Enabling Grids for E-science in Europe

www.eu-egee.org

JRA1 all-hands-meeting, RAL 28.06.2004

build & integration system
Framework & configuration

Joachim Flammer
Integration Team

EGEE is a project funded by the European Union under contract IST-2003-508833

Contents

Enabling Grids for E-science in Europe

- The GLite puzzle
- gLite build system overview
- gLite Integration process
- Some important build system aspects
- Adding new subsystem / components
- External libraries
- Configuration
- Documentation

The glite puzzle

Enabling Grids for E-science in Europe

Build system overview

Enabling Grids for E-science in Europe

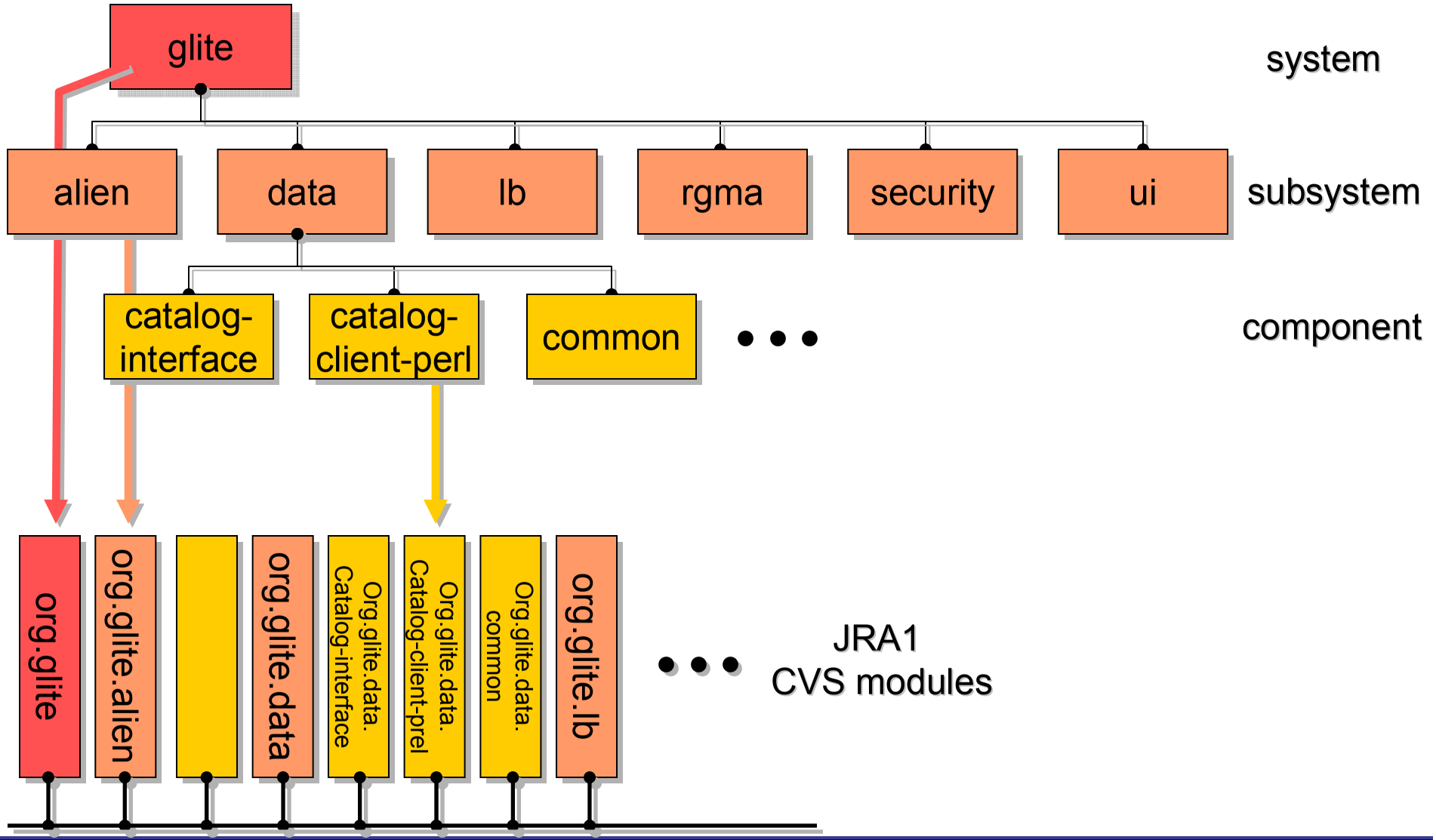
- Main ideas of glite build system
  - Ease development and the release of quality software
  - Common build system using ant
  - Three level hierarchy: Global – subsystem – component
  - Each component is individual module in CVS
  - Possibility to build on global/subsystem/component level
  - Uniform process as much as possible
    - Use common elements (targets, settings …) / move up common stuff
  - Allow user settings to easily adapt system
    - E.g. testing a new library
  - Provide src/bin tar ball
  - Produce installation pieces: rpm, dll (stay tuned for Roberts talk …)
  - Automatic build system (stay tuned for Marian's talk …)

