

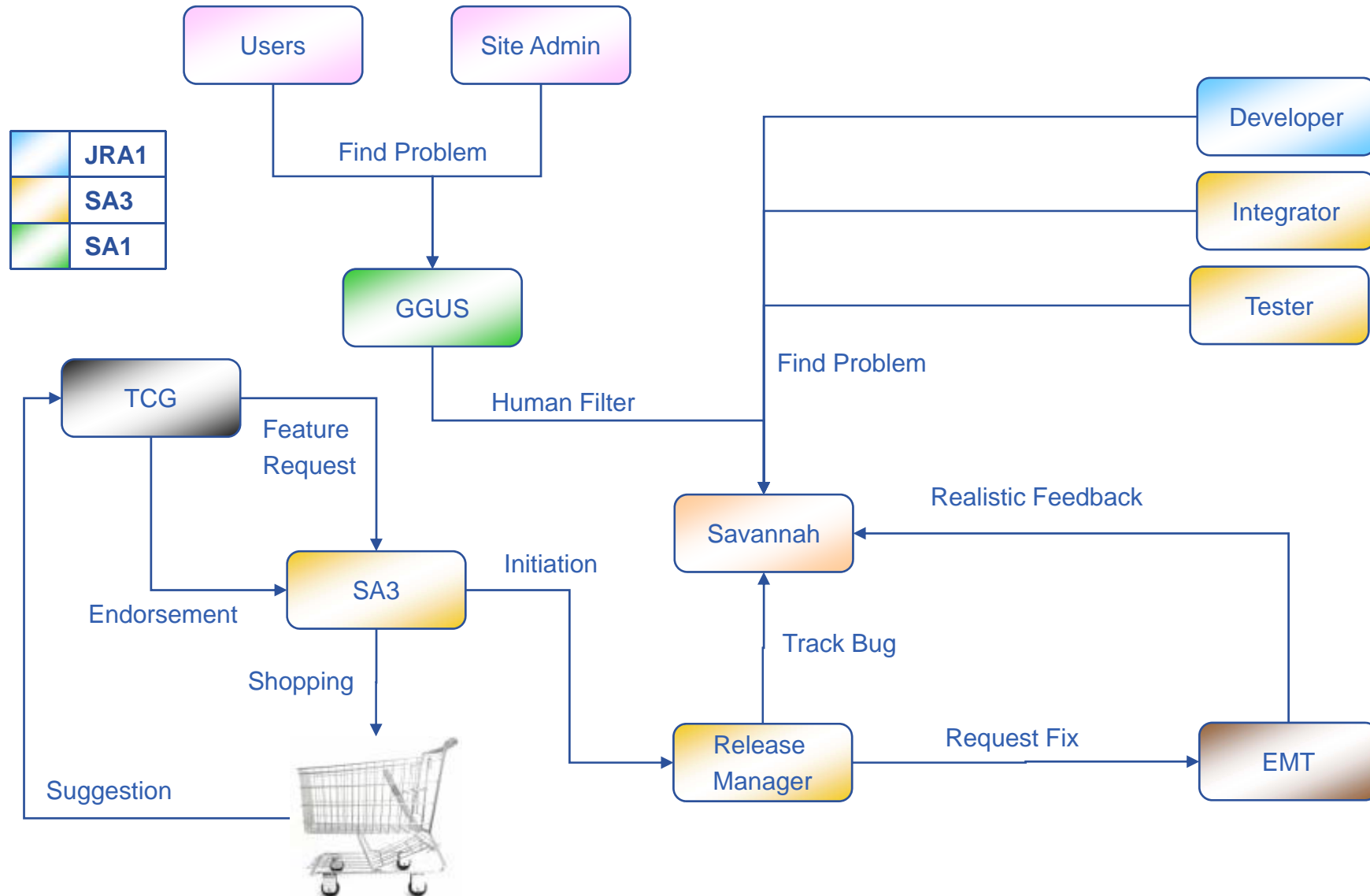
Integrating glite 3.1: an evolutionary approach

