Subject: **Use Cases for WP4 discussion**

Author: **WP4/ German Cancio**
Partner:

Diffusion: **WP4 Internal**
Information:

1. USE CASES

1.1. GRIDSCHEDULER TO SUBMIT A GRID USER JOB TO A FABRIC (`"/BIN/LS /ETC"`)

- o Published available farm resources
- o Authentication
- o Authorization
    - o Available resources
    - o Quotas
- o Local credentials
- o JDL translation
- o Job handle to GRID scheduler
- o Queueing
- o Scheduling on CPU node
- o Copying script on CPU node
- o Execution on CPU node
- o Job status information
- o Job control operations by user (kill,…)
- o Output file generation,
- o Resource de-allocation on job termination
- o Accounting

1.2. OPERATOR TO REQUEST THE UPGRADE OF A TRIVIAL PACKAGE (IE. GZIP) ON ALL NODES ON SERVICE A

See the use-case described in the 'Discussion-WP4-part2 document'.

1.3. OPERATOR TO REQUEST THE UPGRADE OF A SYSTEM PACKAGE (IE. GLIBC) ON ALL NODES ON SERVICE A

See the use-case described in the 'Discussion-WP4-part2 document'.

1.4. OPERATOR TO REQUEST THE IP ADDRESS CHANGE OF A SERVER (IE. NFS OR ORACLE) AT 3AM IN THE MORNING

See the use-case described in the 'Discussion-WP4-part2 document'. Here some additional considerations:

- o Operator User Interface (GUI/CLI) – how to enter these atomic operations and their inter-dependencies
- o Scheduling of intervention time (*not before* 3AM)
- o Validation of operator's request *at submission time*
    - o authentication
    - o authorization
    - o valid IP address
    - o valid server
    - o valid timeslot
    - o (…)
- o Validation of operator's request at *execution time*
- o notification of success/failure of all/part of "atomic operations"
    - o to the operator
    - o to the monitoring system

1.5. PURGING LOG FILES (OR /TMP) ON A NODE

- o files in use
- o how to schedule
- o how to (eventually) detect an urgent cleanup is necessary

1.6. VERIFYING INTEGRITY OF CONFIGURATION FILES (IE. /ETC/SENDMAIL.CF OR /ETC/PASSWD) ON A NODE

- o how to schedule periodic checks
- o how and what component(s) to perform these checks
- o how to schedule "repair" if integrity is broken

1.7. SERVER CRASHES (IE. NFS OR ORACLE SERVER)

- o client/server interactions – how to handle dependent clients

1.8. INSTALL AND INTEGRATE A NEW NODE TO A FARM

- o who and how declare existence in cf DB (?) [if at all]
- o Installation server updates
- o Bootstrap agent on node

- o (eventually) add to RMS queues
- o installation: role of FT and Monitoring
- o security (Gridification!)

2. ADDITIONAL USE-CASES

There are additional use-cases, which we should think of. We should also interview system administrators for knowing what is missing here.

- o Fault Tolerance
- o User Management:
    - o Adding/Removing a Grid user and granting access to a set of resources
    - o Adding/Removing a local user " " " "
- o Job queue/resource management:
    - o Creating/modifying job queues
    - o Adding/removing a node from a queue
    - o Adaptative/automated resource management (eg. Nodes vs. queues)
    - o Resource reservations
- o Monitoring:
    - o QoS, Load/Performance,
    - o Faults,
    - o Auditing
    - o Application (ie. user job) monitoring
- o Notifications
    - o Changes in the system
    - o Alarms
- o Inventory
    - o CDB information (HW available, installed SW)
    - o Monitoring information for inventory
- o Administrator intervention
    - o (exceptional) direct intervention on nodes (ie run a admin-defined script on a set of nodes)