gLite Restrucuring Plan

John White, Helsinki Institute of Physics.
EGEE JRA1 Deputy Middleware Manager.

www.eu-egee.org

Information Society

Contents

- Background.
- Recommended Steps (the plan).
  - Boundary Conditions.
  - Milestones.
- Steps taken.
- Future.

Contents

- EGEE All-Activity, Bologna, January 16 and 17, 2007.
- Recognized the problems occurring with the evolution of the gLite distribution.
- Current status of gLite distribution doesn't allow for:
  - Easy inclusion of further functionality;
  - Porting to new platforms.
- The need for structural changes in the gLite distribution was identified.
  - In particular disentangling of dependencies,.
- Make a radical examination of the code base with a view to:
  - Removing unnecessary dependencies.
  - Cleaning up sections of the code that cause porting (and build) difficulties.
- Plan to be put to the PMB for endorsement.

Boundary Conditions

- The execution of the plan should be time-limited.
  - End of August 2007 planned.
  - In general, a 6-month task envisioned.
- no existing functionality must be affected.
- existing applications running on EGEE are not effected.
- major users of the infrastructure get all the required suppor t to continue their usage.
- effected activities need to fulfill their reporting duties to EGEE.
  - in particular the periodic report.
  - their duties related to the EU review in May,.
  - as well as providing input for the EGEE- III proposal.
- The following tasks cannot be delayed by executing the plan.

"Protected" Tasks

- Security patches.
  - Evident. Prioritized by GVSG/EMT.
- Blocking bugs.
  - What is really blocking to be assessed by EMT.
- Move to SL4 on worker nodes and user interface
  - JRA1 takes this task as highest priority and manages the process
  - Mostly done now. Check run-time.
- Move to ETICS
  - JRA1 takes this task.
  - As above.
- SRMv2.2 support in DPM, lcg-utils, GFAL, FTS.
  - .
- Update to GLUE v1.3, required for SRMv2.2 support.
  - .
- Group-based accounting (APEL/DGAS)
  - Ongoing.

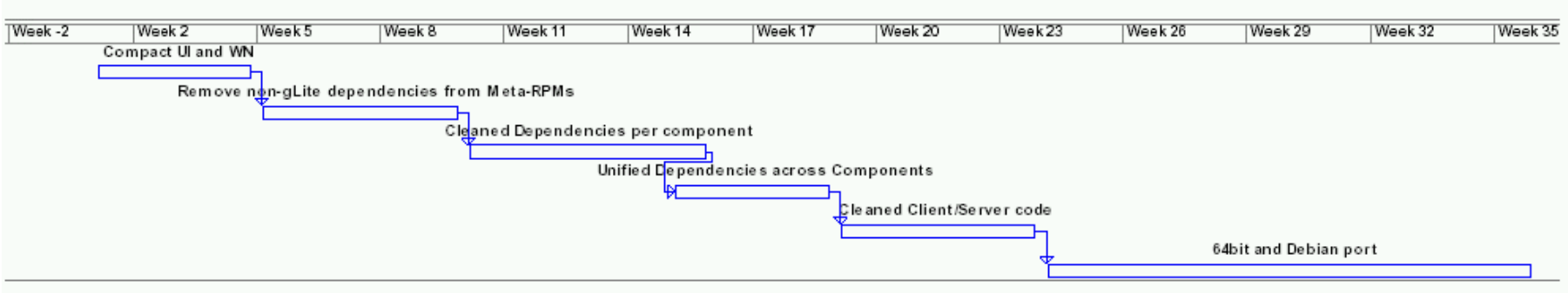
"Protected" Tasks Continued

- Some FTS improvements.
  - These need to be defined precisely.
- Support of Secondary Groups on DPM & LFC.
  - Resolution of the rfio problems?
- Encrypted data and catalogue access (GFAL extension and LFC work)
  - STILL important for Biomed.
- Deployment of Job Priorities Work Group.
  - Implement their recommendations (and config. support).
- Stabilization and scalability of WMS and LB
  - Ongoing, shifted to INFN.
- Stabilization and scalability of the gLite- CE
  - Work starts now.
- VOMS Admin v2 supporting generic attributes
  - Ongoing.