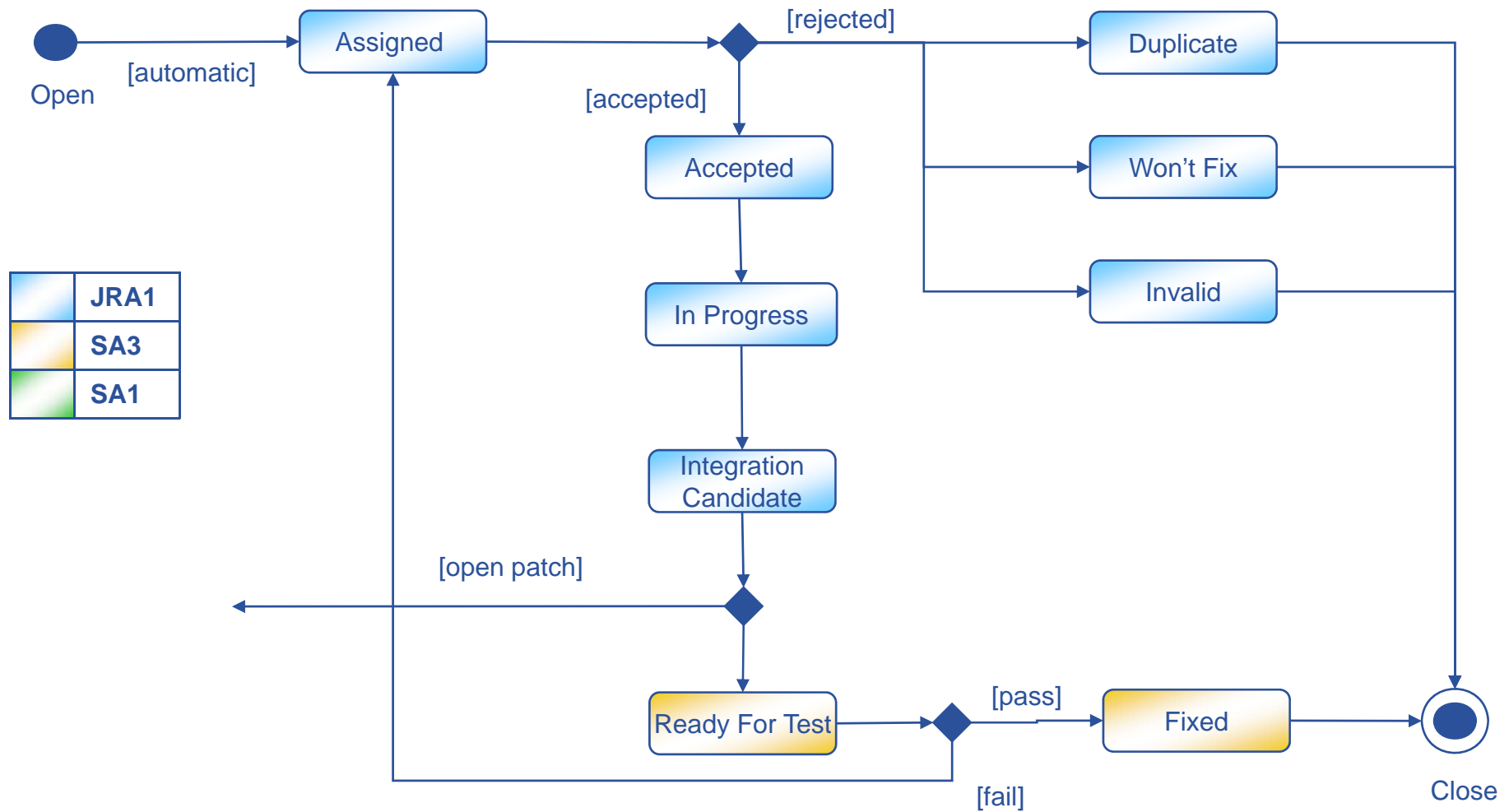
Author: Laurence Field (CERN)

- **General Observations**
- **The Software Process**
- **Building and ETICS**
- **Integration Procedure**
- **Integration Tools**

- **SA3 has more work since glite-3.0**
 - But the same number of people
- **Two extra dimensions**
 - Releases per node type
 - Multiple platforms
- **Need to do everything in parallel**
 - Automate everything!
 - Generate and archive information
 - What, where, when and the result
- **Main Problems**
 - Communication overhead
 - Information recording
 - Time for iteration cycles

