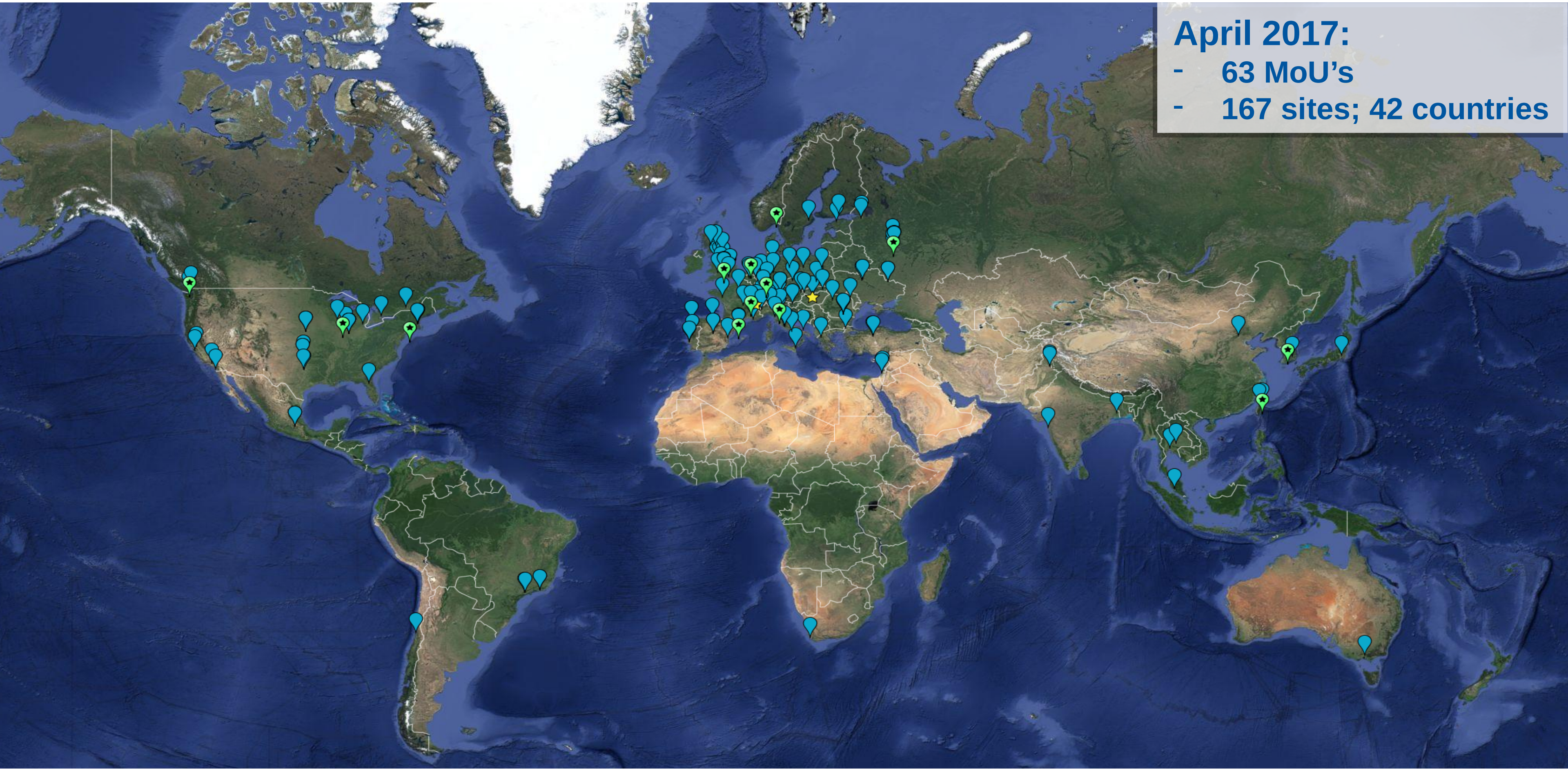
Viewgraph: Foster, Kesselman, Tuecke, the Globus Alliance

Virtual Organizations needs

- **In Grid resources are maintained but their owners, not centralized.**

- **But Virtual Organization need control its members**
  - **authorize a group of users / ban (Authorization methods)**
  - **Authorization Methods**
- **Control of the availability of the resources**
  - **Monitoring**
- **Control who is accessing the resources, what kind of jobs are running**
  - **Accounting**

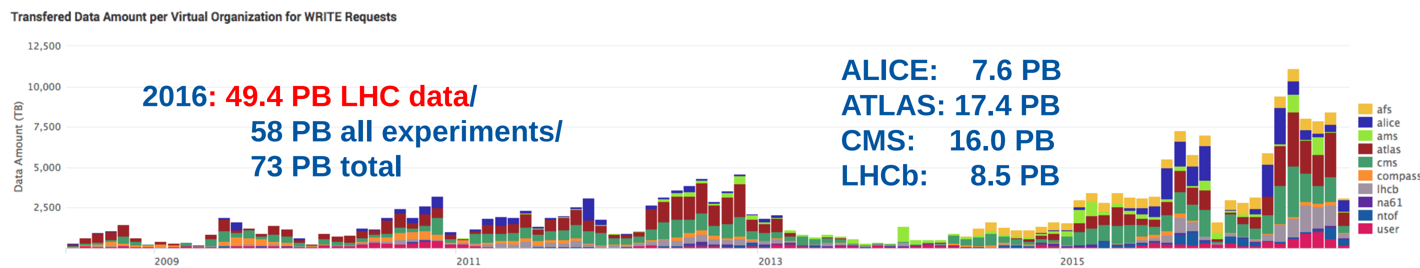
Use case physics : LHC experiments

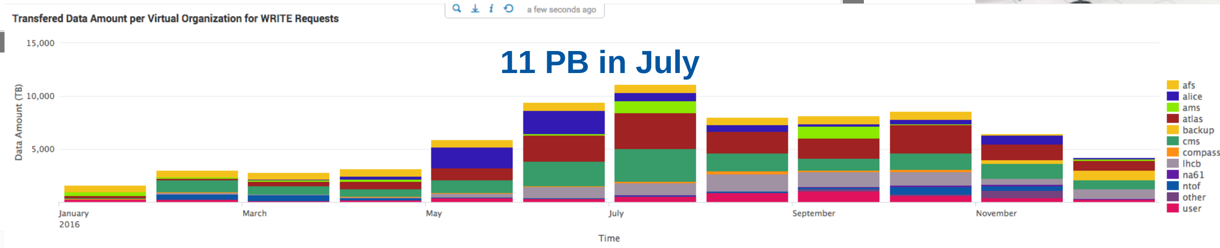


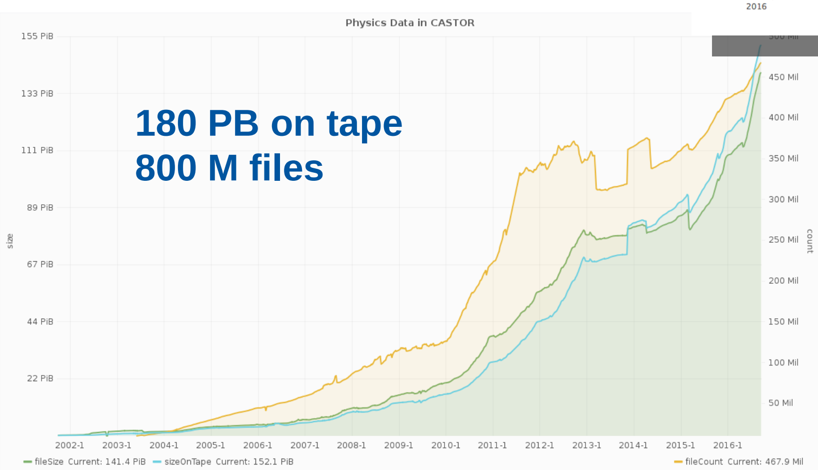
April 2017:
- **63 MoU's**
- **167 sites; 42 countries**

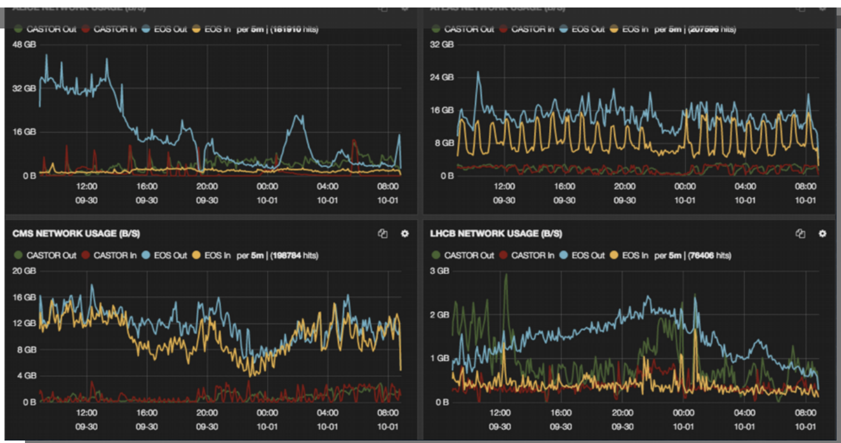
Grid large infrastructure: WLCG Collaboration

