



Enabling Grids for E-science

# Multi-Platform Experience in EGEE-III

***Author***

***Eamonn Kenny***

***Location***

***CESNET, Prague***

***Event***

***SA3/JRA1 All Hands Meeting***

[www.eu-egee.org](http://www.eu-egee.org)



- **Obtain a working, sustainable SL5 glite-WN as quickly as possible.**

## How?

- **Change some configurations and introducing bug reports into Savannah.**
- **Perform some local builds and confirm results using the NMI builds in CERN.**
- **Use per node builds.**

## Expected Issues:

- External Dependencies might need some changing (not expected to take much time)
- Base OS dependencies are newer and some features under SL4 could be deprecated, requiring reworking of components (could be time consuming)
- SA3 generate a few bug reports which will need some attention
- Bugs are fixed and further bugs are generated until all bugs are solved (this could be a time consuming process, because of the iteration cycle)
- gLite-WN goes to certification where it is tested and returns issues to developers that need fixing (not expected to be too long)

## Surprisingly:

- External Dependencies: VDT 1.10.1-1 was available in July, VDT 1.8.1 was available in September. Classads caused a few issues for the WMS.
- Base OS dependencies biggest problem was newer swig for LCG-DM
- SA3 generated a few bug reports which will need some attention (all defined by July 2008)
- Bugs are fixed and further bugs are generated. This wasn't the time consuming process we thought it would be. Gridsite biggest issue.
- gLite-WN went to certification and passed okay.

## Summary of Why SA3/JRA1 had Issues (Five Main Points)

- 1. External dependencies core to our infrastructure must be made available more quickly: VDT, dcache.**
- 2.  $N$  Base OS dependencies on  $M$  platforms is not very scalable when generating ETICS configurations.**
- 3. Grouping externals/OS dependencies together is a gLite build system legacy fudge!**
- 4. ETICS is not fully embraced as a build system. Waiting on the next iteration to solve multi-platform issues isn't always the answer.**
- 5. Client/server split is not clear enough. (more explanation to follow)**

# 1. External Dependencies

- **VDT globus can be built from source.**
- **VDT 1.8.1 and 1.10.1 build easily on all Linux platforms and Mac OS X but require some extra work for AIX and Solaris.**
- **gpt, gsiopenssh, myproxy (with VOMS support) and uberftp-client are easily ported.**
- **A file list + script can be used to extract all required vdt\_globus\_<packages> from the source build (as individual items – needs a little more work).**
- **dCache (dcap) is something we have no control over?**
- **Continue to take binaries from VDT but in parallel start to build from source?**

- **package.location + package.version should be enough to describe all base OS dependencies.**
- **ETICS 1.3.10-1 will have a fallback to build from source or binaries found in AFS.**
- **Decentralisation of services will never be possible unless we make this move.**
- **Versions increase with newness of OS, whilst external dependencies remain fixed.**
- **Version increase introduce deprecation issues, while externals don't suffer from this effect.**
  
- **Intermediate fix: Make a Base OS dependency an external dependency until deprecation issue are fixed?**
- **Effect of multiple versions of a package?**

