



Enabling Grids for
E-science in Europe

www.eu-egee.org

IT/CZ cluster meeting, Milano, 8-9 June 2004

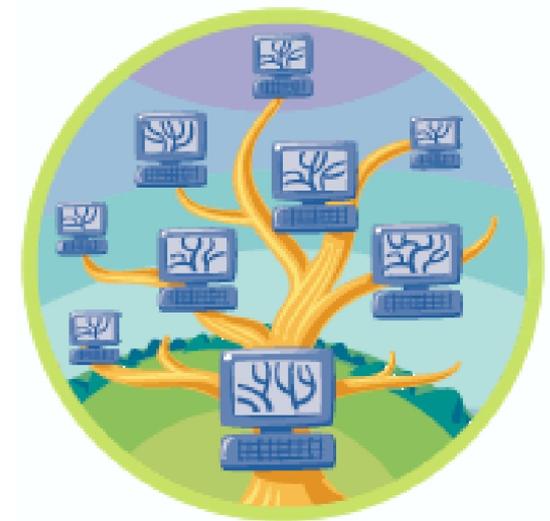
CESNET Status Report

Aleš Křenek



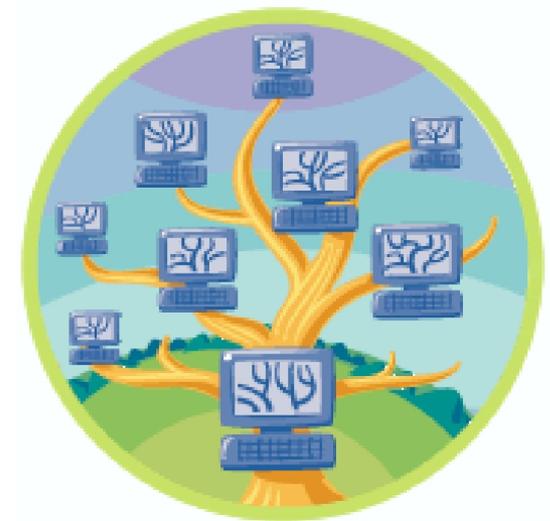
Overview of Activities

- DJRA1.1 – design discussion on Job Provenance



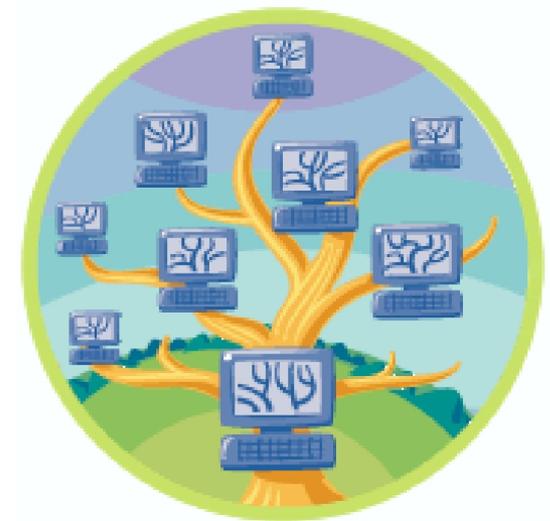
Overview of Activities

- DJRA1.1 – design discussion on Job Provenance
- migration to SCM



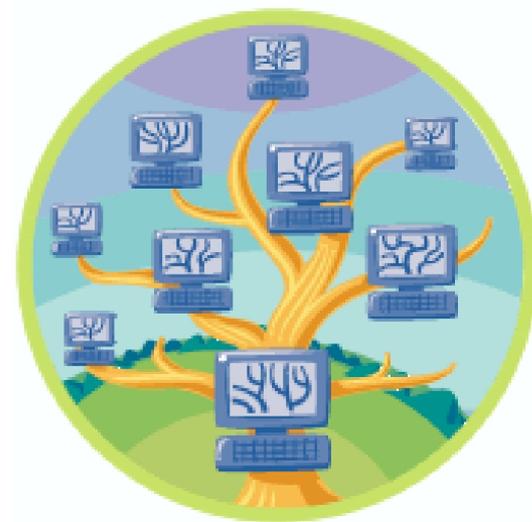
Overview of Activities

- DJRA1.1 – design discussion on Job Provenance
- migration to SCM
- “R-GMA convergence”