Data in 2016 - updated



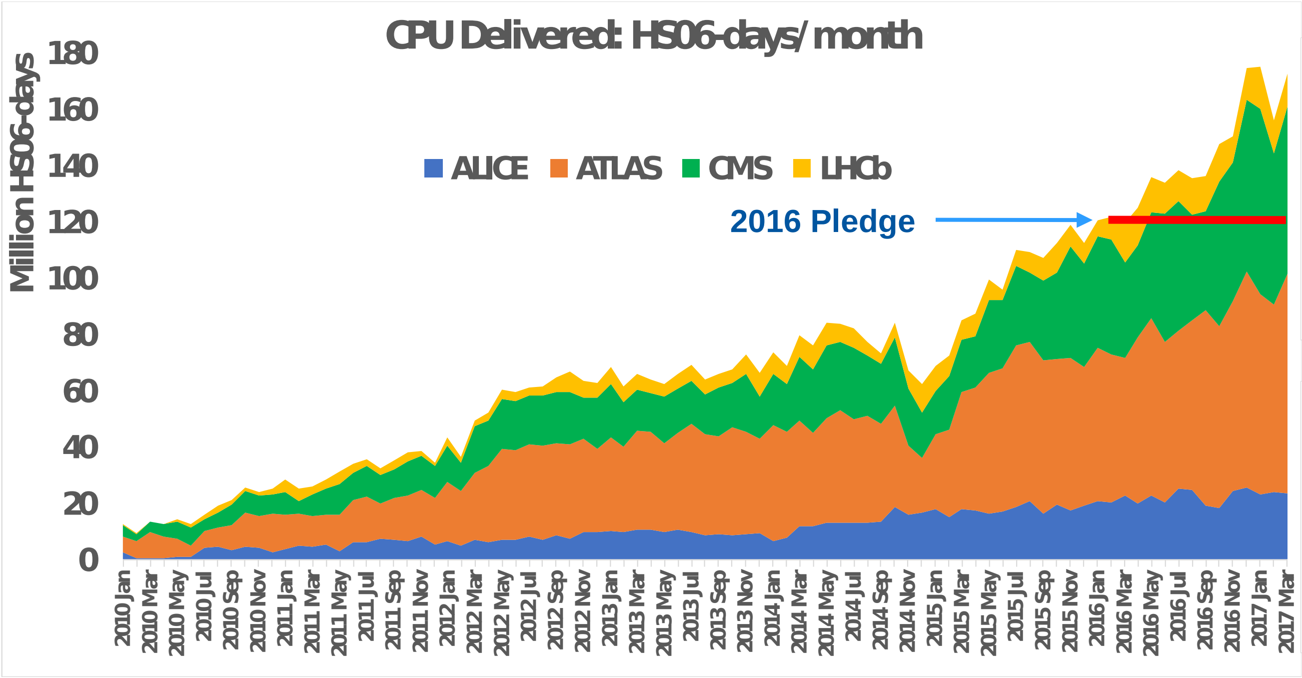


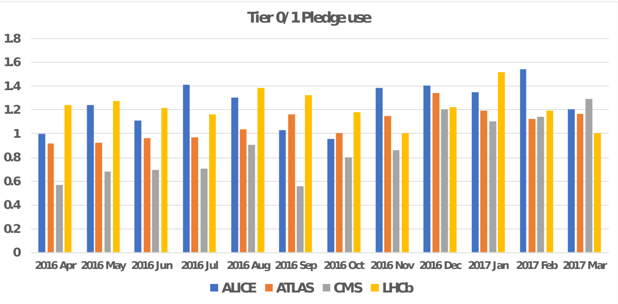


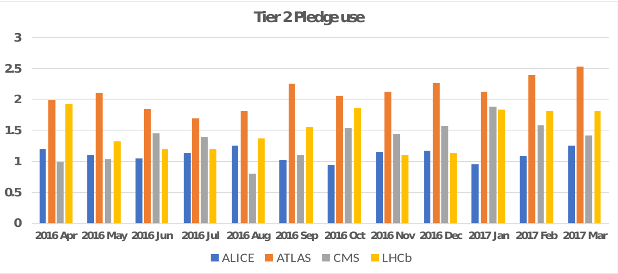


WLCG Workshop, Manchester

WLCG- CPU Delivered

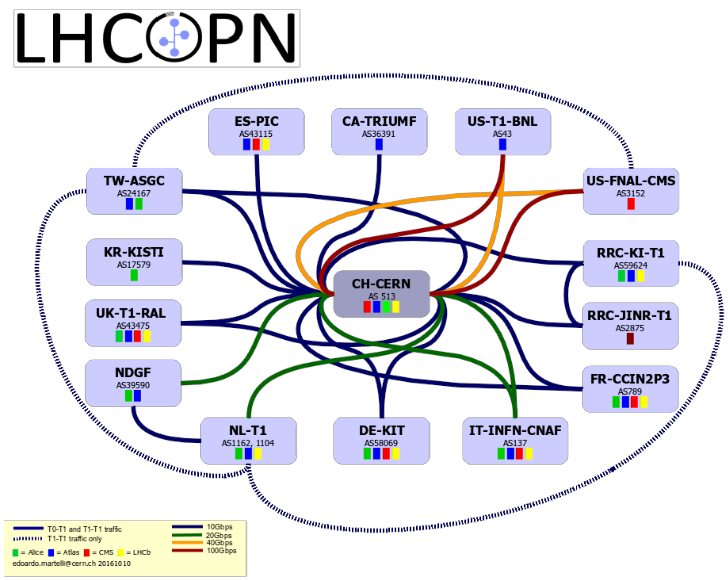


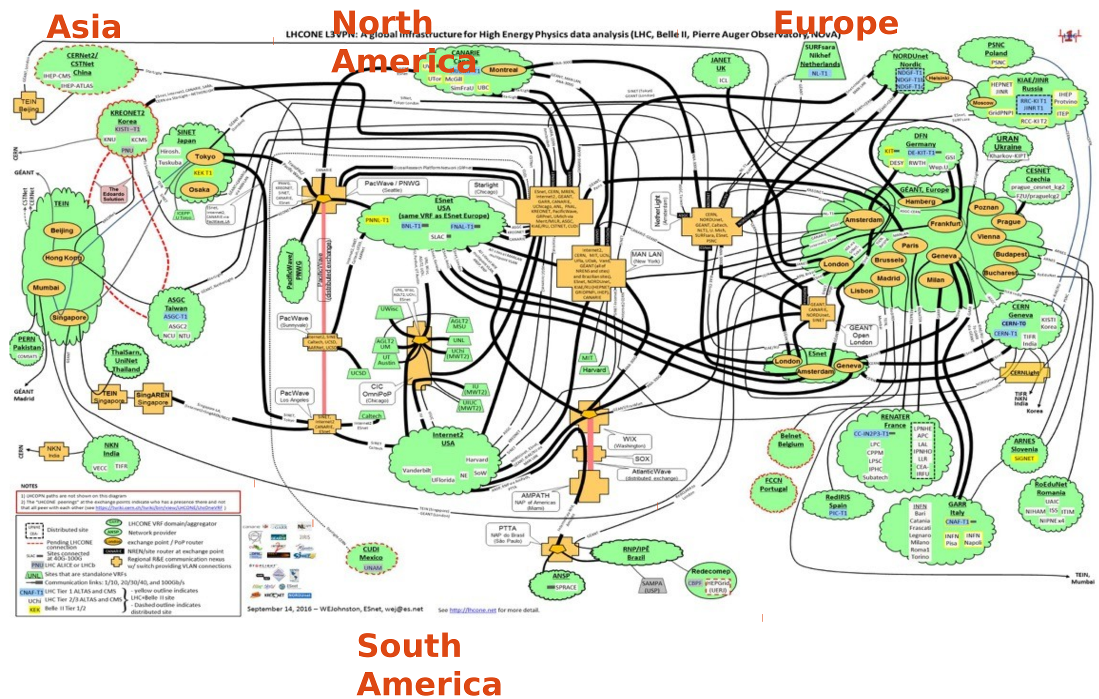


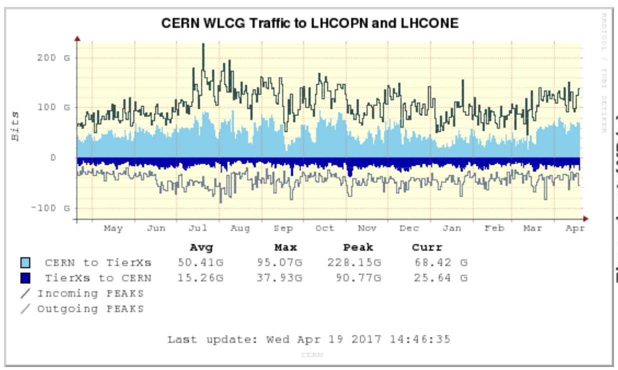


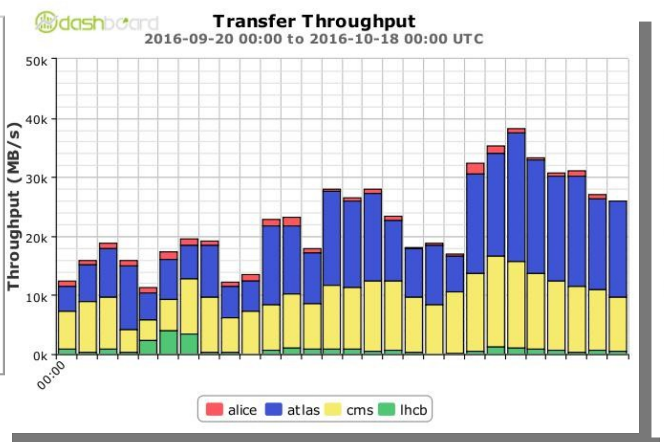
New peak: ~180 M HS06-days/month
~ 600 k cores continuous

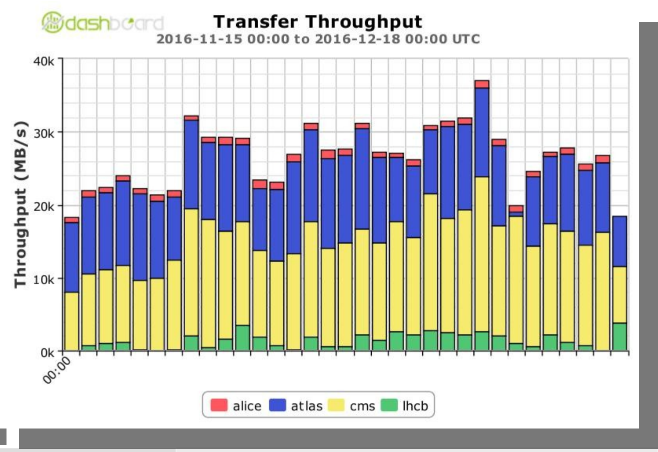
Data transfers











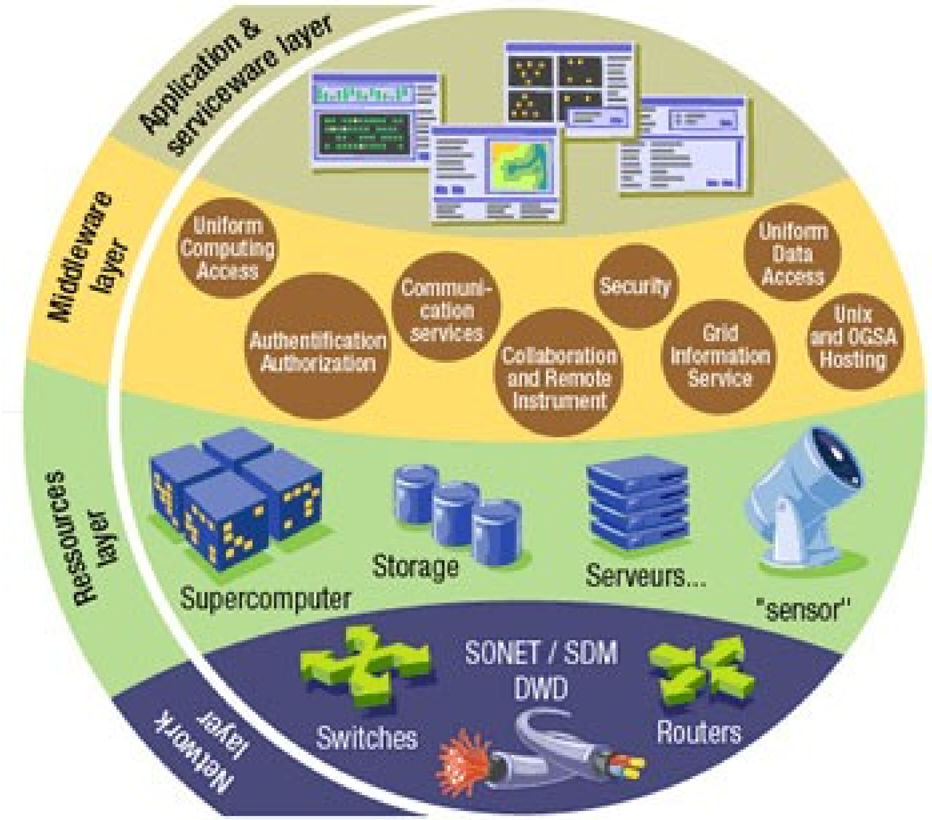
WLCG Results

- Run 2 in 2016 delivered 50 PB of new data, following exceptional performance of the LHC
  - Continued to set new performance records in all areas
- WLCG infrastructure continued to be even more active in the EYETS
- 2017/18 look to be challenging in terms of resource availability, esp if LHC meets expected luminosities, availability
- Activity (& engagement) is ramping up to look at evolution of the computing models for the future

Future: HL-LHC Computing TDR

- Agreed with LHCC to produce TDR for HL-LHC computing in 2020
- In 2017 we will provide a document describing the roadmap to the TDR (strategy document)
  - Using the CWP as input
  - Describing potential new computing models
  - Defining prototyping and R&D work that will be needed
- The TDR will not be the end – technology evolution in 6-7 years will be significant, cannot afford not to follow it
- NB. Very different situation from the original TDR –
  - we have a working and well-understood system that must continue to operate and evolve into the HL-LHC computing programme