Integration process

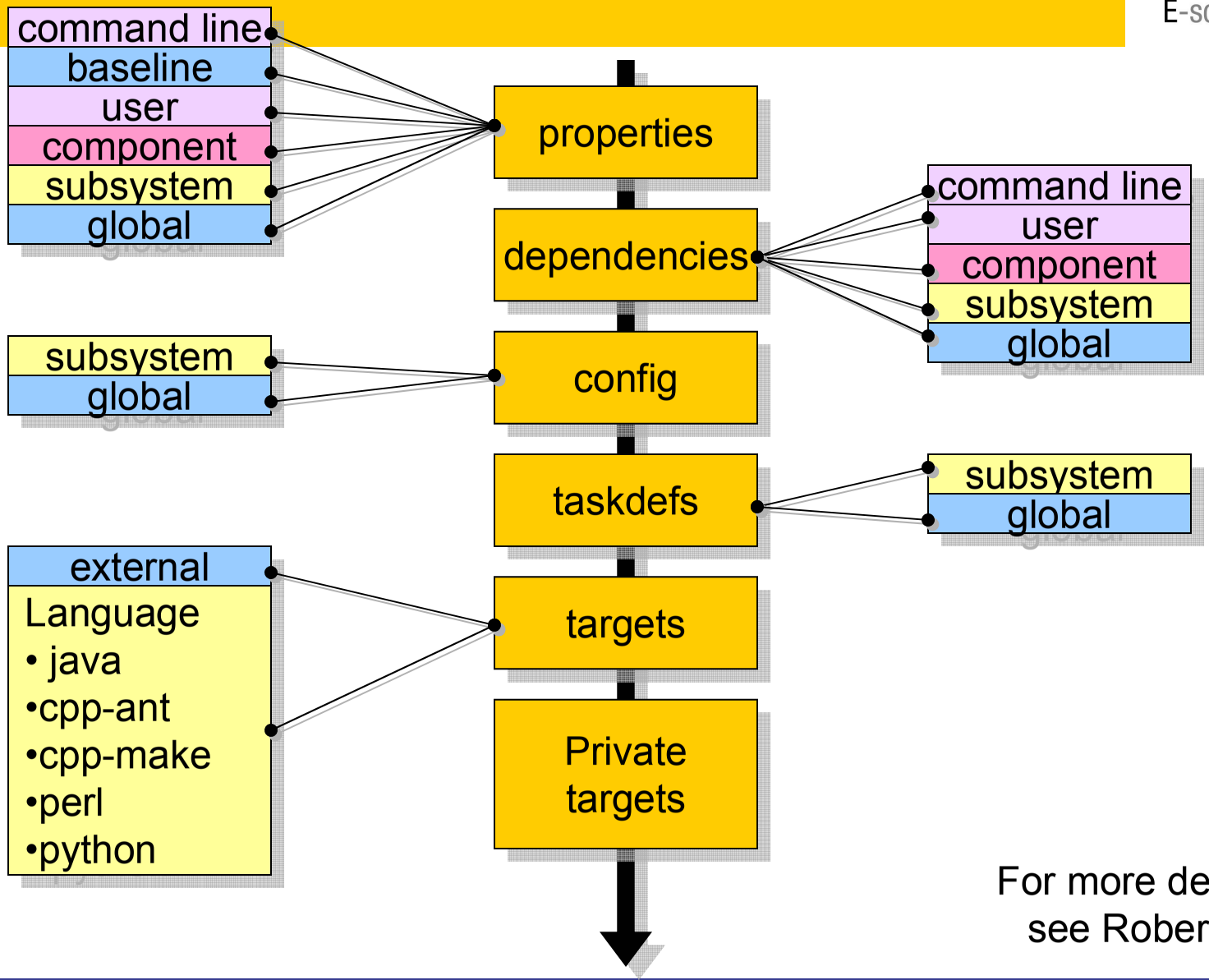
- Integration started
  - Different subsystems are putting code in
- Training of developers
  - So far only a small group of developers involved
  - Integration team offered some training for developers – only minor fraction of developers involved
  - Is there a requirement for more training at the different sites?
- Information flow
  - We try to announce every major change well in advance
  - Information about build system changes is not propagated fast and widely enough
  - How to make sure that everybody knows what changes are done to the build system