### 3. Grouping External/Internal Dependencies

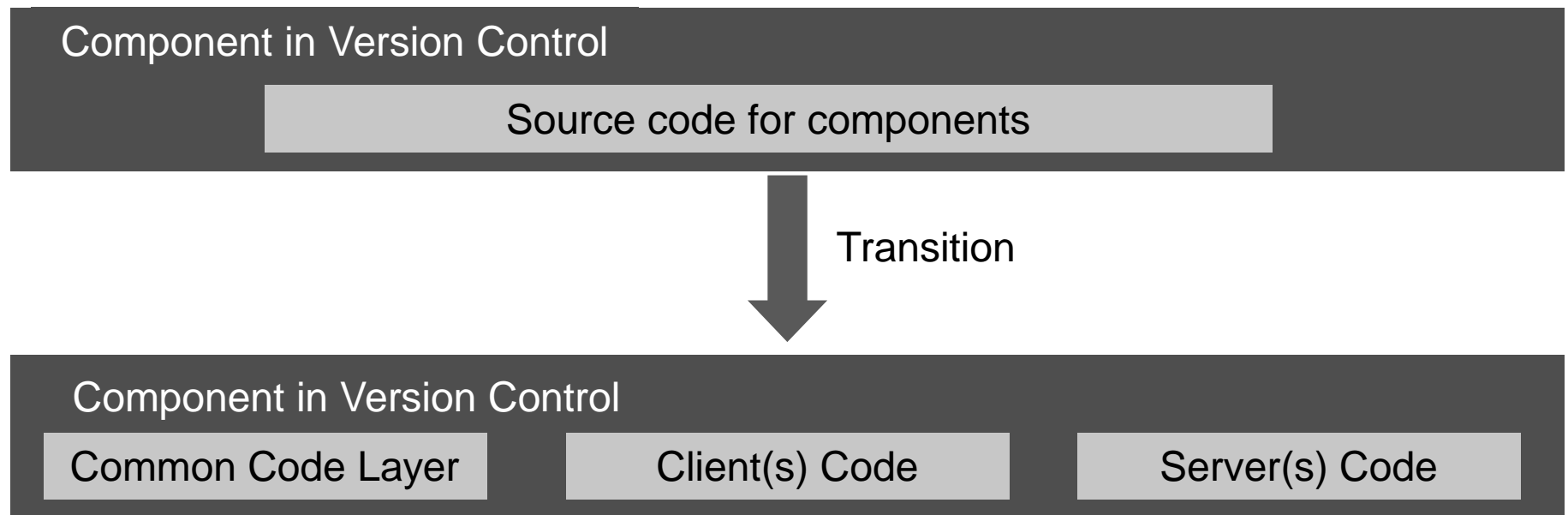
- **Large numbers of packages are still grouped together.**
- **Understanding is lost over time on the implications of such grouping. ETICS/SA3 have mistakenly packaged one RPM as a tarball.**
- **Would the use of `package.version` and `package.location` make the transition quicker?**
- **Would ETICS NMI build machines using Quattor or yum `groupinstall` help get packages into place on the different platforms?**



- **Where is the SL5 x86/x86\_64 build results that all developers are tracking?**
- **Locking and ease of interrogating ETICS reports is very problematic. Clearly represented build rules, error statements and hidden traces are required.**
- **ETICS current appears to do enough for any well organised code base.**
  
- **Can developers get by with how ETICS handles builds currently?**
- **Is locking the real issue or is platform independence the real issue?**