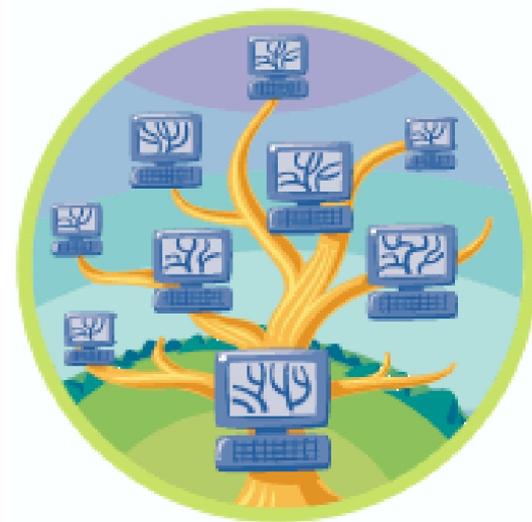
Overview of Activities

- DJRA1.1 – design discussion on Job Provenance
- migration to SCM
- “R-GMA convergence”
- HOLUB service
 - extended interlogger functionality almost completed
 - refinements in client library (thread safety)
 - C++ wrapper ready, not tested thoroughly
 - bugs are quite sure to be there :-)



Overview of Activities

- DJRA1.1 – design discussion on Job Provenance
- migration to SCM
- “R-GMA convergence”
- HOLUB service
 - extended interlogger functionality almost completed
 - refinements in client library (thread safety)
 - C++ wrapper ready, not tested thoroughly
 - bugs are quite sure to be there :-)
- replacement of `ssl_utils` with GSS API
- LCG support



- using plain Makefile's (not autoconf/automake)
- libtool for building libraries and linking with them
- configuration information (location of dependencies etc.) taken from the build environment (ant properties), dumped into `Makefile.inc` within `localinit` target
- current L&B modules (may be still changed ...):
 - `org.glite.lb` - top level subsystem
 - `org.glite.lb.client-iface` - interface only (.h's), public (i.e. gets installed)
 - `org.glite.lb.common` - original L&B common (network communication, protocol parsers, ...)
 - `org.glite.lb.client` - implementation of the client library
 - `org.glite.lb.logger` - local- and interlogger
 - `org.glite.lb.server` - bookkeeping server, dump & load, indexing utilities
- build framework ready (mostly), L&B common module nearly migrated
- CSF missing (I need support; is it critical for the July Big Bang?)
- feasible deadlines: June 18 for L&B, June 30 for proxy renewal

Migration to SCM – general comments

(maybe already obsolete)

- not all dependencies of `org.glite.wms` pulled in correctly
- sometimes it seems unnecessary stuff is recompiled
- still `edg::*` namespaces – what are the plans for change?
- hardcoded directories and software versions in `configure.ac`
- `#include "../other/header.h"`
 - ugly but works as long as it remains within a module
 - would break compilation when exported eventually
 - suggestion: don't include .h's of the same module in other .h's
 - interface-only modules are encouraged, it's the only way to avoid cyclic dependence

- Current situation
 - L&B as an infrastructure (not regarding the meaning of delivered data) is GMA in fact, its components can be mapped to GMA ones
 - L&B preferences are different (e.g. reliability and security in favour of generality)
 - relational model is quite suitable for L&B
 - two independent monitoring architectures in the EGEE middleware, having many common features

- Current situation
 - L&B as an infrastructure (not regarding the meaning of delivered data) is GMA in fact, its components can be mapped to GMA ones
 - L&B preferences are different (e.g. reliability and security in favour of generality)
 - relational model is quite suitable for L&B
 - two independent monitoring architectures in the EGEE middleware, having many common features
- Foreseen goals
 - converge to a common abstract “GLite monitoring architecture”
 - define clean interfaces (WS) so that different implementations (with emphasis on different features) are possible
 - implement the required components for new generation R-GMA compliant L&B

R-GMA Convergence (cont'd)

- Requirements on the R-GMA group
 - stable internal R-GMA interfaces (WS) between producers, consumers, and registry eventually – they are going this way
 - collaboration in negotiating changes to those interfaces
 - extraction of certain functionality (mediator – resource discovery) from R-GMA Java libraries into a standalone service

R-GMA Convergence (cont'd)

- Requirements on the R-GMA group
 - stable internal R-GMA interfaces (WS) between producers, consumers, and registry eventually – they are going this way
 - collaboration in negotiating changes to those interfaces
 - extraction of certain functionality (mediator – resource discovery) from R-GMA Java libraries into a standalone service
- State of negotiation
 - semi-formal detailed proposal sent to R-GMA people
 - first reaction was positive
 - dedicated meeting scheduled in RAL, just before all-JRA1