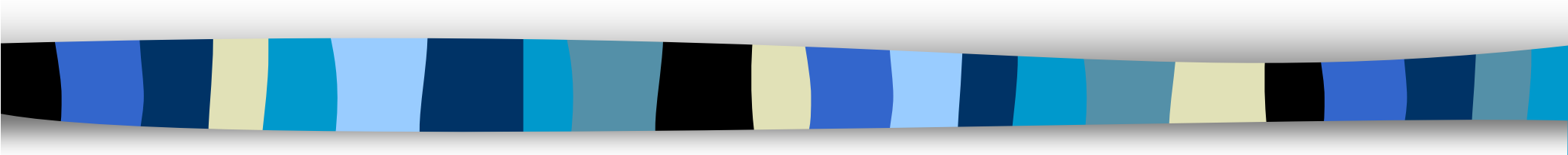




Current Status of the Certification TestBed

A decorative horizontal bar composed of various colored segments (black, blue, light blue, yellow, dark blue, cyan, grey) arranged in a pattern that tapers at both ends.

Gonzalo Merino
(*IFAE-Barcelona*)
on behalf of
Gilbert Grosdidier
(*LAL-Orsay / IN2P3 / CNRS*)



Role of the Certification Testbed

- The long term framework
 - 0) Nightly build for **one platform** (RH 6.2 or RH 7.2)
 - provides middleware components under form of RPMs
 - **1) On a local Cert. TB** (if some quality threshold QL0 is reached)
 - **reinstalls a small cluster of local interconnected machines (5/6-?)**
 - **reboots the machines, and starts the required daemons**
 - **runs the middleware testsuites and builds a quality map for these components**
 - **2) If another quality threshold QL1 is reached**
 - **enlarge the testbed by including remote certification testbeds**
 - **again only for middleware component testing, not applications components**
 - **I think these must also be only small certification testbeds (CE+SE+WNs)**
 - 1) If some quality threshold QL2 is reached
 - release the corresponding build to the Application TB, to run now wider tests
 - **these are now application test suites** (managed by WP8-10)



About this Local Cert. TB

- Focusing on first 2 steps (#0 and #1 on previous slide)
 - all 3 tasks must be fully automated (**build, install, testing**)
 - meaning they must be able to be running from scripts
 - every output must be automatically scanned to proceed to next step
 - Testing step will not target any application packages at this stage
 - only global or unit components functionality tests are foreseen
- About step #1 (the **local Cert. TB, only at CERN**)
 - it will first be a **minimal grid** to allow for testing **every component**
 - 1 CE, 1 SE, 1 RB, 1 UI, 1 WN, 1 (MDS + Proxy Server + RC)
 - to allow for RC+RM+SE testing, it will evolve **very soon** to
 - 1 CE, **2 SEs**, 1 RB, 1 UI, 1 WN, 1 (MDS + Proxy Server + RC)
 - to go to something more **realistic**, it must become, in a 3rd stage
 - 1 RB + 1 UI + **3 * (1 CE + 1 SE + 2 WNs)** + 1 (MDS + Proxy Server + RC)
- Step #2, involving remote centers, will be detailed on slide 7



What could be in the M/W testing suites ?

■ Examples:

- are all daemons (ex: gatekeeper, wu_ftp, information servers, ...) present ?
- are they all alive (simple responsiveness, like a ping) ?
- sophisticated internal test of each daemon
 - with extensive checking of each major function separately
 - then with all merged functionalities
- job storms (many jobs submitted within small delays)
- replica storms (many replication requests submitted within small delays)
- network storms (don't know yet how to generate this one :-)

