

WP1 activity, achievements and plans

Where we came from, what we did and learned, how we can put this to good use.



Francesco Prelz
Francesco.Prelz@mi.infn.it

Talk Outline

- ◆ Objectives of WP1, and how they matched the structure and timeline of the DataGrid project.
- ◆ The WP1 workload management solution, and what WP1 delivered over time.
- ◆ Lessons learned.
- ◆ Plans for the 'present' and the future.
- ◆ Questions. (10')

WP1 objectives



Or, where we started from...

- ◆ Task we were given in the TA: *"To define and implement a suitable architecture for distributed scheduling and resource management on a GRID environment"*.
 - Real life scientific applications meet 'computer science' grid projects, with mutual benefit.
 - And with the guarantee of a fast-paced three-year schedule.
 - ◆ First version of software to be delivered at project month 9!

In other words:

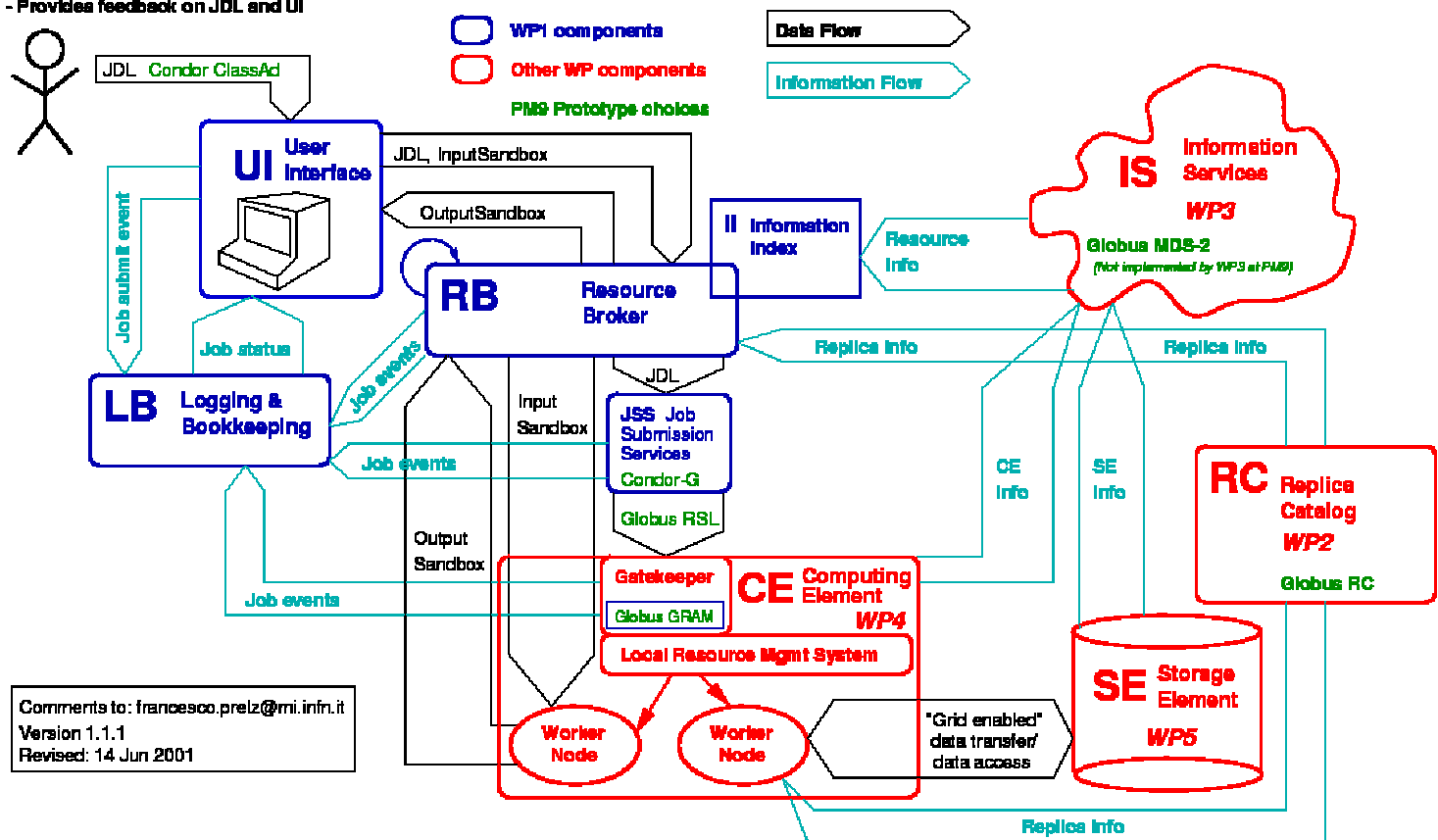
- We found little room for a traditional requirements/architecture/implementation/deployment cycle.
- But this was good to get early user involvement and feedback.