## 5. Client/Server Split: EGEE II

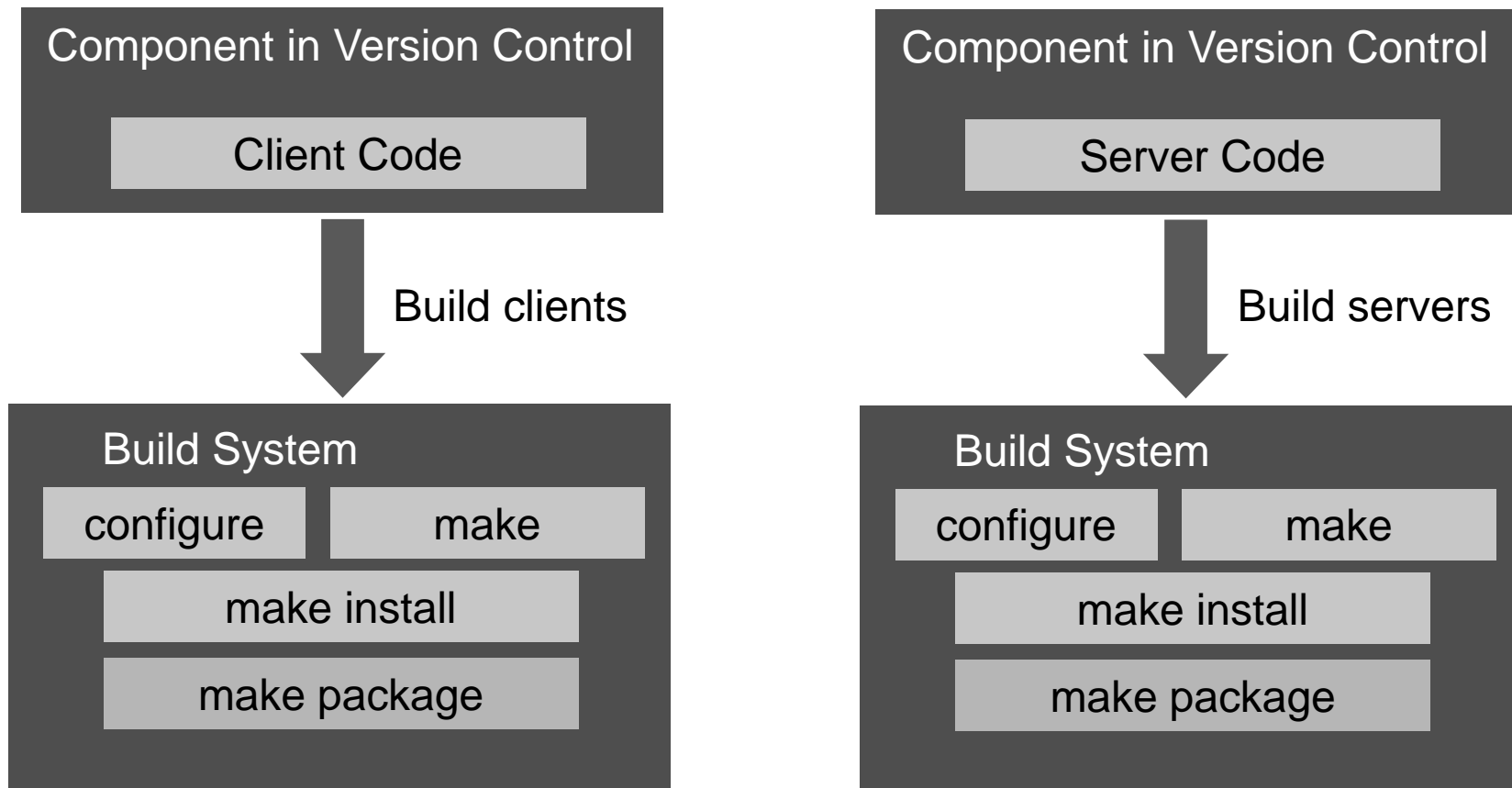
24. SA3. Build on the SA3 processes in EGEE-II to ensure multi-platform support is achievable with available resources. In particular, reinforce the client-server code separation to ease the expansion of client platform support.



- **Client/server split was a reviewer recommendation from EGEE II.**

# 5. Client/Server Split: Many codes, many builds, many packages

Total code separation



Total build separation

Optional

# Alternative: One code, one build, many packages

Component in Version Control

Source code for components

Build Everything

Build System

configure

make

make install

make package

Package Clients

Package Servers

Build System

make install






make package

Build System

make install

make package

# Version control and ETICS split

Number of Components	Build Steps	Packaging Steps	Overall Correctness	Examples
Many	Many	Many		WMS LB
One	Many	Many		dCache
One	One	Many		VDT VOMS
Many	One	One		LCG-DM
One	One	One		AMGA myproxy Gridsite

- **Faster integration of externals/internal dependencies is required.**
- **Client/server code split needs to be pushed more.**
- **ETICS needs to be embraced.**
- **Development and release branches partially solve the locking problem but create extra configurations.**
- **Platform independent build configurations reduces ETICS database content making transition to new platforms easier.**
- **Meta-package and associated provides need to be uniquely stored (e.g: problems arise in use of 2 VDT's with one VOMS registered).**