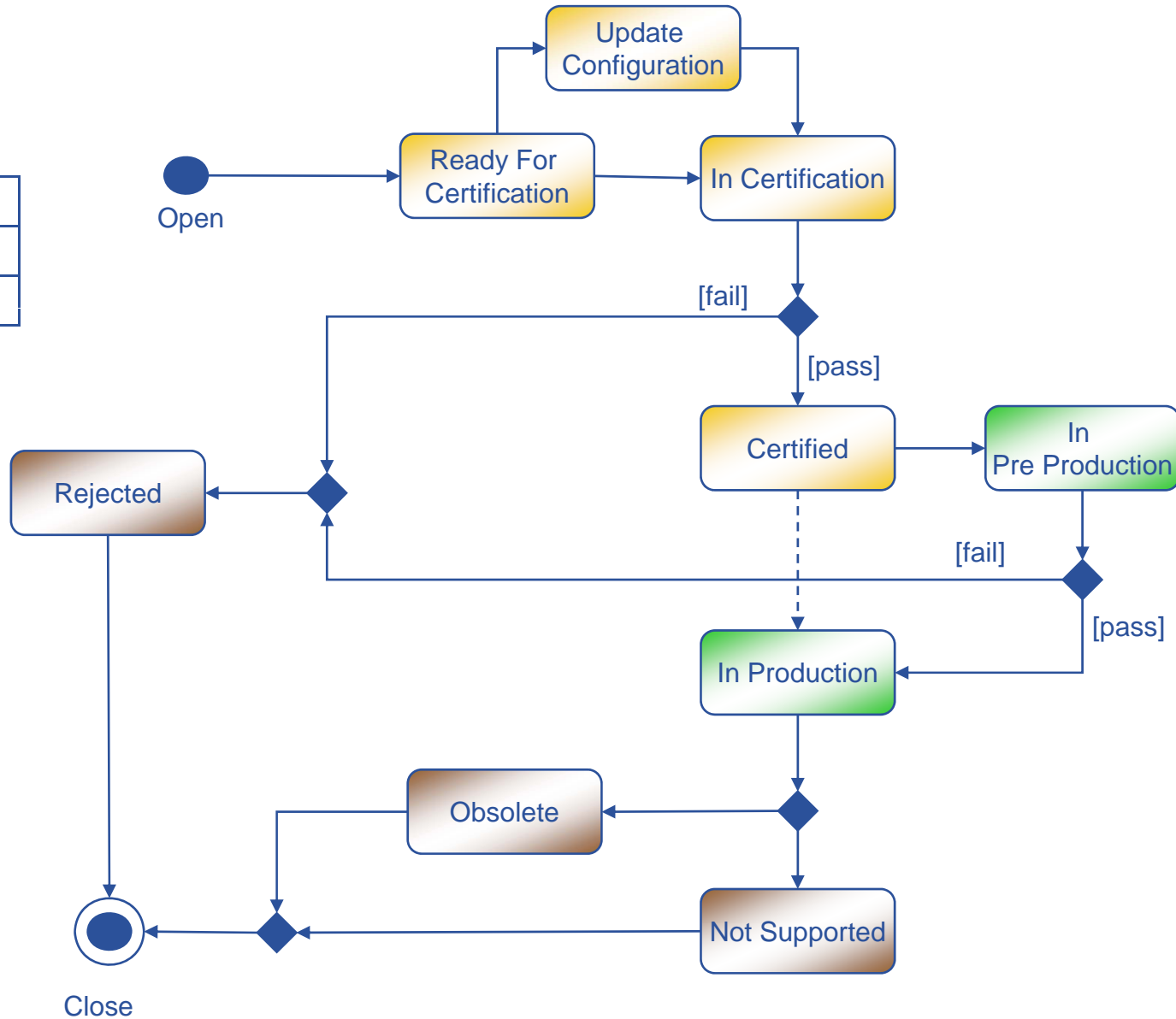
- **Documented May 2006**
 - Jointly by SA3, SA1 and JRA1
 - Official EGEE Deliverable
 - <http://edms.cern.ch/document/724371>
- **Defines how we should work**
 - Roles, Responsibilities and Interactions
 - The workflow between JRA1, SA3 and SA1
 - Defined jointly by representatives from each activity
- **Implemented using Savannah**
 - Problems recorded as bugs
 - Solutions recorded as patches