Anatomy of the grid



- **Application Layer: applications and interfaces**
- **Middleware Layer: sits between App and OS to provide basic access services**
- **Resource Layer: computing, storage, instruments**
- **Network Layer**

- Provides a set of common services to access remote resources in a coherent manner.
- *Globus* is the most common middleware (http://globus.org/) used in WLCG
- Also Nordugrid ARC: http://www.nordugrid.org/documents/whitepaper.pdf
- Higher middleare on top of these:
  - EGI : https://www.egi.eu
  - Built from previous Datagrid/Crossgrid/EGI/EMI initiatives
- WLCG stablishes minumum requirements:

https://twiki.cern.ch/twiki/bin/view/LCG/WLCGBaselineVersions

Globus

- NSF has announced end of support for open source Globus toolkit, from end 2017
  - I have been in touch with NSF to ask about support for LHC – they recognize the problem
    - No feedback yet
  - What will OSG and EGI do?
- Fall-back – WLCG takes relevant packages and maintains them
  - gsi, gridftp, myproxy
  - And perhaps eventually replaces them

WLCG Workshop, Manchester

19 June 2017

Middleware : Important areas

Middleware covers important areas needed for working with these distributed grid infrastructures:

Security Services:

- **Authentication, Authorization**

Information Service

Job Management

Data Management

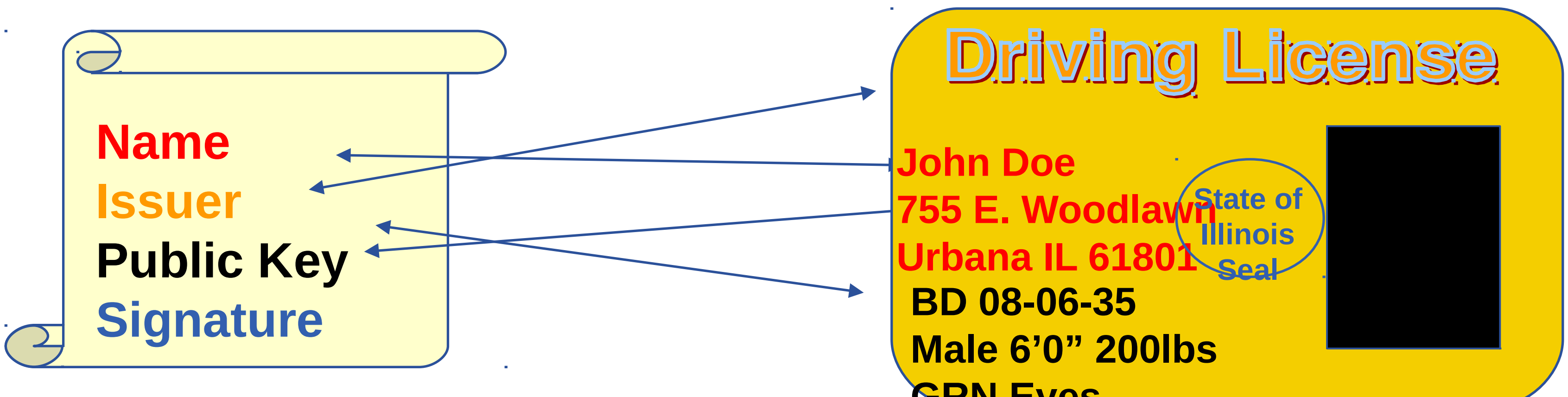
Security Services

- You want to be sure that that people access your resources the way you want. Possible problems:
  - Unathourized access: by users not known, using your resources
  - Attacks to other sites: Large distributed farms of machines, perfect for launching a Distributed Denial of Service attack.
  - Access and distribution of sensitive information: access to sensitive data, or store

- Authentication
  - Are you who you claim to be?
- Authorization
  - Do you have access to the resource you are connecting to?

Public Key Infrastructure (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. SO YOU NEED A CERTIFICATE TO IDENTIFY YOURSELF



slide based on presentation given by Carl Kesselman at GGF Summer School 2004

Security: Basic Concepts

Authentication based on X.509 PKI infrastructure

- Certification Authorities (CA) issue certificates identifying individuals (much like a passport or identity card)
  - Commonly used in web browsers to authenticate to sites
- Trust between CAs and sites is established (offline)
- In order to reduce vulnerability, on the Grid user identification is done by using (short lived) proxies of their certificates

- **Proxies can**

Be stored in an external proxy store (myProxy)

Be renewed (in case they are about to expire)

Be delegated to a service such that it can act on the user's behalf

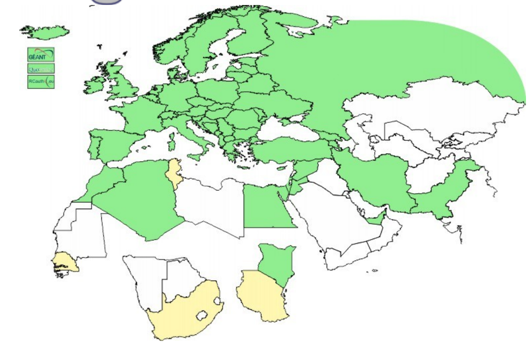
Include additional attributes (like VO information via the VO Membership Service VOMS)

Certification Authorities

Common trust domain for all of Europe: the EUGridPMA

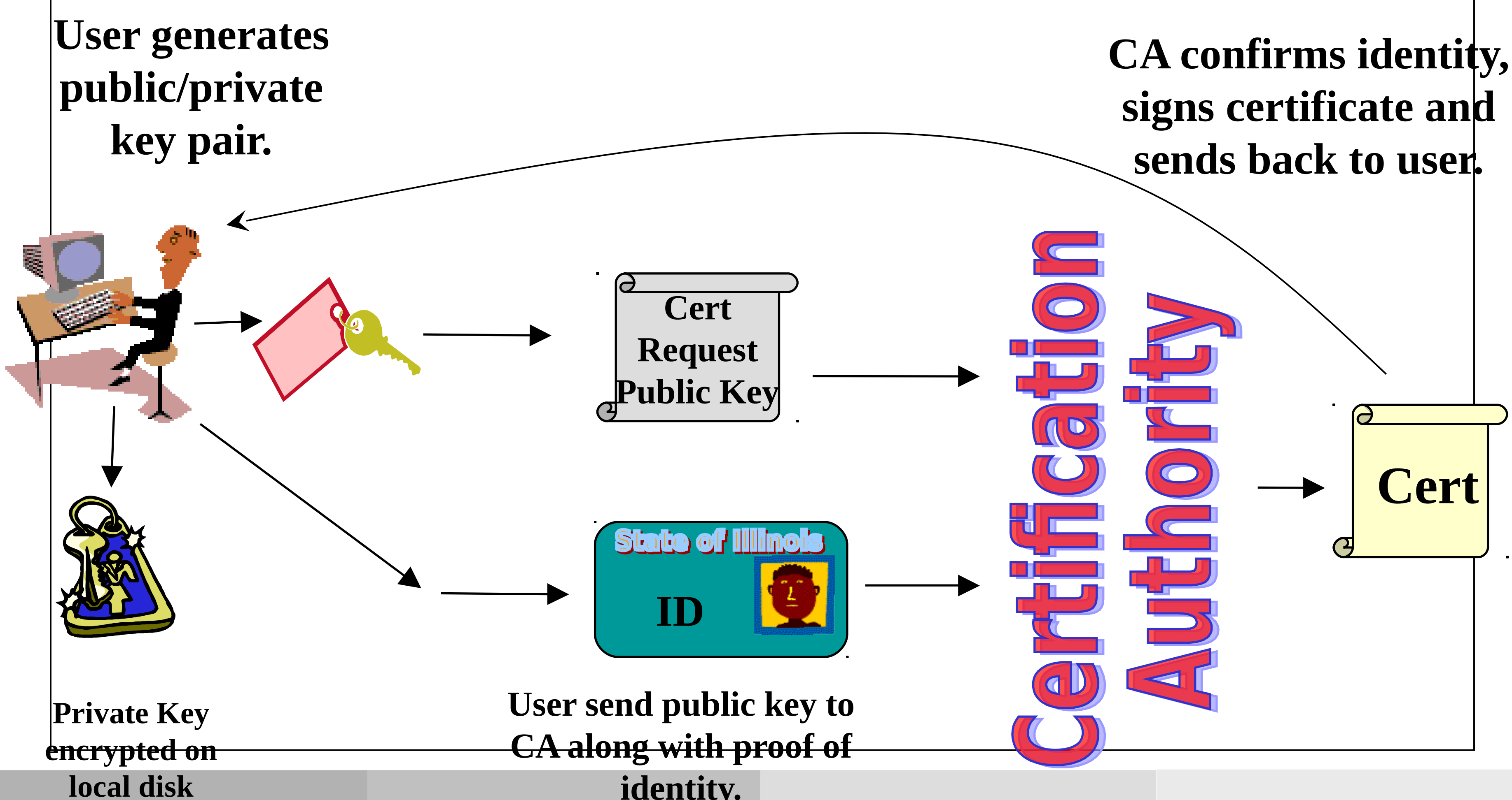
- **23 national certification authorities**
- **catch-all CAs for EGEE, LCG, etc**
- **all comply to the same minimum standards**
  - **in-person checking with a photo-ID**
  - **secure signing machine**
  - **certificates valid for 1 year**
  - **…**
- **your Grid certificate works across all of Europe**

- ***other CAs exist: for students, demonstrations, tutorials ☺***

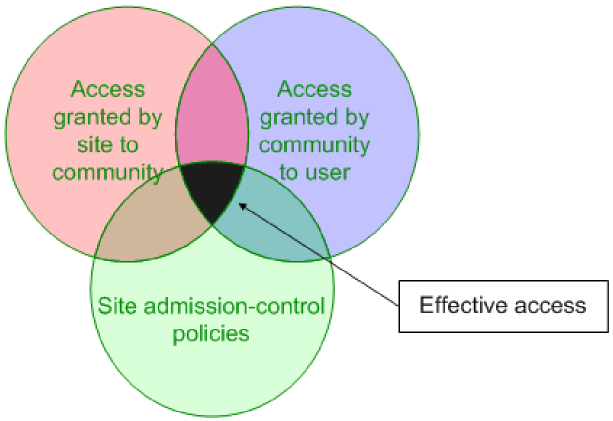


Name: CA
Issuer: CA
CA's Public Key
CA's Signature

Certificate Request



- **Based on *Virtual Organisations* (VO)**
- **you join both a VO and (implicitly) an Infrastructure:**
  - **agree to the Acceptable Use Policy**
  - **request VO membership**
  - **wait for the VO administrator to approve**
  - **resource providers will then *automatically give you access!***
- ***You can join several Vos with a user certificate***