SUBYSTEM	# of components
ALIEN	6
DATA	13
LB	5
RGMA	5
SECURITY	11
UI	2
WMS	15
7 subsystems	57 components

Build structure



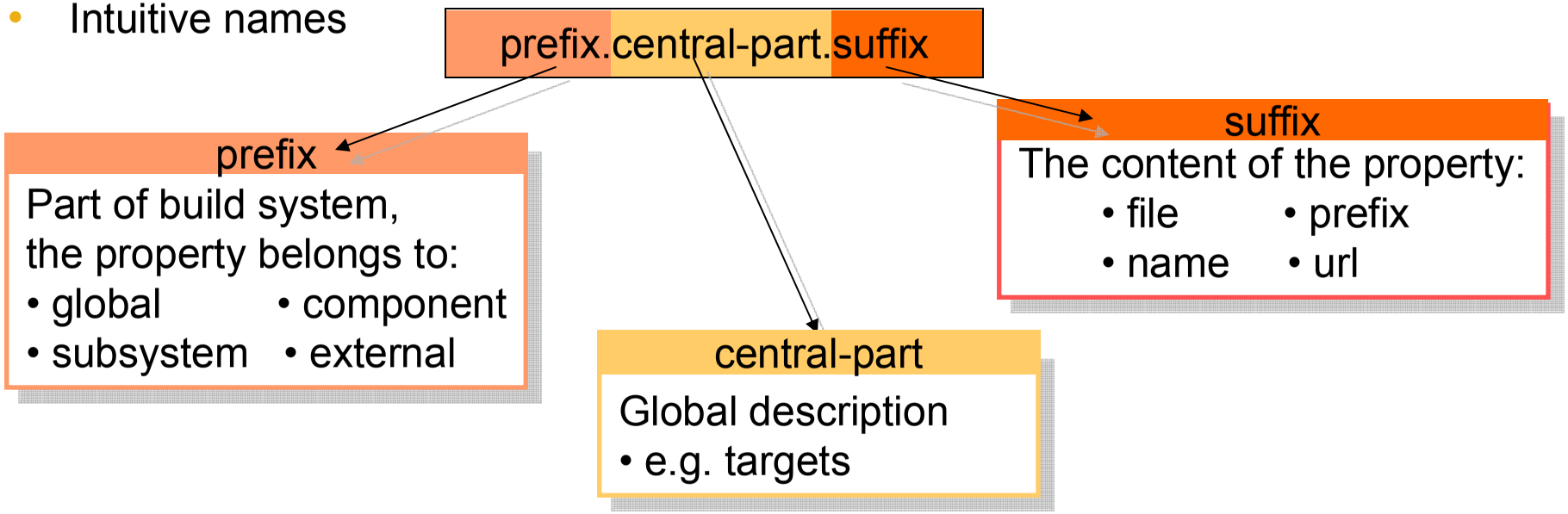
A typical build file



For more details about
see Robert's talk …

Properties in build system

- Use properties whenever possible in build files
  - Flexibility
  - Common approach
  - Maintainability
- Properties in build system should follow common naming convention
- Intuitive names