	JRA1
	SA3
	SA1

	JRA1
	SA3
	SA1



- **Transparent release process**
 - It is clear when and why things happen
 - All information recorded
 - Statistics available
- **Improved communication**
 - And information retrieval/retention
- **Coarse granularity**
 - Need finer granularity for SA3 processes

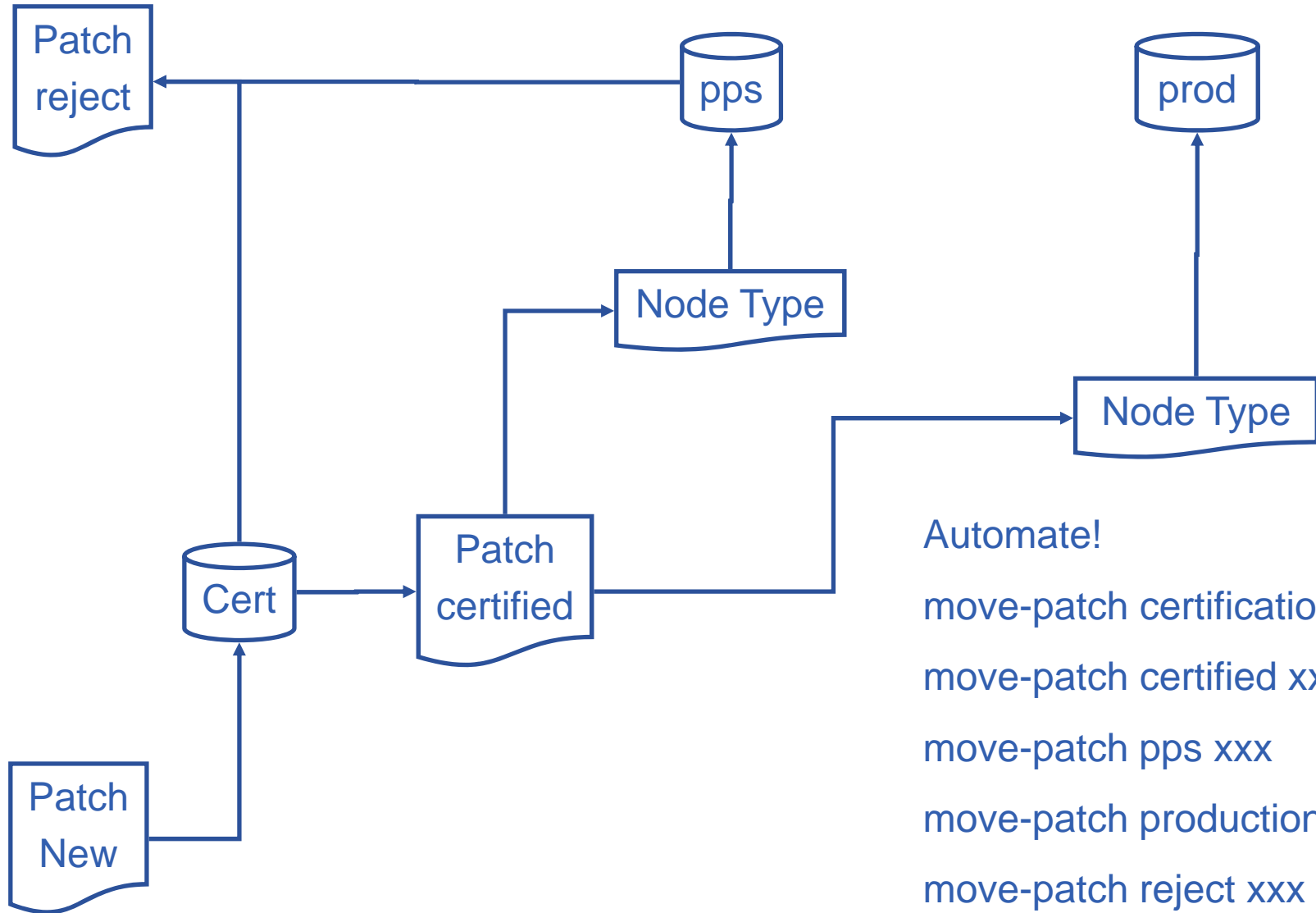
- **ETICS evaluated over the past year**
 - Still many areas for improvement
 - Specifying dependencies
 - Automatic Packaging
 - Information management
 - Performance/Documentation/Quality
 - Basic functionality of a build system is available
- **ETICS can replace the LCG build system**
 - Specifies build targets
 - Maintains static version list
 - Remote builds available
- **ETICS can replace the glite build system**
 - glite 3.1 100% successful