The beginnings of WP1

- ◆ A number of products and services that were in existence when the project started were evaluated, and many were adopted.

End User WP8,9,10
 - Specifies job using JDL
 - Submits job using UI
 - Controls and monitors job(s)
 - Provides feedback on JDL and UI

WP1 PM9 Integration



What did WP1 develop?



- ◆ To provide “an integrated workload management system, with distributed scheduling”, we needed to add:
 - A well-defined, as lightweight as reasonably achievable, User Interface, API and GUI.
 - A modular match-making/"brokering" service. Its policies make use of resource authorisation information and data requirements.
 - Scaffolding around CondorG, the “reliable job submission” component.
 - A Logging and Book-keeping service to build a uniform view of jobs out of information coming from many system components.

The screenshot shows the Job Monitor GUI with a menu bar (File, Checkpoint, Credential, Sort, Help) and a current time display of Fri Sep 12 16:06:07 CEST 2003. A job ID input field contains the URL https://ibm139.cnaf.infn.it:9000/OPF5RBIJU3usyjk1L_JL3w. Below the table, there are buttons for Details, Log Info, Job Cancel, Job Output, Update, and Back.

Job Id	Job Type	Status	Submit...	Destination
https://ibm139.cnaf.infn.it:9000/ayh6...	normal	Submitted		
https://ibm139.cnaf.infn.it:9000/wOf...	normal	Submitted		
https://ibm139.cnaf.infn.it:9000/HUKI...	normal	Submitted		
https://ibm139.cnaf.infn.it:9000/CDN...	normal	Submitted		
https://ibm139.cnaf.infn.it:9000/EEZr...		- Status Err...		
https://ibm139.cnaf.infn.it:9000/ONd...	normal	Submitted		
https://ibm139.cnaf.infn.it:9000/VPZj...	normal	Aborted		skurut.cesnet.cz:2...
https://ibm139.cnaf.infn.it:9000/OPF...	normal	Done		lxde01.pd.infn.it:...
https://ibm139.cnaf.infn.it:9000/0qj9...	normal	Done		lxde01.pd.infn.it:...
https://ibm139.cnaf.infn.it:9000/tQzZ...	normal	Done		lxde01.pd.infn.it:...
https://ibm139.cnaf.infn.it:9000/wZ7...	normal	Aborted		bbq.mi.infn.it:211...
https://ibm139.cnaf.infn.it:9000/WRm...	normal	Aborted		
https://ibm139.cnaf.infn.it:9000/rUyQ...	normal	Waiting		

The WP1 development cycle

- ◆ The first WP1 prototype release was delivered at PM 10.
- ◆ It was functional, with the novel ability to deal with data requirements, but soon met a *very* steep scalability barrier.
- ◆ Early feedback and troubleshooting in the field were vital in locating the trouble spots (will talk briefly about this process).
- ◆ The experience was merged into the design for Release 2.
- ◆ Batch, interactive, parallel and checkpointable jobs are supported.
- ◆ Release 2 survived tests to the scale of 10000 workload requests a day. The services on an individual RB node are routinely tested with storms of 2000 requests from 20 concurrent users.
- ◆ The cycle is continuing...

Expectations and reality



◆ Problems we expected:

- Service scalability problems, with increasing user and job load.
- We imagined we would simply hit the resource *roof* on one node, and address this with service configuration.

◆ Problems where we spent most time:

- Controlling the direct and indirect interplay of the various integrated components.
- Addressing stability issues and bottlenecks in a non-linear system.
- Predicting (or failing to predict) *where* the *next* bottleneck will appear in the job processing network.

