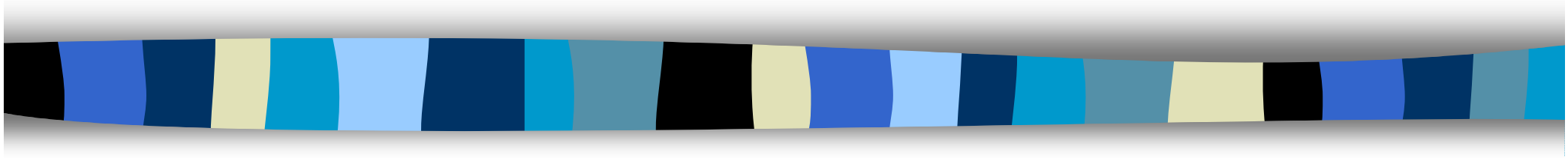
JTB meeting: Update about Certification

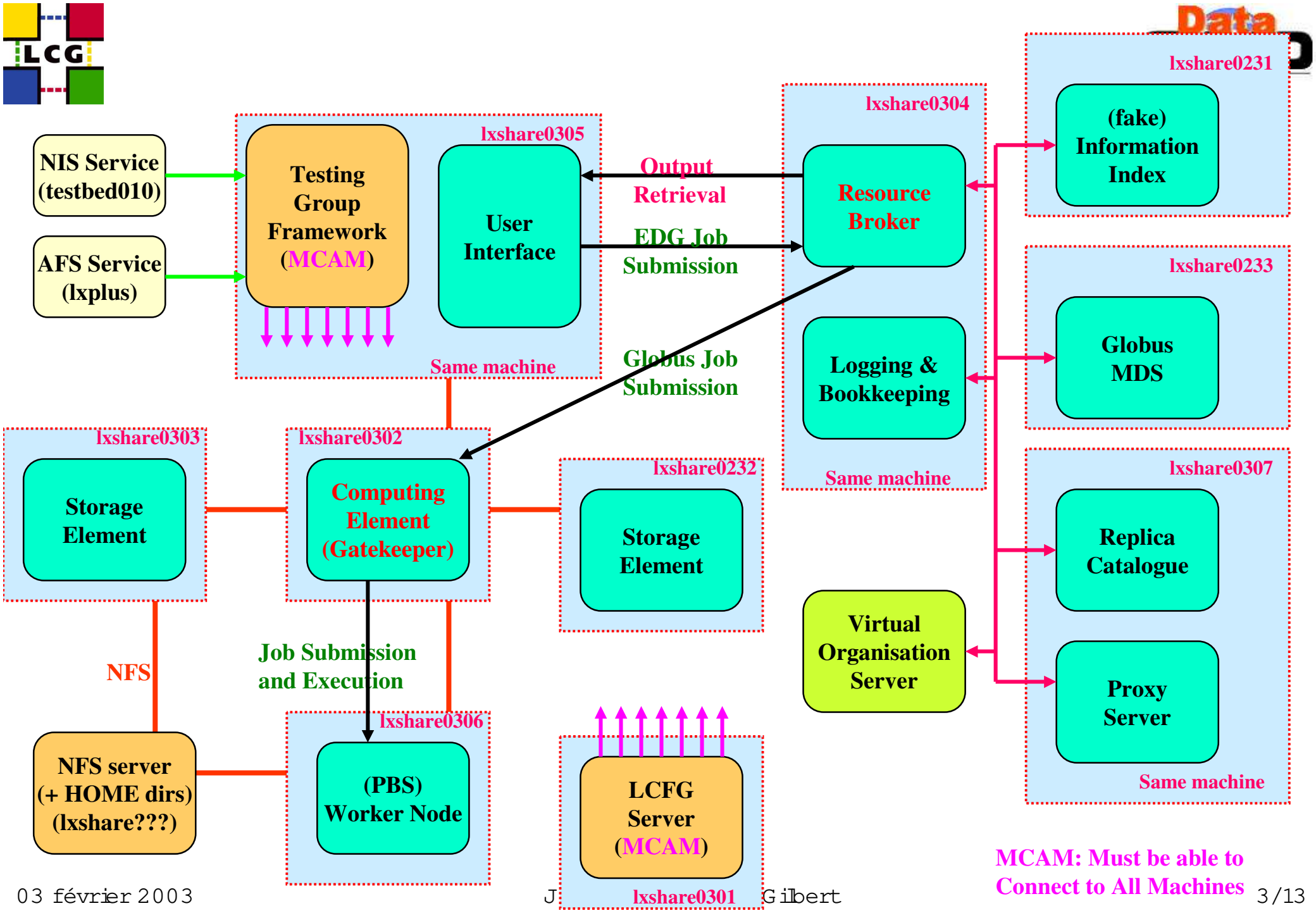


Gilbert Grosdidier

(LAL-Orsay / IN2P3 / CNRS)

Testing Suites: Goals

- Testing Suites can be aiming at 4 targets
  - Check that a GRID cluster is correctly installed and running
  - Check that a newly built M/W release is coherent and complete
  - Check that a Production cluster is fully operational
  - Check that a Testing Suite is adequate, and covers all bases
- The current goal for us is building/testing the main script
  - meaning the 4th target
- To check our suites, we need
  - a well known/understood M/W version (currently edg-1.4.3)
  - a full GRID cluster with a controlled install
    - available at CERN
    - installed thru LCFG (reproducible)
    - UI, CE, 2xSEs, RB, LB, WN, MDS, (BDII), PX, RC
      - there are 2 SEs to allow for full data managements tests
  - this GRID cluster will evolve to something more realistic when needs arise


lxshare0231
(fake) Information Index
lxshare0304
Resource Broker
lxshare0305
NIS Service (testbed010)
AFS Service (lxplus)
Testing Group Framework (MCAM)
User Interface
Output Retrieval
EDG Job Submission
Logging & Bookkeeping
lxshare0233
Globus MDS
Same machine
Globus Job Submission
lxshare0303
Storage Element
lxshare0302
Computing Element (Gatekeeper)
lxshare0232
Storage Element
lxshare0307
Replica Catalogue
Virtual Organisation Server
Proxy Server
Same machine
NFS
Job Submission and Execution
lxshare0306
NFS server (+ HOME dirs) (lxshare???)
(PBS) Worker Node
LCFG Server (MCAM)
lxshare0301
MCAM: Must be able to Connect to All Machines

Testing Suites: Structure

- These tools currently exist under 2 forms
  - GUI tool installed and running locally on each machine of the cluster
    - required by sysadmins when achieving manual or non-standards install
  - a large Testing Suite being run centrally from a single machine
    - covering however all the components of the cluster