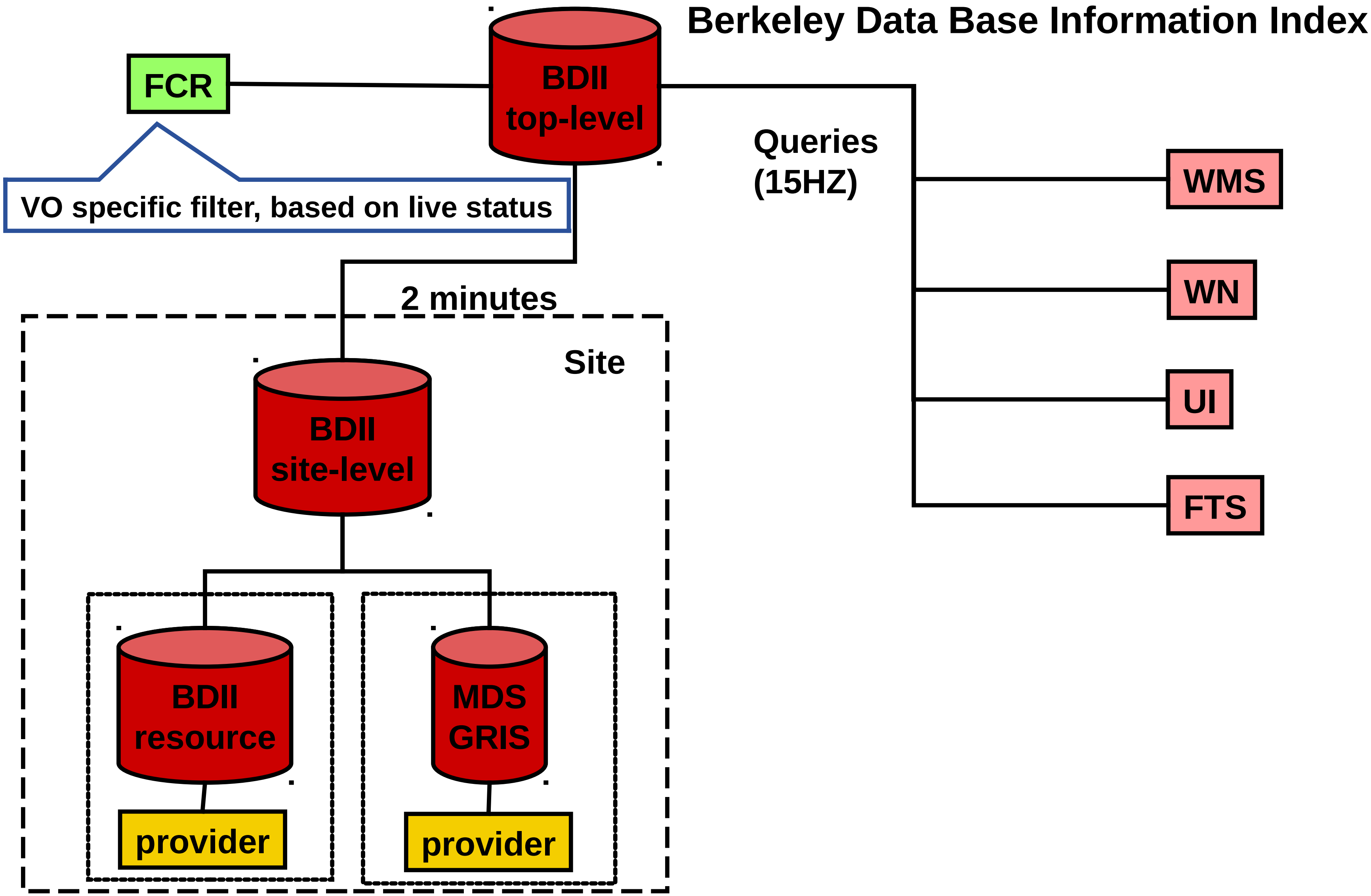


IFIC VOMS

https://swevo.ific.uv.es:8443/vomses

Information System

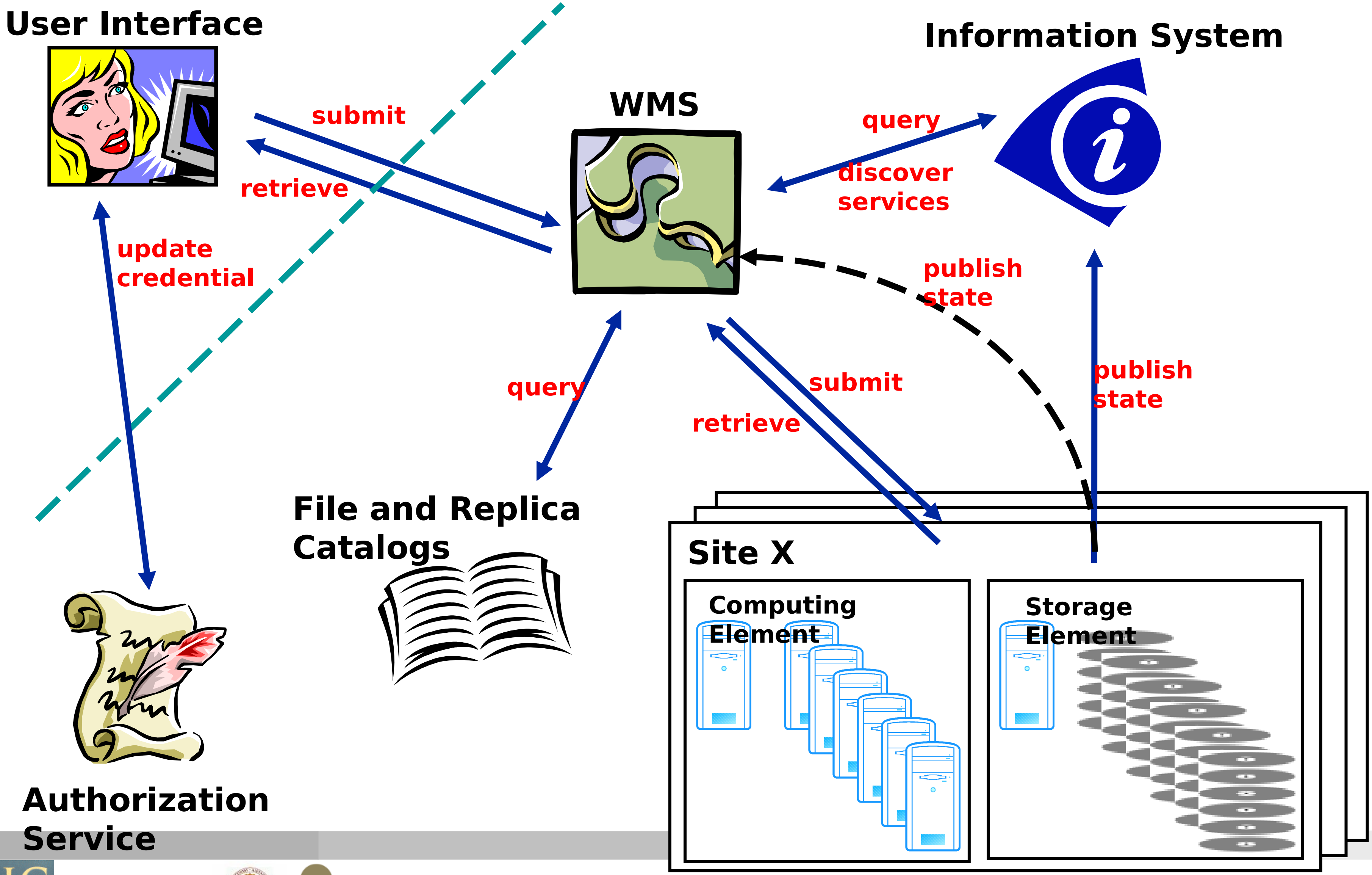
- A way to locate resources and to know its state
- User use it to known: where to run jobs? Where to store data? Complex queries: site that can run 72h jobs with installed Matlab v.xx, that can store 1 TB (and rank it by Estimated time It will finish)
- It has a hierarchic architecture:
  - Each service publish its state
  - Each site groups and publish its services
  - At the top, several sites are published
- Resources publish information and its collected by higher instances (pulling)
- There is a schema known to publish attributes (Glue Schema)
- Other parts of the Middleware use it
  - Job Management Services to locate best resources to run
  - Data Management Services to characterize storage and locate directories
  - Monitoring : to locate working services


Berkeley Data Base Information Index
FCR
BDII
top-level
VO specific filter, based on live status
Queries
(15HZ)
WMS
WN
UI
FTS
2 minutes
Site
BDII
site-level
BDII
resource
MDS
GRIS
provider
provider

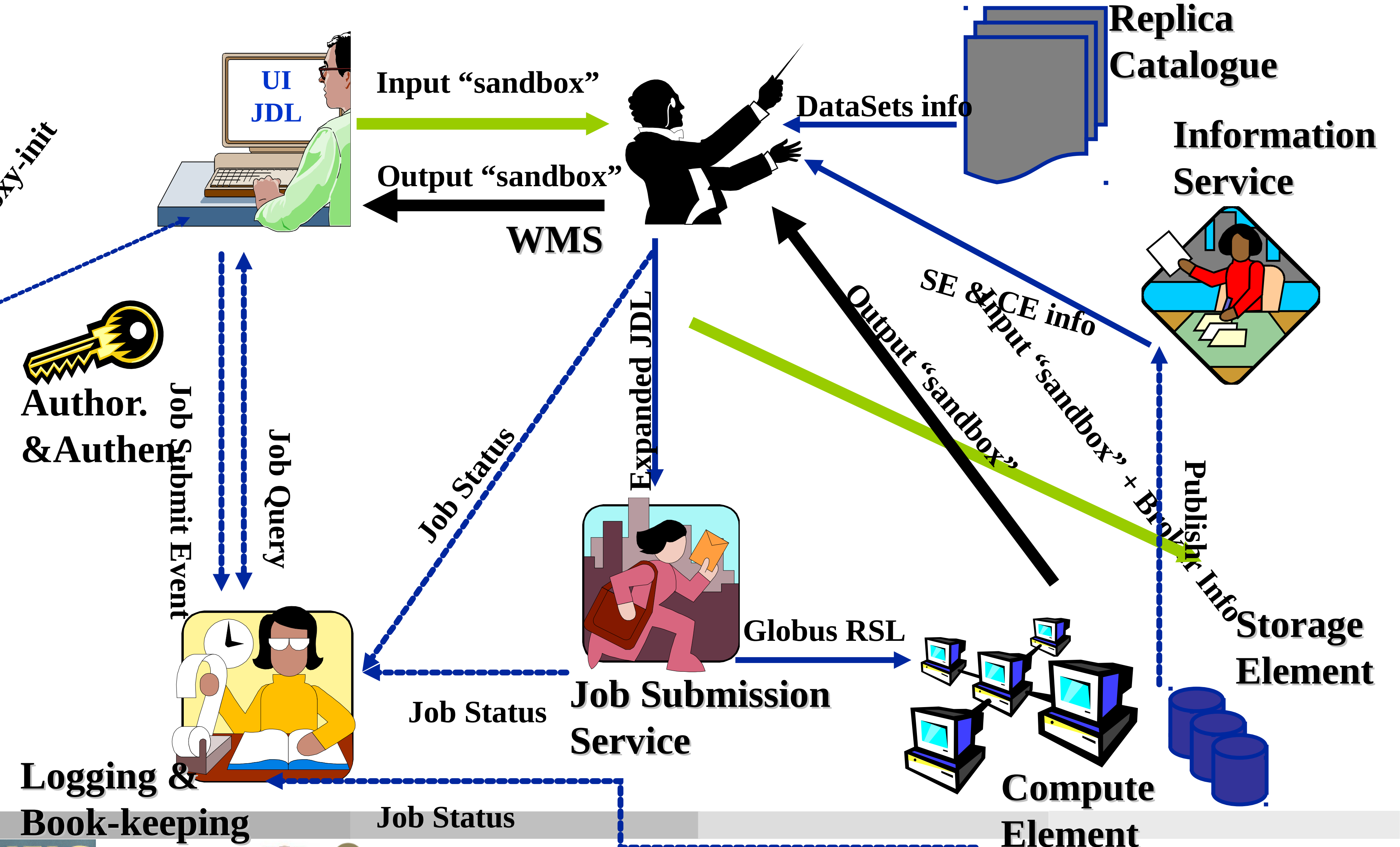
Job Management Services

- Grid can be quite complex, a way to orchestrate and complete jobs
- So we need a scheduler to
  - Accept jobs in the name of a user
  - Select and send them to the best resources
  - Maintain state of (hundreds/thousands) jobs, resubmit if necesarry,
  - Maintain output, until retrieved by the user
- Workload Management System (WMS)
- A Language to define your job requirements (JDL)