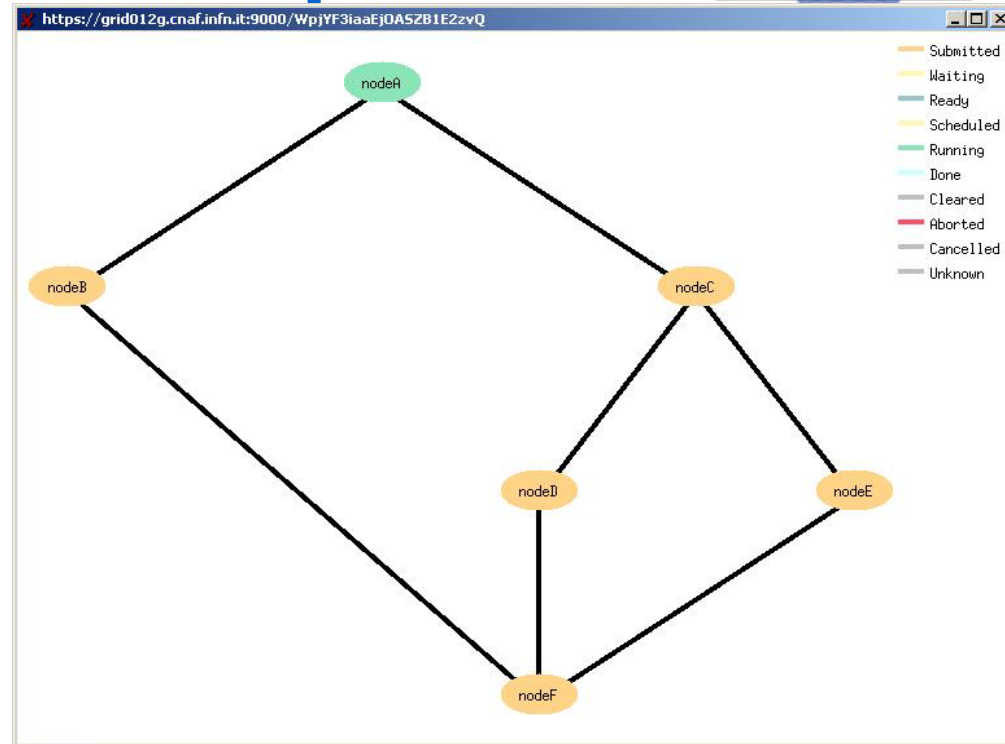
Lessons we learned



- ◆ Many “principles” at the level of implementation:
 - Always apply *reliable*, properly backtraceable communication among services
 - For every component there should be another one ‘taking care’ of it.
 - Monolithic, long-lived processes should be avoided when there are many, complex layers of external dependencies.
 - Etc, etc...: these could be captured for Release 2!
- ◆ The effort of integrating together 30+ external packages tends to be minimised.
- ◆ A test strategy should appear at the same time as the code.
- ◆ To safely process thousands of requests through, the ‘high watermark’ of the system should be in the tens of thousands...
 - ➔ Many thanks to our colleagues in the LCG Certification Group.

What else did WP1 develop ?

- ◆ Grid accounting infrastructure, based on an economic model (DGAS).
- ◆ Infrastructure and user API for automatically partitionable jobs.
- ◆ Support for job dependencies (via DAGman), demonstrated at this review.



EDG-WL-DGAS-INTERFACE

Account type Identity

Server

ID	type	.509 subje	gid	credits	wall time	cpu time	job
Andrea Guarise	User	/C=IT/O=II	wp1 -84088	170100	95370	394	
grid002A	CE	NA	wp1 17598	34577	21896	92	
grid002B	CE	NA	wp1 17518	32185	21750	93	