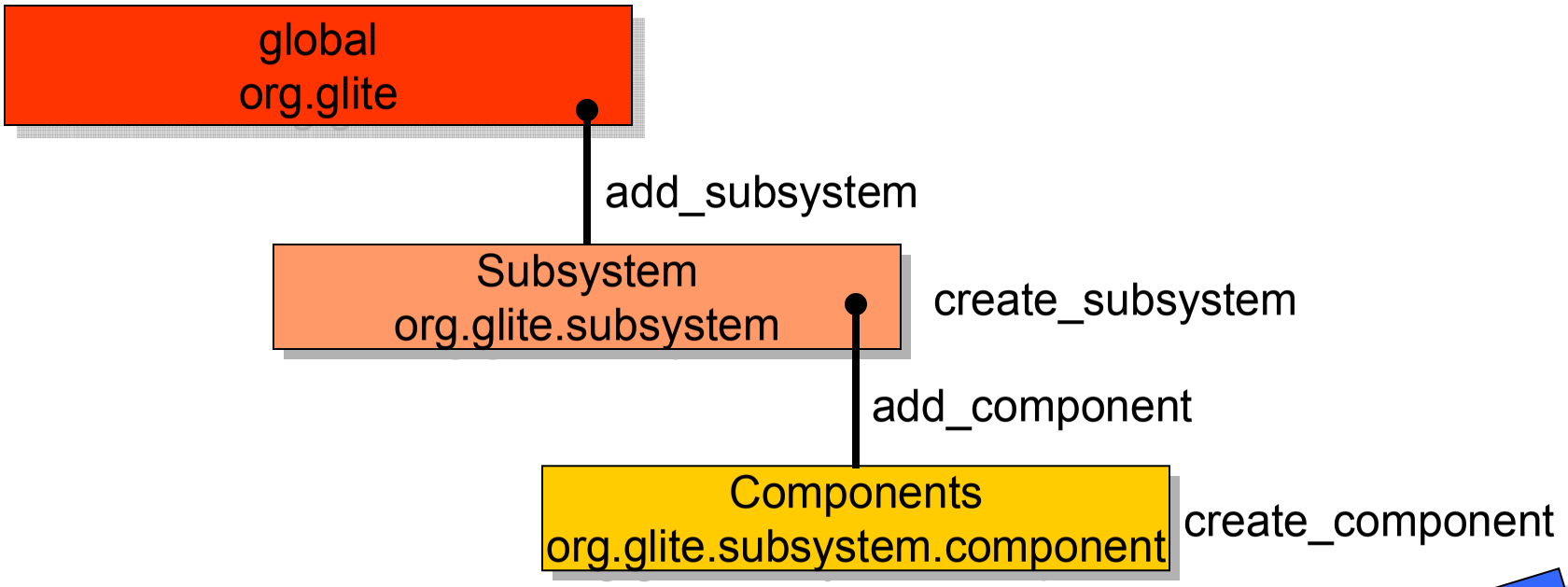


- You can get a list of the properties via

```
ant print.properties                          to print all properties
ant print.properties –Dproperty.regex="ext.*"  to print regex selected properties
```

Adding a new component/subsystem

- Scripts to automatic produce new components and subsystems
  - CVS module org.integration.administration



- Adapt subsystem/components
  - e.g. add language targets, dependencies
- Put the module to CVS