Grid Topology and Services



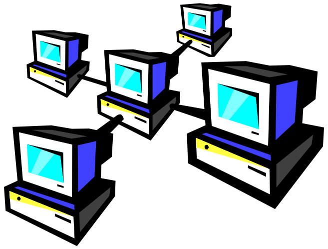
A typical job workflow



User Interface

- Entry point to the grid:
  - Usually It is a special machine with all the clients necessary
  - Every site/organization has one
- Access to you certificate to create proxies and delegate to the services
- You can also compile your programs there and submit from there

Computing Element (CE)



- Represents a computing node at a remote site
  - A batch sytem that schedules jobs
  - A set of computers (Worker Nodes) behind, able to run jobs
- A site can Have Several Ces grouping omogeneous Worker Nodes
- Jobs wait in the batch system at the CE, until can be executed
  - Wall time: Total time that is in queue and executing
  - CPU time: time that your job consumes
- Jobs will be executed finally in a WN, and when finished return the output (to the WMS and the User)

CE and batch system trends

- CE types
  - CREAM by far the most numerous
  - ARC and HTCondor on the rise

- Batch systems
  - PBS/Torque by far the most numerous
  - HTCondor and SLURM on the rise

- PIC working on APEL parser for HTCondor CE + batch system

Job Requirements

- LHC experiments run a mix of single and multi core jobs
- Generally the jobs require largely less than 2GB of memory but there are special cases (e.g. Heavy Ions and Upgrade samples) requiring more
- Flexibility to provision what you need at low/no extra cost will be increasingly important
- A challenge for site administrators and for the batch systems
- Do we have the right (modern) tools? http://cern.ch/go/jx9S

Batch system	Instances
HTCONDOR	86
LOADLEVELER	2
LSF	52
OGS/GE	2
PBS	88
PBSPRO	1
SGE	40
SLURM	57
TORQUE	169

Configuration methods

- YAIM – still there, but minimally maintained
- Puppet – on the rise
- Ansible – ditto?
- Quattor / Aquilon
- A bunch of others used at some sites
  - Chef
  - SaltStack
  - CFEngine
  - …

WorkerNode: CentOS/EL7

- ALICE, ATLAS and LHCb can run on it today
- CMS are making good progress
- For now CMS can run on CentOS/EL7 <u>if</u> the site provides *Singularity* <u>at the same time</u>
  - To allow the jobs to have an SL6 environment
- EMI WN and UI meta packages foreseen to be released in the May update of UMD-4
  - Preliminary versions available in the community preview repository

Experiment trends

- ATLAS, CMS: multi-core vs. single-core jobs
  - "*Tetris*" problem: up to 7 unused single-core slots to create an 8-core slot
- LHCb: job "masonry"
  - Use a job slot's remaining time as efficiently as possible
  - Would benefit a lot from MJF deployment
    - See next page
- ATLAS Event Service jobs
  - Save often, lose little work when killed by the batch system

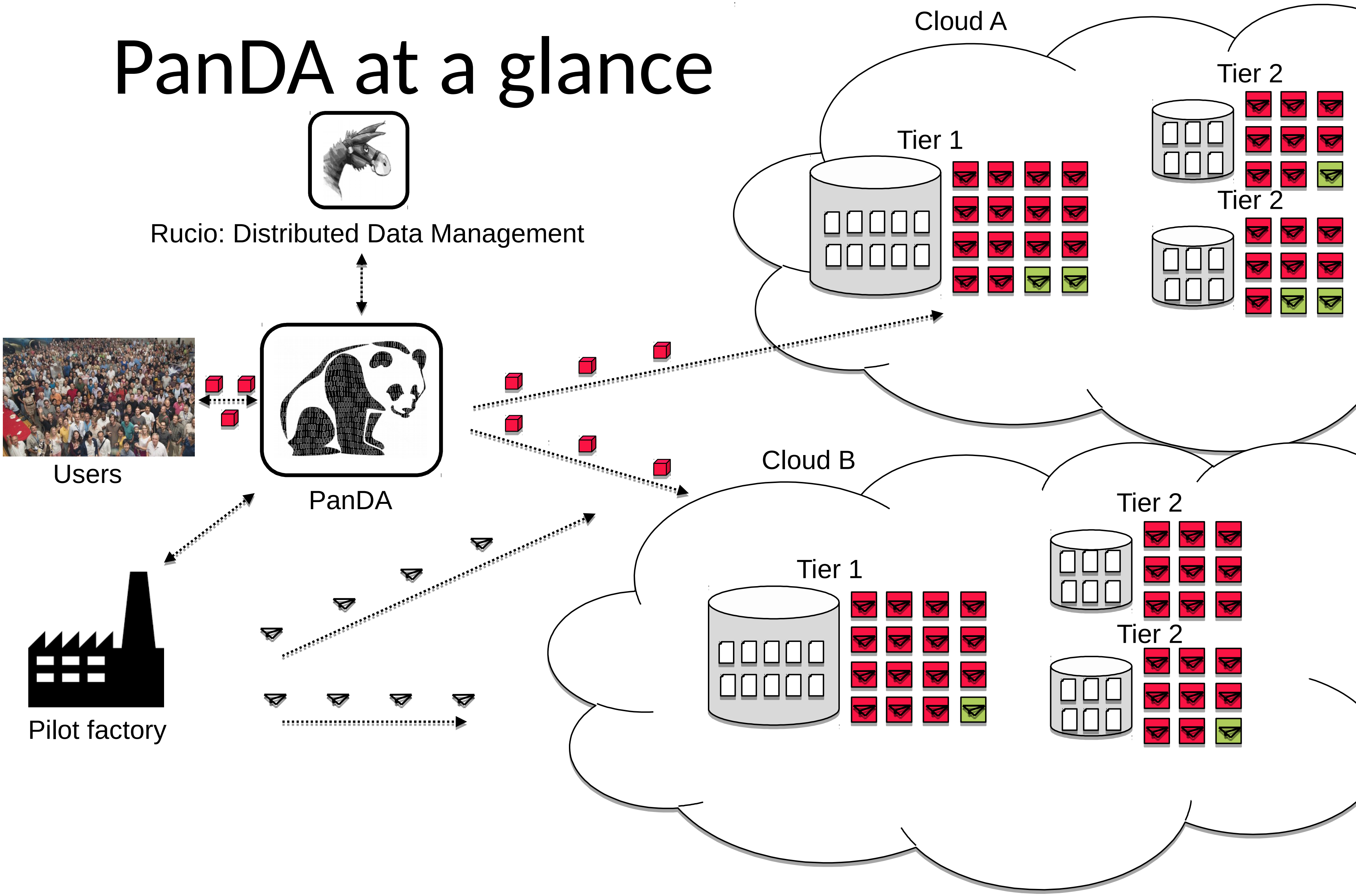
Workload Management Systems

Submit jobs to the grid from a user (or a role 'production' from the Virtual Organization)

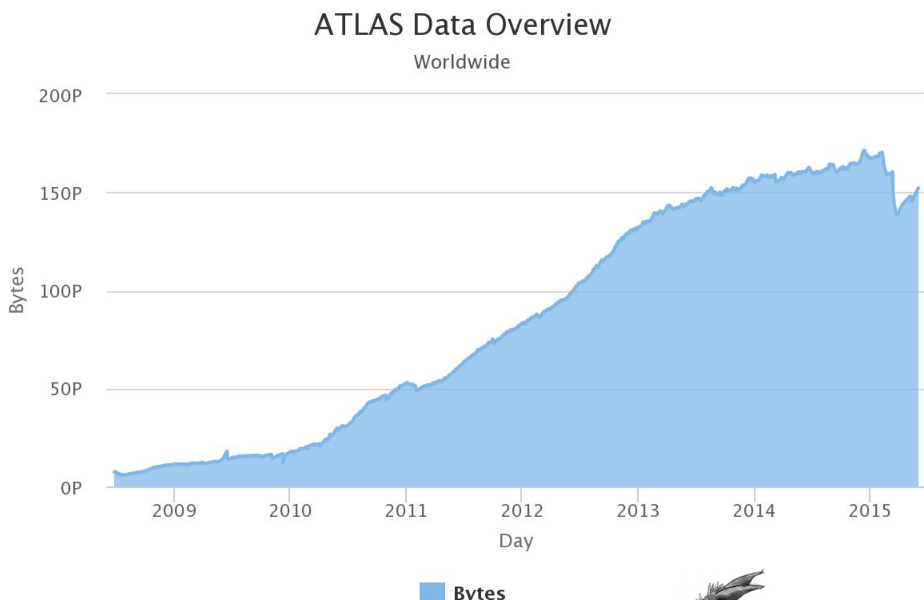
Examples:

- ATLAS Prodsys / Panda:
  - Tutorial: https://indico.cern.ch/event/626719/
  - https://twiki.cern.ch/twiki/bin/viewauth/AtlasComputing/ProdSys
- LHCB Dirac
- Glite WMS (to be decommisioned)

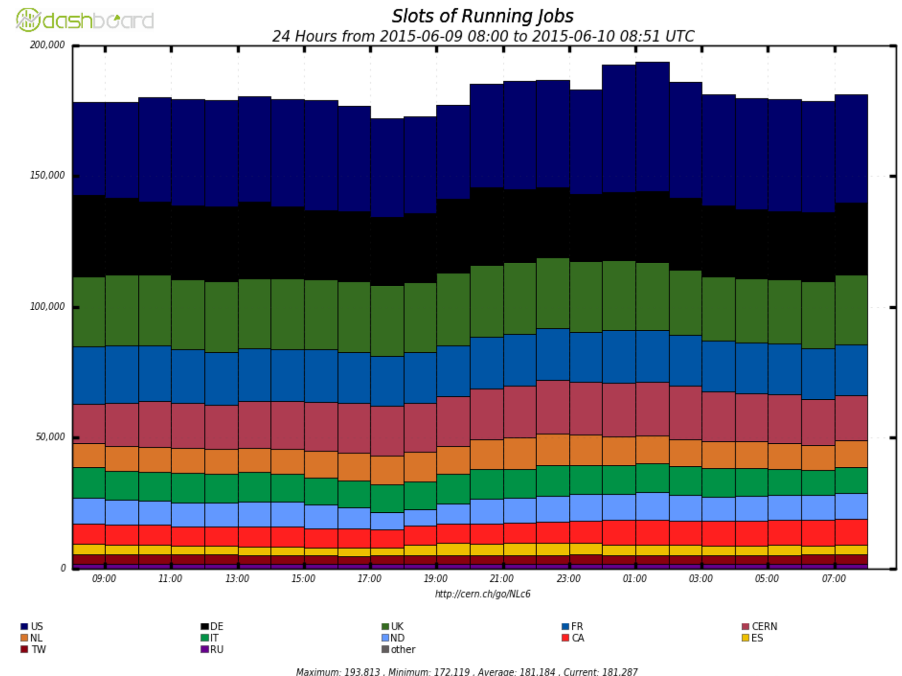
PanDA at a glance



Orders of magnitude



https://rucio-ui.cern.ch/



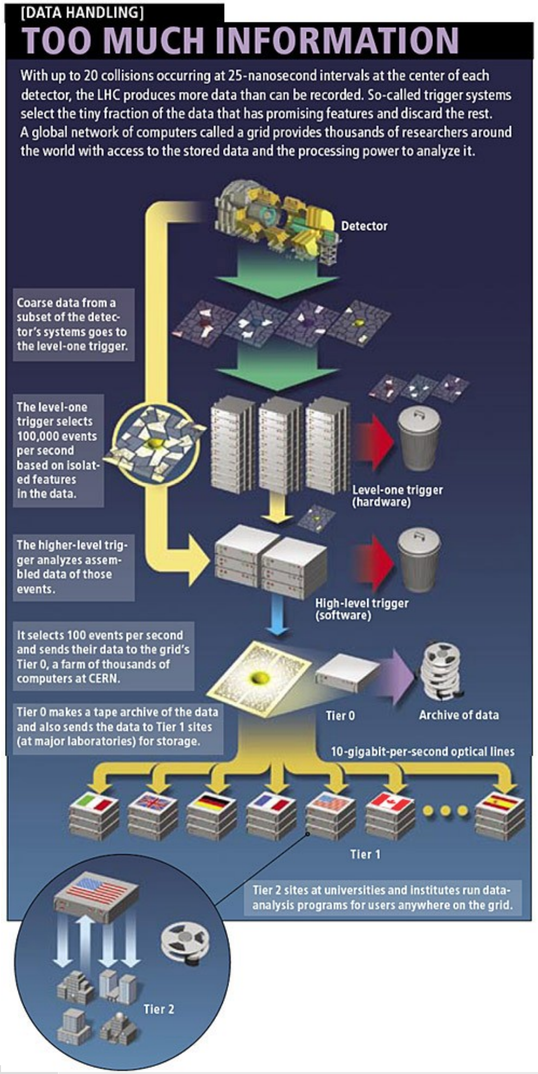
http://bigpanda.cern.ch/

Classic example of job defined with JDL file

```
[
JobType = “Normal”;
Executable = "$(CMS)/exe/sum.exe";
InputSandbox = {"/home/user/WP1testC","/home/file*”,
"/home/user/DATA/*"};
OutputSandbox = {“sim.err”, “test.out”, “sim.log"};
Requirements  = (other. GlueHostOperatingSystemName
== “linux") && (other.GlueCEPolicyMaxWallClockTime >
10000);
Rank = other.GlueCEStateFreeCPUs;
]
```