- **Porting code is not sustainable**
 - It is relying on “hero” effort
- **A release must be available for all supported platforms**
 - At the same time, should stop after the release porting efforts
- **Release process must be done in parallel on supported platforms**
 - Code must be made portable
- **ETICS has the ability to do parallel builds**
 - This must become a standard part of the process
- **Aim: Parallel support for multiple platforms**
 - Porting does not describe this process
- **Activity Should provide**
 - Build platforms for ETICS
 - Testing facilities for the supported platforms
 - Problem notification
 - And problem follow-up.

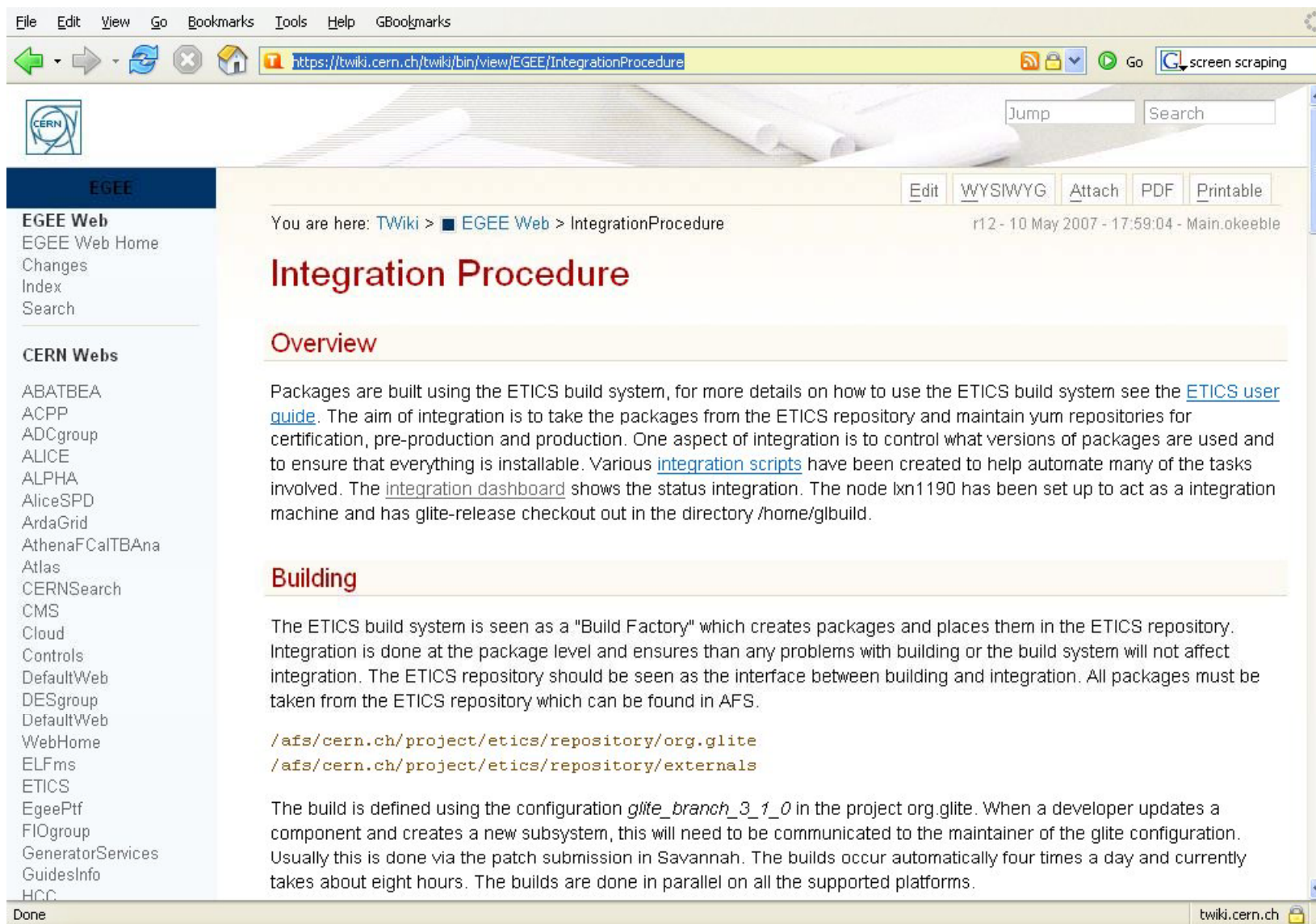
- **Currently using project config “glite_brach_3_1_0”**
 - Updating to build each new subsystem
 - Effectively building “HEAD”
 - Causing many problems
- **Need to freeze “glite_branch_3_1_0”**
 - This will be the “baseline”
- **Developers should do remote builds**
 - To build the packages
 - And test these before submitting a patch