Description on web page

External dependencies

Enabling Grids for E-science in Europe

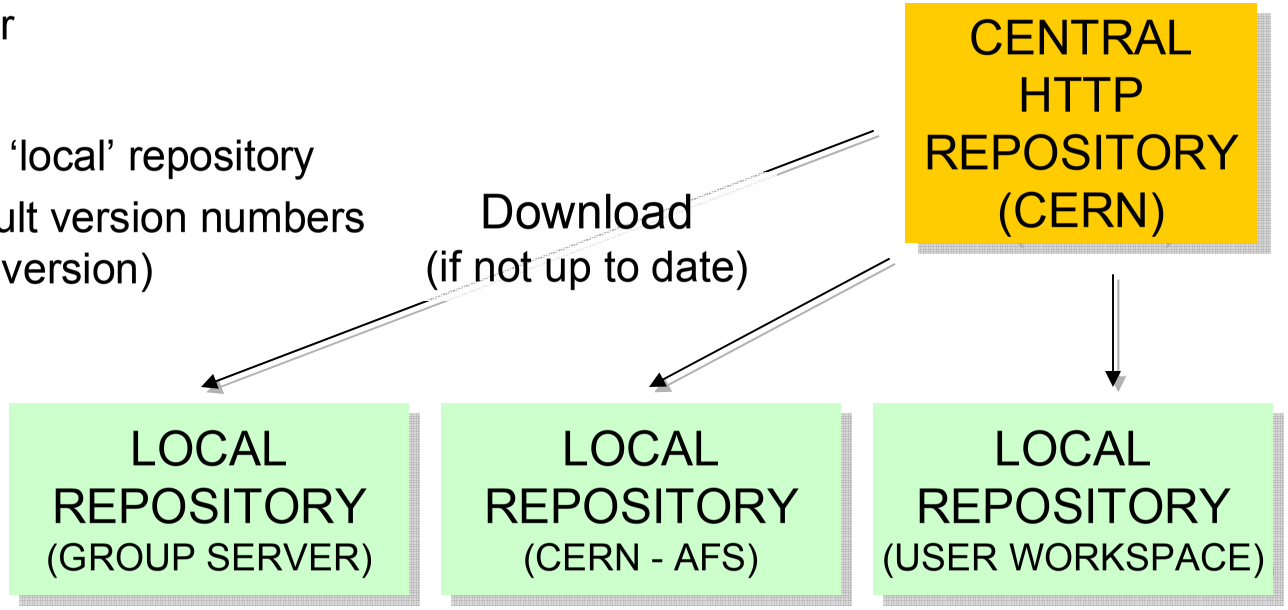
- Agree on one set of official external dependencies only
  - new version of external libraries have to agreed via the Change Control Board
  - Update to newest version now if necessary/possible …
- Common repository that contains all external dependencies
  - directory tree in repository to distinguish/support several versions and platforms
    - 🗁 package_name
      - 🗁 package_version
        - 🗁 platform
- Each package contains
  - Necessary parts of external component in original vendor structure
    - We will omit the included secondary components
  - one tarball with only the necessary parts but otherwise untouched package
- External packages are defined in org.glite/project/dependencies.properties file
  - Name
  - Description
  - Download place …

We need your input !!!

External dependencies II

Building a module:

- External dependencies are copied to local repository
  - make sure that components use same version
  - allow standalone work (e.g. on a laptop without network connection)
- Possible places
  - Workspace (default): workspace/repository/…
  - Afs (CERN)
  - Common group server
- User can
  - (a) define place of his 'local' repository
  - (b) overwrite the default version numbers (e.g. to test a new version)



External dependencies - overview

Enabling Grids for E-science in Europe

	GLITE	ALIEN	R-GMA	DM	WMS	Integration
ant-contrib	0.6	UNDER INVESTIGATION – to be included by CPAN				0.6
axis	1.1			1.1		
boost	1.29				1.29	
bouncy castle	1.23		1.19.2	1.23		
cgsi-soap	2.6-1.1.3			2.6-1.1.3		
checkstyle	3.3					3.3
condor classADs	0.9.5				0.9.5	
condorG	6.5.3					
cpptasks	1.0b2					1.0b2
Cppunit	1.9.14				1.9.14	
EDG Java Security	1.5.10		1.5.10			
EDG Java Security Client	1.5.10		1.5.10			
expat	1.95.2					
gfal	1.3.2			1.3.2 ?		
gsi libs	2.4			2.4		
gSoap	2.6			2.6		
jas	1.0.0-1		1.0.0-1			
java	1.4.2_04		1.4.1_01			
java-cog	1.1			1.1		

External dependencies - overview

Enabling Grids for E-science in Europe

	GLITE	ALIEN	R-GMA	DM	WMS	Integration
jalopy	1.0b10	UNDER INVESTIGATION – to be included by CPAN				1.0b10
Junit	3.8.1			3.8.1		3.8.1
jxUtil	1.0.1-1		1.0.1-1			
lcg srm	1.2.2			1.2.2		
log4j	1.2.8		1.2.6-1	1.2.8		
myproxy	1.1.8					
mySQL	4.0.18		4.0.13	4.0.18		
mySQL java driver	2.0.14-1		2.0.14-1			
openssl	0.9.7			0.9.7		
perl	5.8		5.6 / 5.8			
prevayler	1.3.3-2		1.3.3-2			
python	2.3			2.2 / 2.3		
swig	1.3.19-1		1.3.19-1			
tomcat	5.0.25			5.0.25		
trio	1.4					
xerces-C	1.7.0-3		1.7.0-3			
xerces-J	1.4.4-12		1.4.4-12			
xml commons	1.0.0.b2.1jpp		1.0-0.b2.1jpp			
xml Commons Apis	1.0.0.b2.1jpp		1.0-0.b2.1jpp			

Configuration – Where do we stand ?