Data Management Concepts

- Services and tools that we will talk about can be applied to every file, but
- Data management is about specificall "big files"
  - bigger than 20Mb
  - In the orden of hundreds of MB
  - Optimized for working with this big files
- Generally speaking a file in the grid is
  - Read only
  - Cannot be modified,but
  - Can be deleted, so replaced
  - Managed by the VO, which is the "owner" of the data
  - Means that all members of the VO can read data.



- **Files are kept in Storage Elements (SE):**
  - Every site has to provide one
  - Consists of a data fabric and a interface to the grid
  - Authorization to store files is at the level of the Virtual Organization

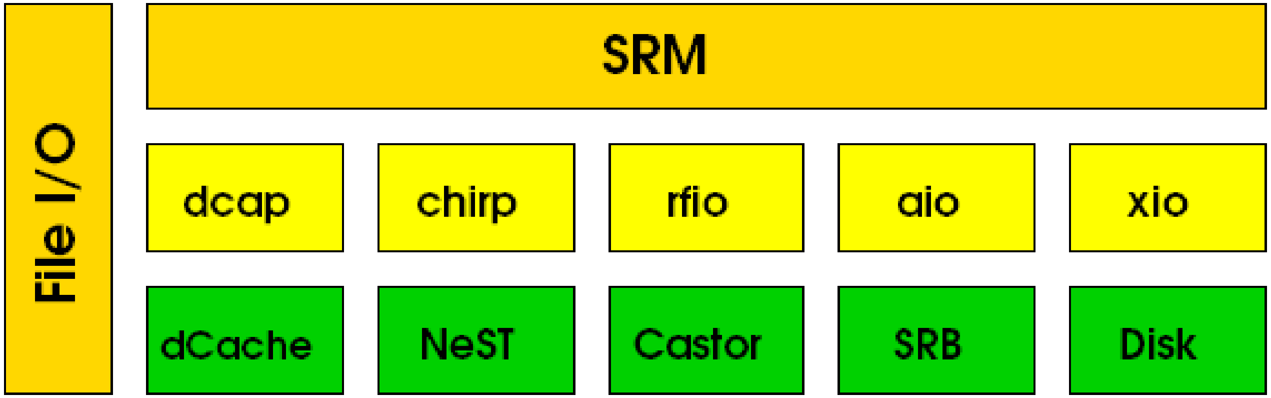
- **Files are replicated for availability and performance accessing to local replicas when needed:**
  - Need a unique identifier for a file (GUID)
  - Need a namespace model to easily locate files and replicas
  - Read only modes ease the replication inconsistencies

Data Services needed

- Where to store a file: Storage Element
- How lo locate a file:
  - GUID (not easy to remember)
  - LFN (Logical File Name, think as a link in lunix)
- But we need a method to associate the: FILE CATALOG
  - Provide a namespace for LFNs
  - Associate files with replicas
- Accessing the files
  - We can access by the file and the prococol if we know location
  - Or locate by the FILE catalogs.
  - Higher level tools to integrate all the services (ATLAS RUCIO)
- Other services:
  - To move data among SEs (FILE TRANSFER SERVICE)

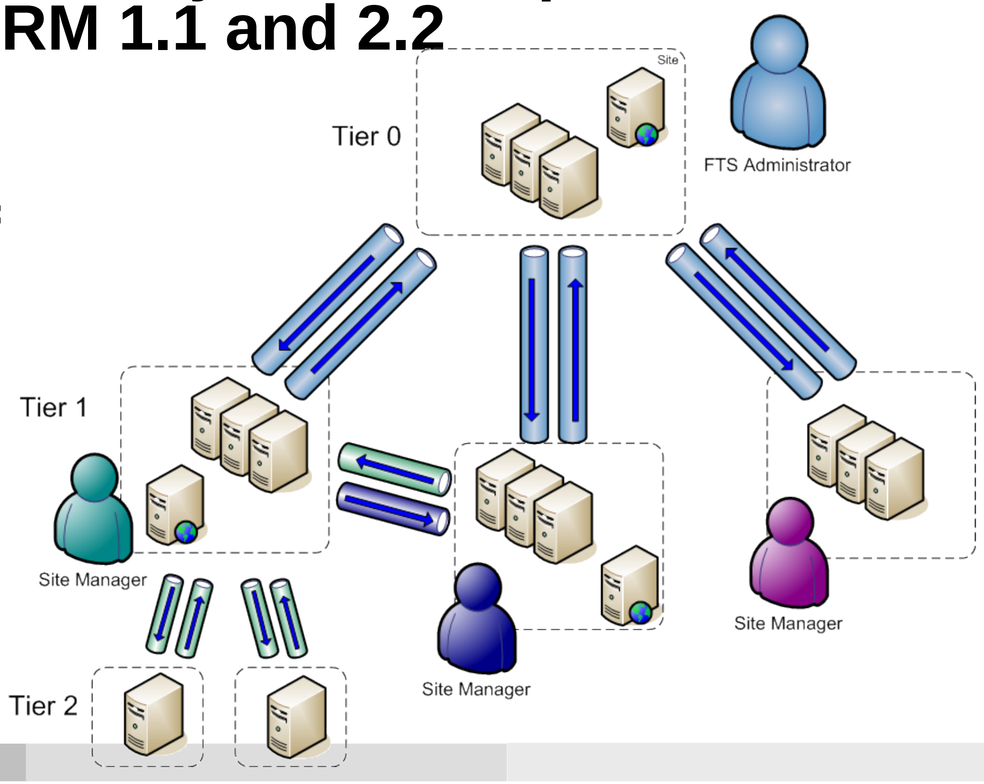
Storage Element

- **Storage Resource Manager (SRM)**
  - **hides the storage system implementation (disk or active tape)**
  - **handles authorization**
  - **translates SURLs (Storage URL) to TURLs (Transfer URLs)**
  - **disk-based: DPM, dCache,+; tape-based: Castor, dCache, StoRM**
- ***File I/O*: *posix-like* access from local nodes or the grid**
  - **→ GFAL (Grid File Access Layer)**



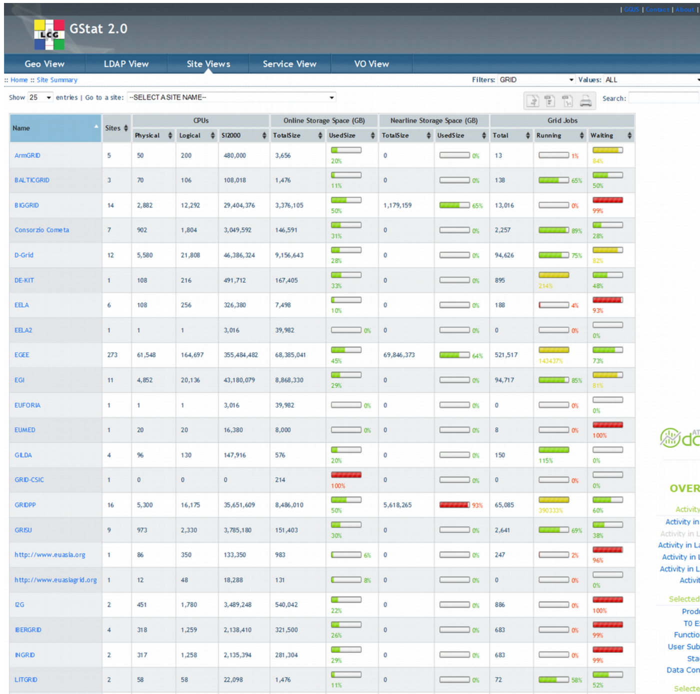
File Transfer Service

- **gLite File Transfer Service is a reliable data movement service (batch for file transfers)**
  - **FTS performs bulk file transfers between multiple sites**
  - **Transfers are made between any SRM-compliant storage elements (both SRM 1.1 and 2.2 supported)**
  - **It is a multi-VO service, used to balance usage of site resources according to the SLAs agreed between a site and the VOs it supports**
  - **VOMS aware**

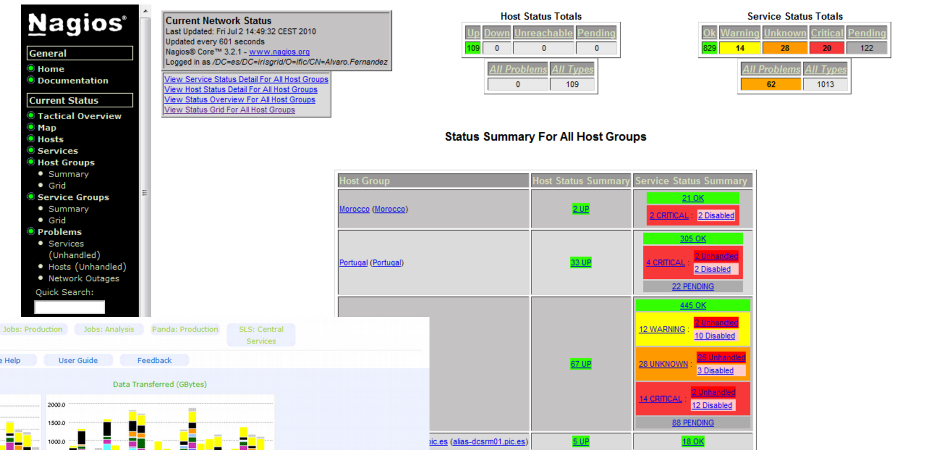


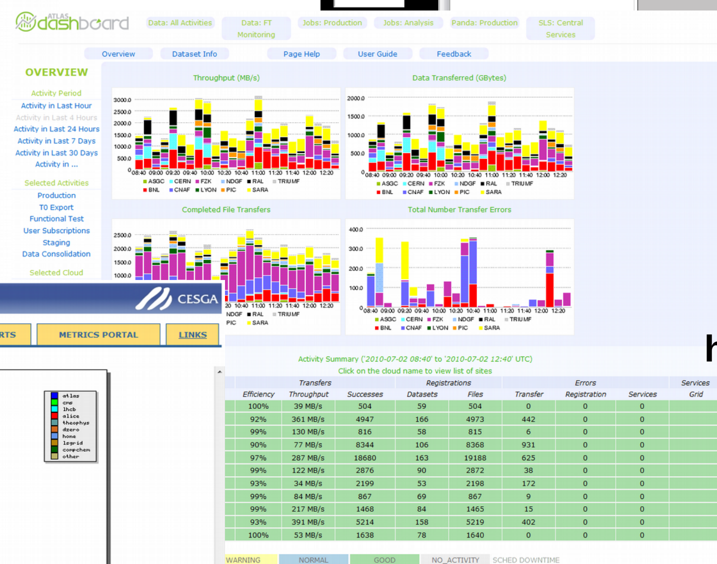
- **Why is it needed ?**
  - **High data volumes to be transferred (10^5-10^6 per day) with 1 GB as average file size. At most one intervention per day.**
  - **For the user,**
    - **reliable point to point movement of Site URLs (SURLs)**
    - **allocation of part of the sites' resources**
  - **For the site manager,**
    - **a reliable and manageable way of serving file movement requests from their Vos**
    - **easy way to discover problems with the overall service delivered to the users**
  - **For the VO production manager,**
    - **ability to control requests coming from his users**
  - **The focus is on the "service" delivered to the user**
    - **It makes it easy to do these things well with minimal manpower**