- **Aim: Provide working software repositories**
 - Taking packages from build system repository
- **The ETICS repository contains all packages**
 - Developers should ensure that they get there
- **Need to control which versions are taken**
 - In addition create a few meta-packages
- **Information enters system as a patch**
 - New node types can be considered a patch
- **Integration finishes with a repository populated**
 - And meta packages created
- **The problem is only information management!**
 - But needs to be automated!

- **Production**
 - release
 - updates
 - externals (not supported)
- **PPS**
 - pps-updates
- **Certification**
 - Patches/Number
- **PPS and Cert also use production repository for release and updates**



Automate!
 move-patch certification xxx
 move-patch certified xxx
 move-patch pps xxx
 move-patch production
 move-patch reject xxx



File Edit View Go Bookmarks Tools Help GBookmarks

https://twiki.cern.ch/twiki/bin/view/EGEE/IntegrationProcedure

Jump Search

EGEE

EGEE Web
EGEE Web Home
Changes
Index
Search

CERN Webs

- ABATBEA
- ACPP
- ADCgroup
- ALICE
- ALPHA
- AliceSPD
- ArdaGrid
- AthenaFCaITBAna
- Atlas
- CERNSearch
- CMS
- Cloud
- Controls
- DefaultWeb
- DESgroup
- DefaultWeb
- WebHome
- ELFms
- ETICS
- EgeePtf
- FIOgroup
- GeneratorServices
- GuidesInfo
- HCC

You are here: TWiki > EGEE Web > IntegrationProcedure

r12 - 10 May 2007 - 17:59:04 - Main.okeebble

Integration Procedure

Overview

Packages are built using the ETICS build system, for more details on how to use the ETICS build system see the [ETICS user guide](#). The aim of integration is to take the packages from the ETICS repository and maintain yum repositories for certification, pre-production and production. One aspect of integration is to control what versions of packages are used and to ensure that everything is installable. Various [integration scripts](#) have been created to help automate many of the tasks involved. The [integration dashboard](#) shows the status integration. The node lx1190 has been set up to act as a integration machine and has glite-release checkout out in the directory /home/glbuild.

Building


The ETICS build system is seen as a "Build Factory" which creates packages and places them in the ETICS repository. Integration is done at the package level and ensures than any problems with building or the build system will not affect integration. The ETICS repository should be seen as the interface between building and integration. All packages must be taken from the ETICS repository which can be found in AFS.

```
/afs/cern.ch/project/etics/repository/org.glite
/afs/cern.ch/project/etics/repository/externals
```

The build is defined using the configuration `glite_branch_3_1_0` in the project org.glite. When a developer updates a component and creates a new subsystem, this will need to be communicated to the maintainer of the glite configuration. Usually this is done via the patch submission in Savannah. The builds occur automatically four times a day and currently takes about eight hours. The builds are done in parallel on all the supported platforms.