Enabling Grids for E-science in Europe

- **What we learned from the past**
  - Configuration is a key functional/operational point
  - System must be easily configurable for inexperienced users
  - SA1 requirements

- **Configuration – status quo**
  - Configuration is distributed over several places and files
  - Each system uses their own configuration system
    - Some values are repeated
    - Some values might be obsolete
    - Very heterogeneous systems
      - Attribute-value pair (R-GMA, DM)
      - Classads (WMS)
      - LDAP (Alien)
      - …
    - Specialized Configuration scripts …

Configuration – Important aspects

Enabling Grids for E-science in Europe

- Use a common structure for glite system
  - common places in file system (e.g. /opt/glite/etc/…)
  - common file type (plain text file, xml, structured file …)
  - common used variables (e.g. no replication of hostname …)
- Guide user in selecting the correct values
  - Default values for configuration wherever possible
  - Example values if feasible
  - Good documentation for each configuration value
  - Configuration tool
  - Human readable configuration files ⇨ be able to work without tools
- Mark/structure configuration elements according their “to be changed” type
  - Values that must be changed (e.g. hostname)
  - Values that could be changed (e.g. the port number)
  - Values that should (normally) not be changed (e.g. the)
- Group/structure configuration elements according their system part
  - Globally used configuration values
  - Subsystem configuration values
  - Component configuration values
- Make system configurable
  - Internal structure of components …

Configuration – Important aspects II

Enabling Grids for E-science in Europe

- Local/remote configuration
  - Keep configuration locally as file
  - Have central configuration server
  - Combination of both
- Update of configurations
  - Update via configuration server?
  - Pulling model
  - Push model
- Different types of configuration different types of system
  - User configuration
  - Server configuration
  - Cluster configuration
- Retrieve Configuration information
  - by user
  - by administrator
  - by help center

Configuration – Getting to a common glite system

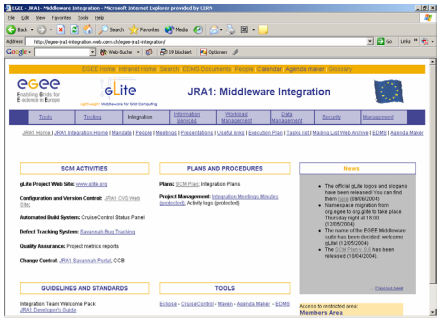
Enabling Grids for E-science in Europe

- Meeting of design team to discuss various ideas
- Investigate current configurations of the subsystems in more detail
  - What type of configuration ?
  - Try to structure configuration elements
  - Identify common elements
  - Understand reason for different configuration tools and their needs
  - Understand configuration needs
  - Start documenting configuration values, default values …
- Try to come up with a common structure

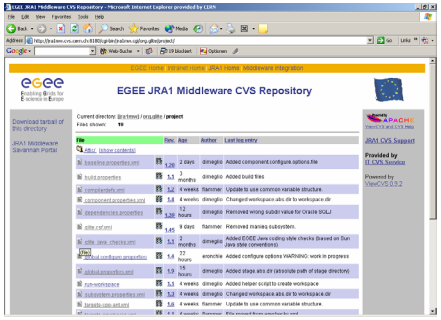
☺ **I will need your help !!!** ☺

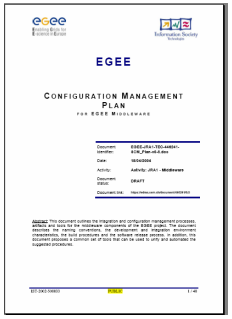
Documentation

- Learning more about the Integration
  - JRA1 Integration Team web pages
    - http://egee-jra1-integration.web.cern.ch/egee-jra1-integration/



  - JRA1 CVS web access pages
    - http://jra1mw.cvs.cern.ch:8180/





  - Software Configuration Management plan Version 1.0
    - https://edms.cern.ch/document/446241/1.0
  - Developers guide
    - https://edms.cern.ch/document/468700/0.1

Summary

Enabling Grids for E-science in Europe

- Integration process in build system
  - Major problems so far for different subsystems ?
  - More training necessary ?
  - Information flow ok ?
- Tools for integration in build system
  - Scripts for component/subsystem creation
  - Other scripts necessary?
- External libraries
  - Finalize common set of external libraries
- Configuration
  - Important aspects
  - Next steps …