Monitoring and accounting

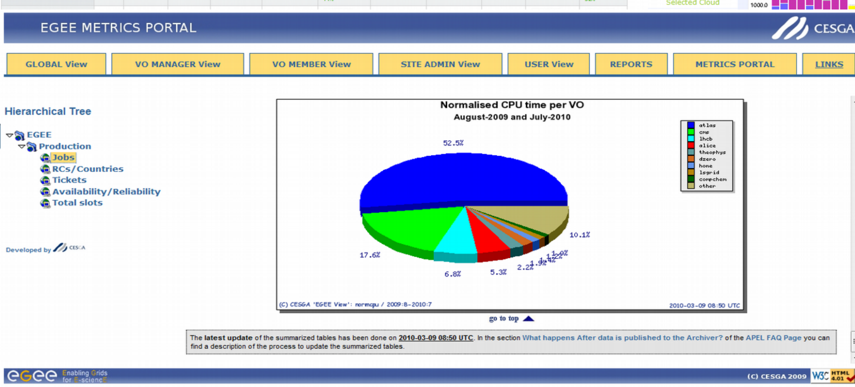


http://gstat-prod.cern.ch/





http://dashboard.cern.ch/

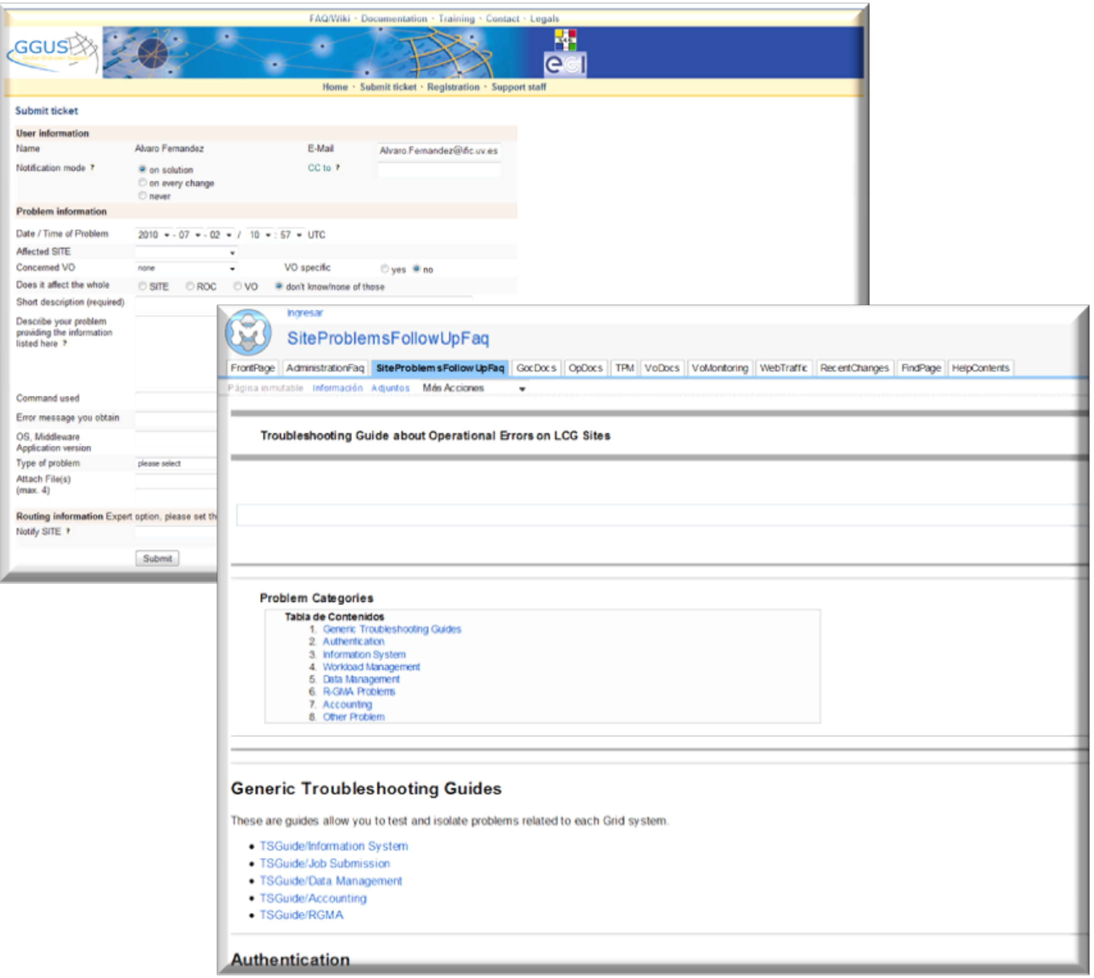


http://rtm.hep.ph.ic.ac.uk/

https://accounting.egi.eu/

http://dashb-wlcg-transfers.cern.ch/ui/#

Solving problems and Asking for Help?



- **Resources:**

https://ggus.eu/

- **Grid is Global: You can send tickets to solve remote problems**
- **Contact you Local Desk - Persons**

2017 Hot Technical topics

- Computing models
  - Different scenarios
  - Use of in-house, commercial, dedicated architectures, HPC, opportunistic, etc. resources
- Technology "choices" – may not be a choice but market-driven
- Data management and data access layer
  - End-to-end performance considerations; models of data delivery, event streaming, etc.
- Networking
- Resource provisioning layer
- Workload management layer
- Analysis facilities – how will analysis be done – traditional vs "query" vs ML, …
- These above lead to ideas about facilities and how they may look
- The stated (and agreed) intention in the CWP discussion is to make these components as common and non-experiment specific as possible
  - Clarify what really needs to be specific
- The CWP will provide the details of progress and R&D roadmaps in many key areas

LHCC: 9 May 2017 Ian Bird

Conclusions

- Grid computing is used since several years in production for multiple experiments (WLCG/ATLAS in particular in our case)
- Classic grid services still constitute the backbone of the computing resources for now.
- On top of these, the experiments build their own custom solutions.

- Sites (Resource Centres) provide their computing and data storage for the Experiment (Virtual Organization).
  - Lightweight site initiatives are expected to help reduce their complexity

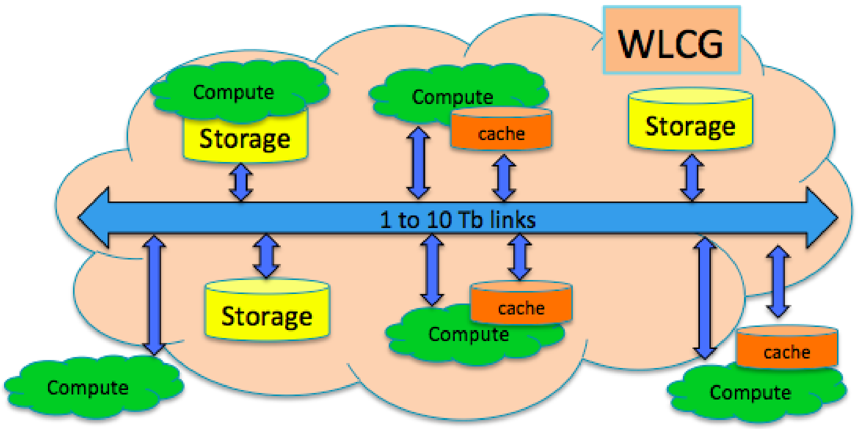
- New technologies and paradigms are gaining popularity
  - HPC usage, Containers and Singularity

BACKUP

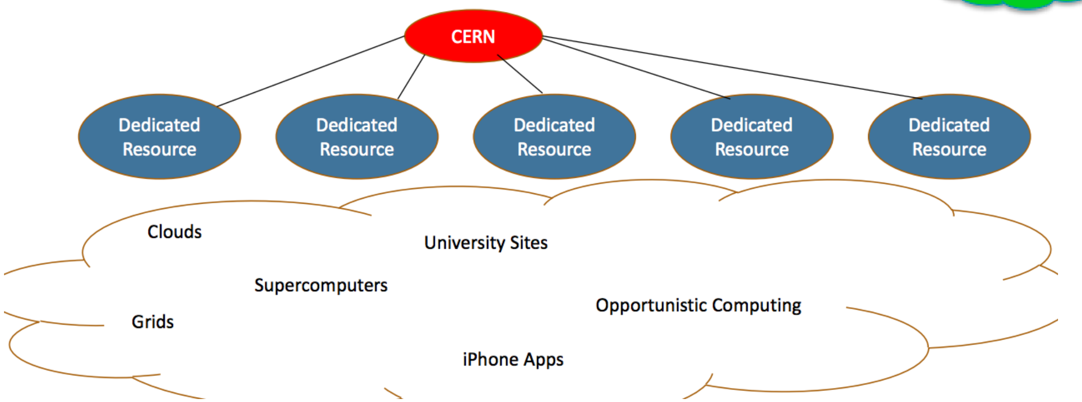
Evolution of the WLCG infrastructure

Evolution in the direction of

- Network centric model
- Consolidation of storage
- Diversification of facilities
- …



WLCG at HL-LHC



WLCG at HL-LHC (I. Fisk's representation)

- … diversification of compute resources

No need to wait 2026 for this

Compute Resources and experiments Computing Model

- Diversification of Compute Resources
  - Provisioning interface: Grid CE, cloud web service, login to batch head-node, none …
  - Resource availability: pledged vs opportunistic, flat vs elastic, …
  - Resource retention: from long living resources to highly volatile
- Diversification of Computing Models
  - LHC experiments are different. More obvious for Alice and LHCb (special physics focus)
  - Also true for ATLAS and CMS (different detector layouts and different sociologies)
  - This has an impact on the computing models, and motivates different choices and different focuses

Clouds

- LHC experiments leverage cloud resources for data processing. Two options:
  1. The experiment workflow management system instantiates VMs (generally through Condor) and the VMs join an experiment Condor pool to which pilots are submitted
  2. A grid site instantiates VMs and the VMs join the site batch system

- Effectively the same thing, but in (2) the facility administers the WNs while in (1) the experiment operates the WNs

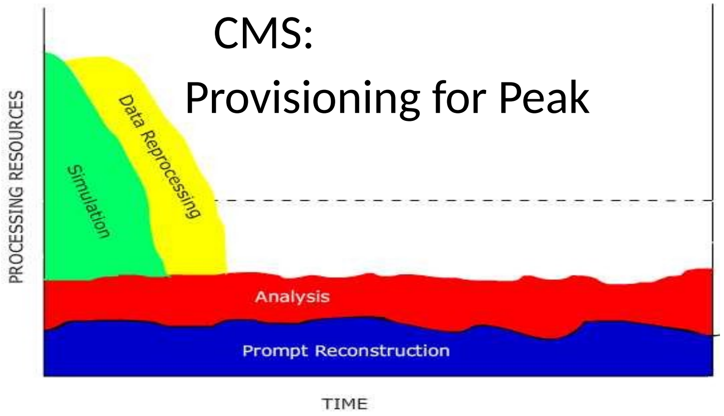
- I prefer (2)

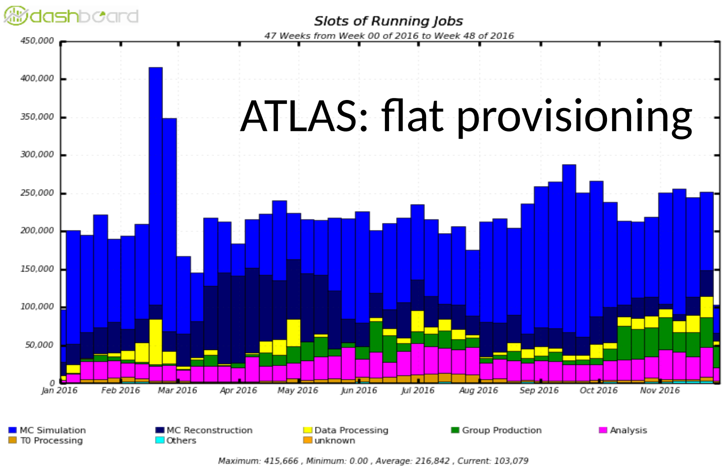
- Condor generalizes the provisioning and access to cloud processing units and proved to be reliable. A standard de facto.