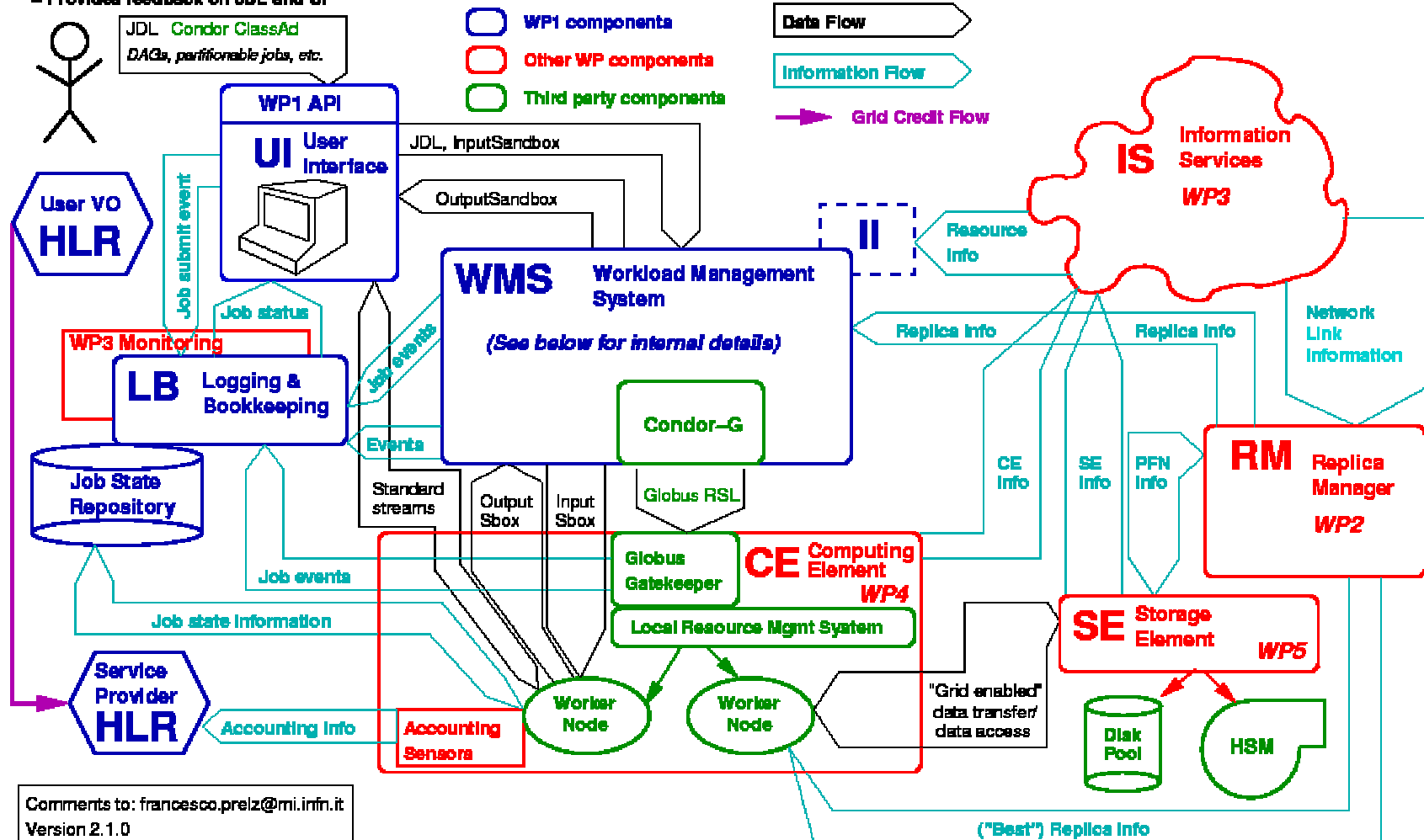
auto refres

A much more complex picture...

WP1 Final Integration

End User

- Specifies job using JDL
- Submits job using UI
- Controls and monitors job(s)
- Provides feedback on JDL and UI



The bright future...

- ◆ LCG, CrossGrid, DataTAG, GridPP and `grid.it` are now using and developing on the WP1 code. We keep supporting them.
- ◆ There still are few issues to be addressed to harden the system:
 - Collection/distribution of resource information.
How do we get to know that a resource is there?.
 - Resource access.
How do we know that a resource is available now ?
 - Handling of data movement/"placement".
How do we move data at the right time?
 - Streamlining of error reporting.
Why is this still going wrong?
 - System scalability.
How can I submit this 1 million jobs and keep them together?
- ◆ There is also a commitment by basically all partners and staff of WP1 to continue in the context of future projects.

Thanks everyone!

- ◆ We thank the European Commission for having given us the opportunity for development in this field.
- ◆ We acknowledge the support of our funding agencies: INFN, CESNET and PPARC.
- ◆ We value very much the cultural exchange and partnership with Datamat Spa, our industrial partner.
- ◆ We wish to thank all the colleagues in DataGrid for the professional growth coming from our many interchanges.

Thanks to you all.

A little bibliography

- Home page for the Grid Workload Management workpackage of the DataGrid project: <http://www.infn.it/workload-grid>.
- Home page for the Condor project <http://www.cs.wisc.edu/condor/>
- Home page for the Globus project <http://www.globus.org/>
- EU DataGrid WP1 (G. Avellino *et al.*), "The EU DataGrid Workload Management System: towards the second major release", Talk from the 2003 Computing in High Energy and Nuclear Physics (CHEP03), La Jolla, Ca, USA, March 2003, [Get PDF](#).
- EU DataGrid WP1 (G. Avellino *et al.*), "The first deployment of workload management services on the EU DataGrid Testbed: feedback on design and implementation", Talk from the 2003 Computing in High Energy and Nuclear Physics (CHEP03), La Jolla, Ca, USA, March 2003, [Get PDF](#).