

A gLite SDK

Author

Location

- **Propose to introduce the concept of the gLite SDK.**
- **The concept is already in the proposed changes to the Description of Work**
- **This would be a versioned list of components which any of the following are true:**
 - The package is required at buildtime by more than one gLite component
 - It should be non-trivial (no need to mention glibc)
 - If it is needed by only one gLite component, it should be 'owned' by the responsible team and does not need to be in the SDK.
 - The package is part of the external interface of a component/service

- **Gives us a precise list of components which are under particular constraints:**
 - Cannot change their interfaces through a gLite release
 - Components in gLite but not on the SDK list can vary more freely
 - Must be properly documented
 - Must be available as rpms
 - Will facilitate building from source rpms
- **Allows planning ahead**
 - We could start on gLite 3.3 (or 4.0...) NOW
 - We take the SDK list and ask what has to change
 - We could then branch code, set up the build and work in parallel
- **Helps inter-product-team coordination**
- **Decouples our build environment from its ETICS implementation (allowing others to build)**

Origin	
OS	Swig, boost
jpackage	Axis, bouncycastle, jdk,
Externals	Libtar, c-ares, classads, gpt, vdt_globus_essentials
gLite	Voms-api-c, gfal, wms-proxy-api, jobid...
BUT NOT	eg wms-common (internal to a sub-project)

- We are mixing a few concepts
 - Things glite components depend upon
 - Internal glite dependencies between sub-projects
 - Externals
 - Things applications will depend upon
- Imagine the SDK as a separate project on which each service depends