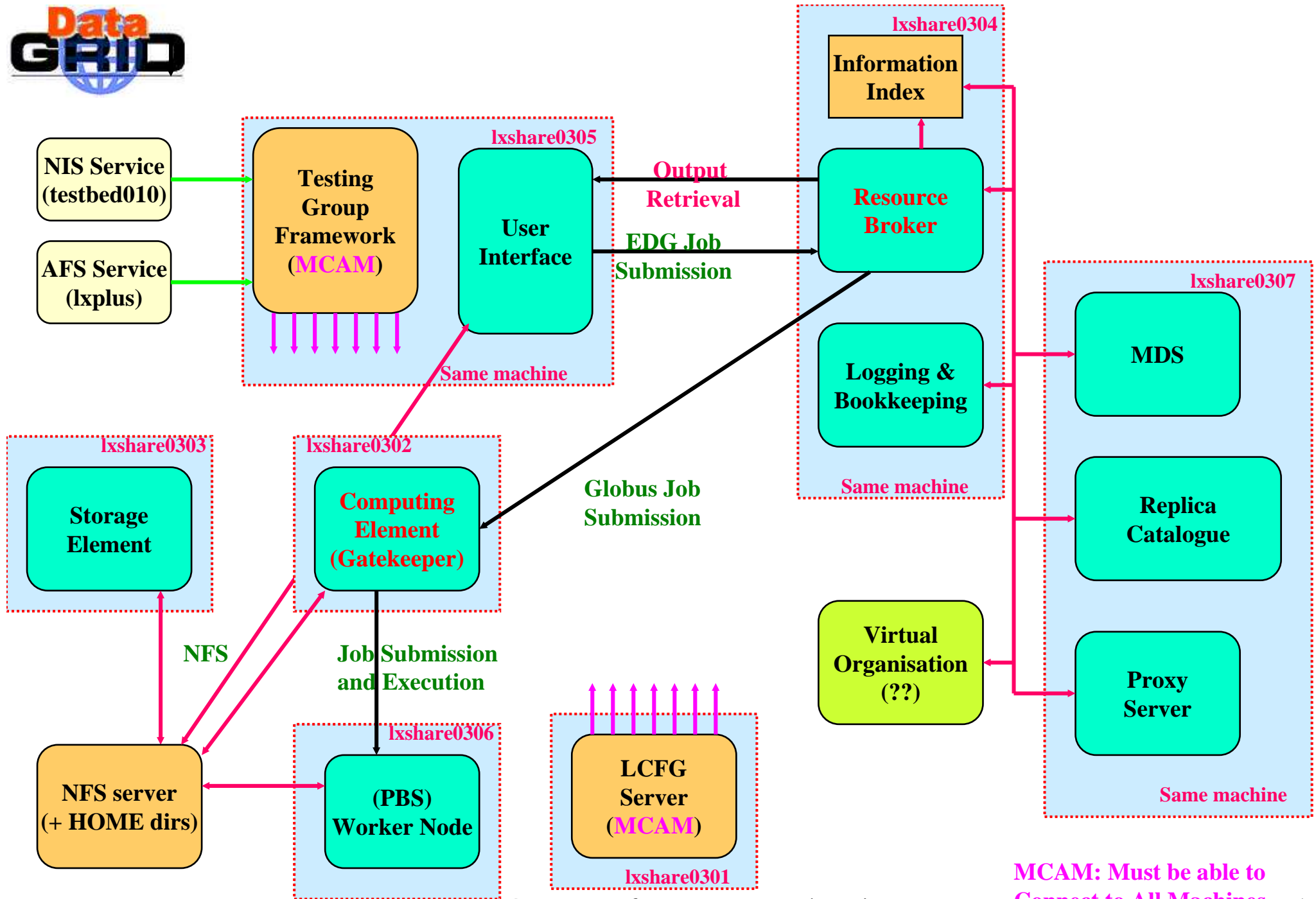
■ Terminology

- **level 1** : Install & Config checking
- **level 2** : Functionalities (unit testing)
- **level 3** : Stress testing (**and global testing**)
- Testsuites used during Steps #1 & #2 will run across all these levels



The 1st (Perfect) Certification Testbed

- A set of 6 (or 7) machines, running RH 6.2
 - installed thru an LCFG server at CERN: lxshare0301
 - CERN local mini-grid cluster, on a private LAN. [Name ?](#)
 - CE, SE, RB, UI, WN, (Proxy server + RC + MDS server): lxshare0302-7
 - don't know yet where the VO server component should be hosted
- These machines should be accessible thru 2 types of gateways
 - lxplus for HEP people able to be granted an AFS standard CERN login
 - a non-AFS front-end accessible from outside CERN for other people
 - testbed010
- Outside servers will provide additional services to this cluster
 - one NFS server for file storage filesystem, for the SE (lxshare073d ?)
 - one NFS server for the home directories filesystem (lxshare072d)
 - testbed010 will be the YP server for this mini-cluster