    - implies some constraining requirements for accessing the remote nodes
    - allows for running in an automated framework, or thru a cron job
      - for Build checking or Production cluster checking
- They share basic scripts as much as possible
- I'll mainly cover the 2nd point (large Testing Suite)
- Two languages used: **bash** and **Perl** (5.005003)
  - bash used for building basic scripts
  - but slowly becoming minority, because Perl offering more functionalities
    - while being a little bit more cryptic
  - Then it is strongly advised to use Perl to develop new tools
    - we will have to wrap bash scripts to embed them into OO Perl suites

Testing Suites: Functionalities

- There are many areas to cover in these tests
  - even when running high level tests (M/W global functionality check, or cluster stress testing), one has to make sure that the lower levels are fine
- We currently distinguish these levels:
  - **Install & Configuration** level
    - checking RPMs, config and log files, and so on
  - **Unit Testing** (and Daemon testing ?): single component functionality checks
    - ex: is the Globus gatekeeper running fine on this CE ?
  - **Global tests**: full functionality checks, involving several components, with single jobs, embedding different kinds of functionalities
    - ex: insertion of a new entry in the RC, + creation and replication of the file
    - ex: auto-coherency of the II, coherency with installed resources, is the II consistent with a smooth running of the GRID cluster ?
  - **Stress testing**: job storms and so on
    - ex: running several 100s of jobs, retrieving their outputs, checking it's OK
    - ex: same thing with copy storms
  - Stability and **Robustness** testing (standby right now)
    - is everything coming up fine and running after a reboot/shutdown/breakdown ?
    - probably aiming at Production cluster tests only
    - are **Security issues** part of this item ?