Elastic Resources

- Different interests due to different needs
- ATLAS needs large CPU resources for relatively low priority G4 simulation (Calorimeters)

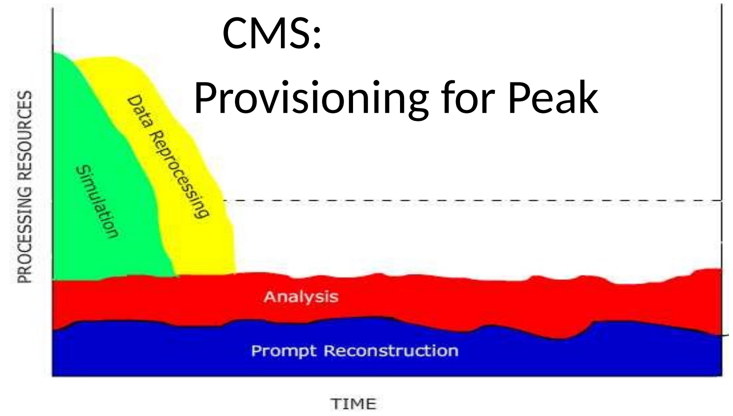
  => High priority tasks pushed through processing shares with flat provisioning
- CMS G4 simulation is faster, so a big interest in provisioning for peaks for bursty activities

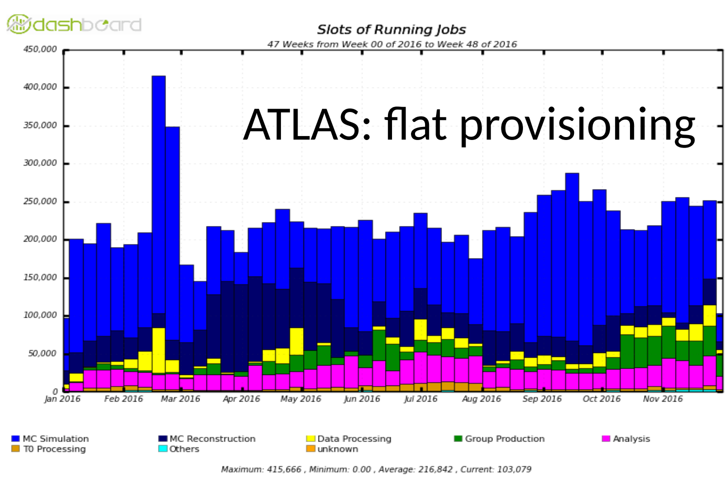




Elastic Resources

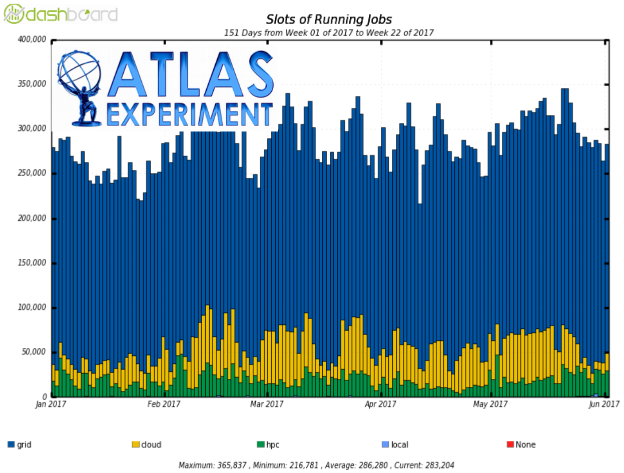
- Different interests due to different needs
- ATLAS needs large CPU resources for relatively low priority G4 simulation (Calorimeters)
  - => High priority tasks pushed through processing shares with flat provisioning
- CMS G4 simulation is faster, so a big interest in provisioning for peaks for bursty activities

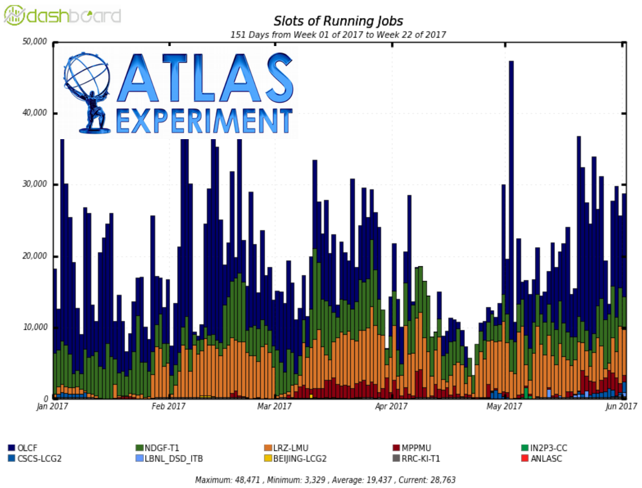




HPCs

- Rather complicated resources to exploit …
- The main challenge comes from diversity (e.g. site policies and CPU architectures)
- HPCs are built for a use case not very suitable for our embarrassingly parallel use case
- … but potentially a large number. So experiments invest on integrating them
- Invest in common tools. Software distribution, data handling, edge services. Try to negotiate consensually policies



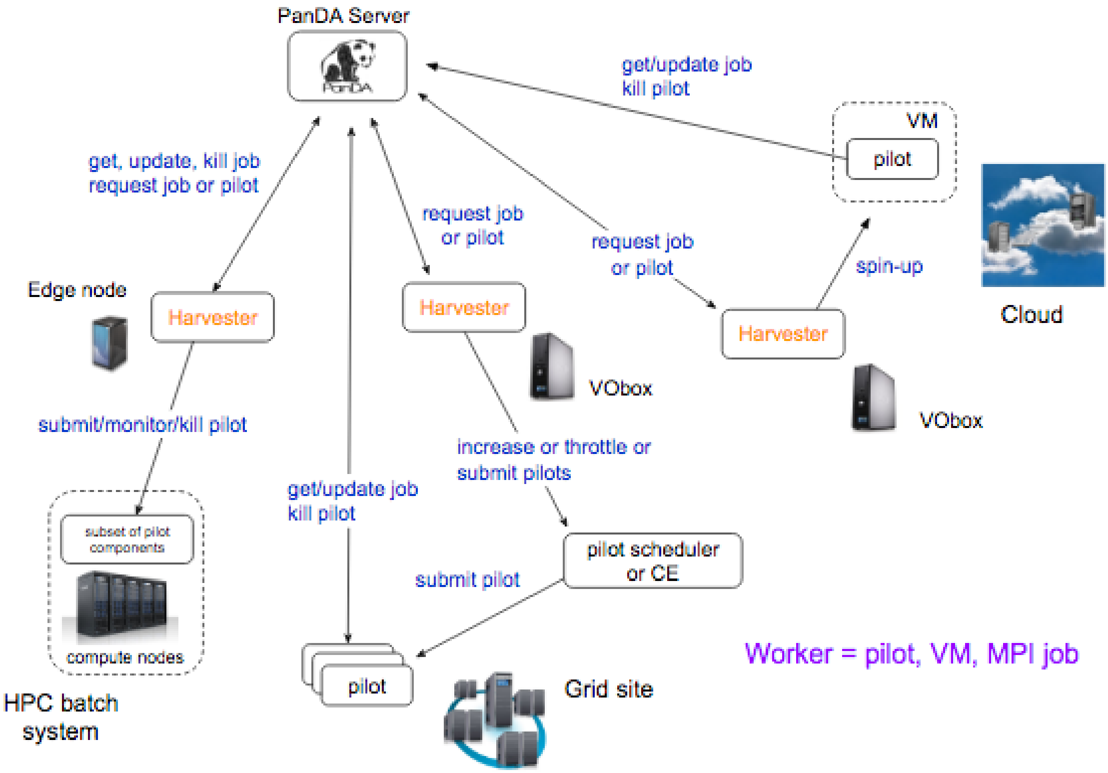


HPC resources

- HPC is not a natural match for experiment jobs
  - Fast interconnects not needed
  - Often no external network, no local disk
    - Getting better on new machines
- NorduGrid HPC sites use ARC functionality to have data staged in and out for jobs
- US HPC facilities each have different edge services and operational policies
  - Has led to ad-hoc complexities in the job frameworks of ATLAS and ALICE
- ATLAS working on new *Harvester* service as intermediary between PanDA and pilots
  - See this recent talk by Tadashi Maeno
  - Common architecture for HPC, grid, cloud, …

Rationalize provisioning layer

- In ATLAS, the strong need to rationalize resource provisioning at HPCs initiated Harvester project
- An edge service, creating a communication channel between resources and WMS (PanDA)
- A plugin-based architecture allows to leverage many of the functionalities for Grids and Clouds as well



Volatile resources

- Considerable processing capacity can be exploited for short periods of time (or with reduced QoS)
  - HLT farms between fills, Spot Market on Clouds, Backfill in HPCs, …
- In many cases, for such resources one can expect pre-emption at any time
  - Could be softer (SIGTERM followed by SIGKILL) or harder (SIGKILL with no merci nor regret)
- Several solutions on the table, with different advantages and different challenges
  - Machine Job Feature to size at runtime the workload depending on what the resource can offer
  - Reduce the data processing granularity in a check-pointable way to one(few) event(s)
  - Ad-hoc short jobs + extra care in retry, monitoring and allarming

Lightweight sites: Vac and @HOME

- Vac - clusters of autonomous hypervisors manage Vms
  - with Vac or Vcycle
  - ➢ In production at many UK sites, VMs in production for ALICE, ATLAS, LHCb, and VOs of the GridPP DIRAC service
  - ➢ Major component for LHCb integrating Clouds, HLT,
  - ➢ Generic VMCondor can connect VMs to an HTCondor pool
  - ➢ Report usage to APEL as “virtual” CEs

- LHC@HOME - interesting outreach project
- Provisions non negligible processing capacity (1-2% in ATLAS) and a technology for lightweight sites
  - ➢ Based on Boinc and various solutions for data handling

Containers

- A lightweight technology to provide isolation, with many benefits for experiments and sites. E.g. :
  - Payload isolation from pilot credentials
  - SL6 environment on other distributions
- A lot of (coherent) interest from the experiments
- Focus on Singularity for WN environment
- This workshop is a great opportunity to discuss and push this further