Comments on the Cert TB (2)

- A similar version of this mini-grid will later be made available also under RH 7.2
 - the LCFG server could be shared with the RH 6.2 one
 - same thing for the NFS and YP servers
- The 3rd stage of the evolution must further also evolve to allow for a fully realistic TB:
 - going from 1 RB + 1 (MDS/PS/RC) + 1 UI + **3 (CE/SE/2WNs) all at CERN**
 - to something like
 - **1 (CE/SE/2WNs) at CNAF**
 - 1 RB + 1 (MDS/PS/RC) + 1 UI + **1 (CE/SE/2WNs) at CERN**
 - **1 (CE/SE/2WNs) at CCIN2P3**
 - **1 (CE/SE/2WNs) at ???**
 - ...
 - this to allow for more cross-testing over realistic network conditions



But nothing is perfect ...

- Given the current conditions for Cert. TB availability (no blame !)
 - some components are very touchy to install (RC, MDS ...)
 - some are also very shaky to run (MDS, ...)
- Given the hurry for TSTG to validate the technical tools
 - we want to test our choices on a preliminary version of the mini-cluster
- We will probably have to shrink the mini-grid configuration available at startup. I selected the following priorities:
 - will install first an EDG 1.2.2 cluster
 - which will run only CE, UI, WN, RB, InformationIndex (no MDS machine)
 - this will not allow for any SE or RC testing, obviously
 - BUT we will be mainly testing **OUR OWN STUFF** only
- Hope everyone agree about this first step



Strategy for testing with a shaky setup

- If, during a global test running a job storm on the RB, we have to cope with some unstable component, for the early stages of our own testing (lets assume a shaky daemon on the CE), here is what I think to propose, from inside of an automated script launched from [Ixshare0305](#):
 - send a first sanity check towards the CE, using a script
 - if an irresponsive or dead gatekeeper is detected, (shut the daemon down), then restart it, using another script
 - use the same scenario to check the daemons on the RB as well, thru scripts
 - at last, launch the job storm, using the test script
 - and check that everything is still alive
 - or restore to normal conditions, if needed
 - on both the CE and the RB
 - using the same couple of scripts as above



Technical strategy for remote testing

- Yes, but: how to run all this from scripts ?
 - using both ssh and sudo in NOPASSWD mode
 - this was tested between lxshare0305 (UI) and lxshare0302 (CE)
 - it works, and in addition you get the output directly back into your script
- We will have to:
 - identify in every script every command requiring root access
 - wrap it appropriately to give proper access thru the sudo setup
- This is not easy, and a little bit heavy, but, at least, this allows **w/o having to log on or to provide any pwd** :
 - to automate all the tests
 - to shut down and/or restart any daemon or process at will when required for our tests (many M/W daemons are running under root)
 - restore normal conditions on any component at the end of every test wave
 - shutdown (for reboot) a machine, at will



Fallback from previous choices

- But I don't know yet how to achieve this when testing also remote components
 - EX: testing CEs at CNAF or CCIN2P3 from CERN
 - should we use a special account in this case ?
 - is RPM packaging of the sudo configuration file for this special account a good and safe solution ?



Current status of TSTG efforts

- A lot of stuff was uploaded in the `edg-tests` tree of the central CVS repository
 - the framework
 - a GUI to be used for launching by hand individual tests
 - the Install & Config test suites
 - level 1: sanity checks verifying that the install steps lead to a coherent state
 - the security checks
 - some functionality test suites for the RC (+SE)
 - level 2: unit testing to check that the tools are providing the expected services
 - the tools to display the results thru an HTML table
 - the entry of several test suites (a merge effort about job storms driven by Cal) is **pending**
- The main effort currently is to implement the main script (what is called the daily-check testsuite)