Testing Suites: Status

- Install & Config level
  - These are static tests
  - For CE, SE, RB, LB, UI, WN
    - Basics scripts: ready
    - Merging to Main Script: ready for all but SE
    - Migration towards edg-1.4.3, and testing: ready for all but SE
    - Presenter wrapping: ready for RB only
      - ideas available
  - For MDS, BDII, PX, RC
    - nothing ready yet
    - looking for manpower
  - later: R-GMA, VO Server
    - looking for manpower as well

Testing Suites: Status (2)

- Daemon Testing
  - some of the daemon testing is currently part of previous level
    - status: fair
    - would rather have it here, but no serious harm
  - will have to integrate monitoring sensors developed elsewhere in EDG
    - no manpower assigned
- Unit Testing
  - many tools provided thru the ITeam already
    - Gatekeeper, GridFTP, II, MDS
    - status:
      - scripts under Perl OO ready
      - merged into the Main Script
      - Presenter feature not ready yet
        - » ideas exist, manpower assigned
  - most probably missing many areas, IMHO
    - will have to dig into material provided thru the WPs (WP1 Testing Plan ?)
    - will have to cross-check with input from loose cannons

Testing Suites: Status (3)

- Global Testing
  - improved tools provided thru the IT eam are available
    - status: fair, merged into the Main Script, Perl OO
    - no presenter wrapping yet
    - overall cluster testing with simple requests
      - full UI+RB+CE functionality
      - long job features
      - BrokerInfo functionality
  - II Consistency check (to allow for smooth running)
    - being developed, ideas exist, no manpower clearly assigned
- Stress Testing
  - framework for a job storm already exists
    - the script for submitting it also exists
    - easy to adapt to Perl OO
  - merged into the Main Script
  - submitting a simple JDL file
  - no presenter wrapping yet

About configuration files

- It is foreseen to use 3 different configuration files
  - the testing framework configuration file
    - describing the path to the scripts, and other utility variables
  - the site configuration file
    - describing the cluster where the tests will be running
    - it is currently built from or extracted from the LCFG config file
  - the M/W configuration file
    - where are located the M/W version
    - containing pointers to version specific features
    - hope to have this one very small
- These config files are sourced by the Main Script upon starting
  - and every value is inherited from this
- We should stick to this model
  - and avoid having too many files to source to describe the different parts
  - no embedded sourcing: no file calling a file calling another file...
    - unless at the very beginning of the implementation process
  - clear sections into the main file are required
    - together with explanations :-)

Presenting the results

- Choosen: the output of each test will be automatically parsed
  - they will be merged onto one or several HTML pages
  - under the form of a result matrix (or matrices)
  - several tests will be merged to allow for fast scanning
- Every entry should itself be a link
  - towards a more detailed page, allowing for in depth checking
  - titles and/or description of each test (hence its meaning) will also be available on the same page, thru links
- A (rather simple) example is available here:
  - http://grodid.home.cern.ch/grodid/test3101/

QuickTime™ et un décompresseur TIFF (LZW) sont requis pour visionner cette image.

Expectations for VDT testing

- What to provide ?
  - (Perl) Scripts for Install&Config level, as well as for Unit Testing, ASAP
  - Later, Robustness tests
  - Global functionality tests, if they have to differ from the EDG tests for the same level
  - Care of Stress Testing should be taken with the existing framework, reusing the previous level scripts
- What will happen when the next version will be released ? As for our EDG partners:
  - the new version of the new tests scripts must be released altogether
  - or a specific advise should be sent, indicating how to update the testing scripts of the previous release

About the PerlOO framework

- This is inherited and adapted from Cal
- Current OO methods for the BaseTest Class
  - sub new
  - sub setContext
  - sub process
  - sub checkArguments
  - sub setDefaults
  - sub checkPrerequisites
  - sub setupTest
  - sub runTest
  - sub evaluateTest
  - sub cleanup
  - sub printHelp
  - sub printDesc
  - sub addOption
  - sub setErrorMessage
  - sub checkError
- Proposal: add new methods for
  - analyzing the output
  - formatting the results
    - sub buildConfFile
    - sub runParserScript
  - they willbe called from the evaluatetest method