The Plan

- Starting point:
  - gLite 3.1.
  - VDT 1.6.
  - GT4 pre-WS (as in VDT 1.6).
  - Condor 6.8.4 (6.6.6 for legacy services).
- At the end of each plan stage a working system has to be available.
- Timelines given are net work- weeks without interruption and other tasks..
- Acknowleged: Given the long list of non deferrable changes...
  - The teams are likely to be required to task switch frequently.
  - Will have a significant negative impact on their efficiency..
- SA3 has to clean up the configuration and testing systems..

Milestones



1. Compact UI and WN.
   - 4 weeks (including testing)
2. Remove non- gLite dependencies from Meta- RPMs
   - 5 weeks (including testing)
3. Reduced and disentangled dependencies per component
   - 6 weeks
4. Unified Dependencies across Component s
   - 4 weeks (1 week can be in parallel with item 3 above)
5. Cleaned Client /Server code
   - 5 weeks (including testing)

Milestone 1

- Compact UI and WN.
- Overall Goal: Remove obsolete components from the repository.
  1. Clean up the repository by removing obsolete components.
  2. Start with the UI and WN to build a list of the release components.
  3. Remove dependencies related components on long on (WN,UI).
- Responsibles: SA3.

Milestone 2

- Remove non- gLite dependencies from Meta- RPMs.
- Overall Goal: Gain a better understanding of the dependencies of individual components.
  1. Publish the list of Meta RPM dependencies.
  2. Select external dependencies and add them to the build configuration of their components (reformulate) .
  3. Provide test UI/WN (probably based on XEN) for developers for testing their dependencies.
  4. Testing reduced dependencies.
- Responsibles: JRA1/SA3.

Milestone 3

- Reduced and disentangled dependencies per component.
- Overall Goal: Dependencies per component cut down to a minimum.
  1. Challenge dependencies.
  2. Understand what can be removed and what can be handled by a few extra lines of code.
  3. Implement those changes suggested by 2) that can be implemented within 1 week (according to the developers).
- Responsibles: Component Responsibles, Dependency Challenge Team (JRA1/SA1/SA3).

Milestone 4

- Unified Dependencies across Components.
- Overall Goal: Common set of external dependencies.
  1. Compare the dependencies of different components.
  2. Unify dependencies and identify where a component can go static .
- Responsibles: Component Responsibles, SA1/3.

- Cleaned Client /Server code.
- Overall Goal: cleaner separation between client and server code.
  - The code use on the clients, servers, and in so- called common parts will be cleaned, aiming at cleaner client / server separation..
  - For each component an assessment has to be made including the list of changes required and the estimate of work.
- Responsibles: Component Responsibles, Dependency Challenge Team.

Resulting Milestone

- After Milestone 5 has been achieved, the port of the WN and UI to 64bit (x86) and Debian will be addressed. This will be used as an assessment of the success of the restructuring exercise..
- Responsibles: Project.

Configuration/Testing Improvements

1. Configuration management
   - Split YAIM (component by component). Underway. 3 weeks
   - Single layer configuration.Underway. 2 weeks for UI/WN
   - Multi platform YAIM for UI/WN. 2 weeks. 3 weeks for Debian.
2. Optimized Testbed operation and test process.
   - Analysis/Design 2 weeks. Implementation 3 weeks.
3. Database access for Savannah to improve reporting and documentation.
   - 2 weeks.
4. Responsibles: Component Responsibles, SA3.

Current Situation

1. Configuration management
   - Split YAIM (component by component). Underway. 3 weeks
   - Single layer configuration.Underway. 2 weeks for UI/WN
   - Multi platform YAIM for UI/WN. 2 weeks. 3 weeks for Debian.
2. Optimized Testbed operation and test process.
   - Analysis/Design 2 weeks. Implementation 3 weeks.
3. Database access for Savannah to improve reporting and documentation.
   - 2 weeks.
4. Responsibles: Component Responsibles, SA3.