Done twiki.cern.ch

- **How do we generate and maintain lists?**
 - Tried many different methods in ETICS
- **Integration requires package name and version**
 - ETICS uses configuration names and component names
 - No easy mapping
 - Require bulk operations on many lists
 - Would like to query across multiple lists


Build System

Project name: org.glite
Project config: Unknown
Module name: org.glite
Module config: glite_branch_3_1_0
Build start time: 21/05/2007 13:48:39
Success rate: 99% (345/349)
Status: Failed

Page generated at 21/05/2007 20:59:34
[Back to module overview page](#)

Component name	Configuration name	Last build time	Result
bdii	bdii_R_3_8_8_1	21/05/2007 13:48:58	Success
glite-version	glite-version_R_3_1_0	21/05/2007 13:49:11	Success
org.glite.misc	org.glite.misc.HEAD	21/05/2007 13:49:15	Success
lcg-info-templates	lcg-info-templates-lcg1_0_15	21/05/2007 13:49:28	Success
lcg-infosites	lcg-infosites_R_2_6_2_1	21/05/2007 13:49:37	Success
lcg-info-dynamic-software	lcg-info-dynamic-software-lcg1_0_3	21/05/2007 13:49:46	Success
org.glite.info.generic	org.glite.info.generic_R_2_0_0	21/05/2007 13:49:55	Success
lcg-info-dynamic-dpm	lcg-info-dynamic-dpm-lcg1_3_3	21/05/2007 13:50:04	Success
glite-info-dynamic-scheduler-lsf	glite-info-dynamic-scheduler-lsf_R_2_0_0	21/05/2007 13:50:14	Success
lcg-info	lcg-info_R_1_8_0_1	21/05/2007 13:50:23	Success
org.glite.info	glite-info_R_0_0_1_0	21/05/2007 13:50:27	Success
vdt_globus_essentials	vdt_globus_essentials v. 4.0.3-VDT-1.6.0	21/05/2007 13:50:51	Success
org.glite.build.common-cpp	glite-build-common-cpp_branch_3_1_0	21/05/2007 13:51:09	Success
org.glite.service-discovery.api-c	glite-service-discovery-api-c_R_2_2_2_2	21/05/2007 13:51:19	Success
org.glite.service-discovery.build-common-cpp	glite-service-discovery-build-common-cpp_R_0_2_0_1	21/05/2007 13:51:32	Success
glib2-devel	glib2-devel v. 2.4.7	21/05/2007 13:51:40	Success
org.glite.service-discovery.cli	glite-service-discovery-cli_R_2_2_1_2	21/05/2007 13:52:19	Success
org.glite.service-discovery.bdii-c	glite-service-discovery-bdii-c_R_2_2_2_2	21/05/2007 13:53:05	Success
org.glite.service-discovery.file-c	glite-service-discovery-file-c_R_2_1_2_2	21/05/2007 13:53:53	Success
globus	globus v. 4.0.3-VDT-1.6.0	21/05/2007 13:54:07	Success
httpd-devel	httpd-devel v. 2.0.52	21/05/2007 13:54:19	Success
gsoap	gsoap v. 2.7.6b	21/05/2007 13:54:27	Success
org.gridsite.core	gridsite-core_R_1_1_18_1	21/05/2007 13:54:52	Success
org.glite.data.build-common-cpp	glite-data-build-common-cpp_R_0_3_0_1	21/05/2007 13:55:10	Success
org.glite.data.util-c	glite-data-util-c_R_1_2_1_3	21/05/2007 13:55:47	Success

Done



- **How do we generate and maintain lists?**
 - Tried many different methods in ETICS
- **Integration requires package name and version**
 - ETICS uses configuration names and component names
 - No easy mapping
 - Require bulk operations on many lists
 - Would like to query across multiple lists
- **Currently using flat files**
 - grep for querying across multiple lists
 - update-version script for bulk operations on multiple lists
 - Patch lists “screen scraped” from Savannah.
 - Node type lists stored in CVS
 - One from PPS and one for Production

- **Get the list of packages**
 - This can be “screen scraped” from the Savannah patch
- **Check that the packages exist**
 - This can be done automatically

File Edit View Go Bookmarks Tools Help GBookmarks

https://grid-deployment.web.cern.ch/grid-deployment/cgi-bin/reports.cgi?action=patch

egee sa3

EGEE-II: SA3 Activity Patch Status

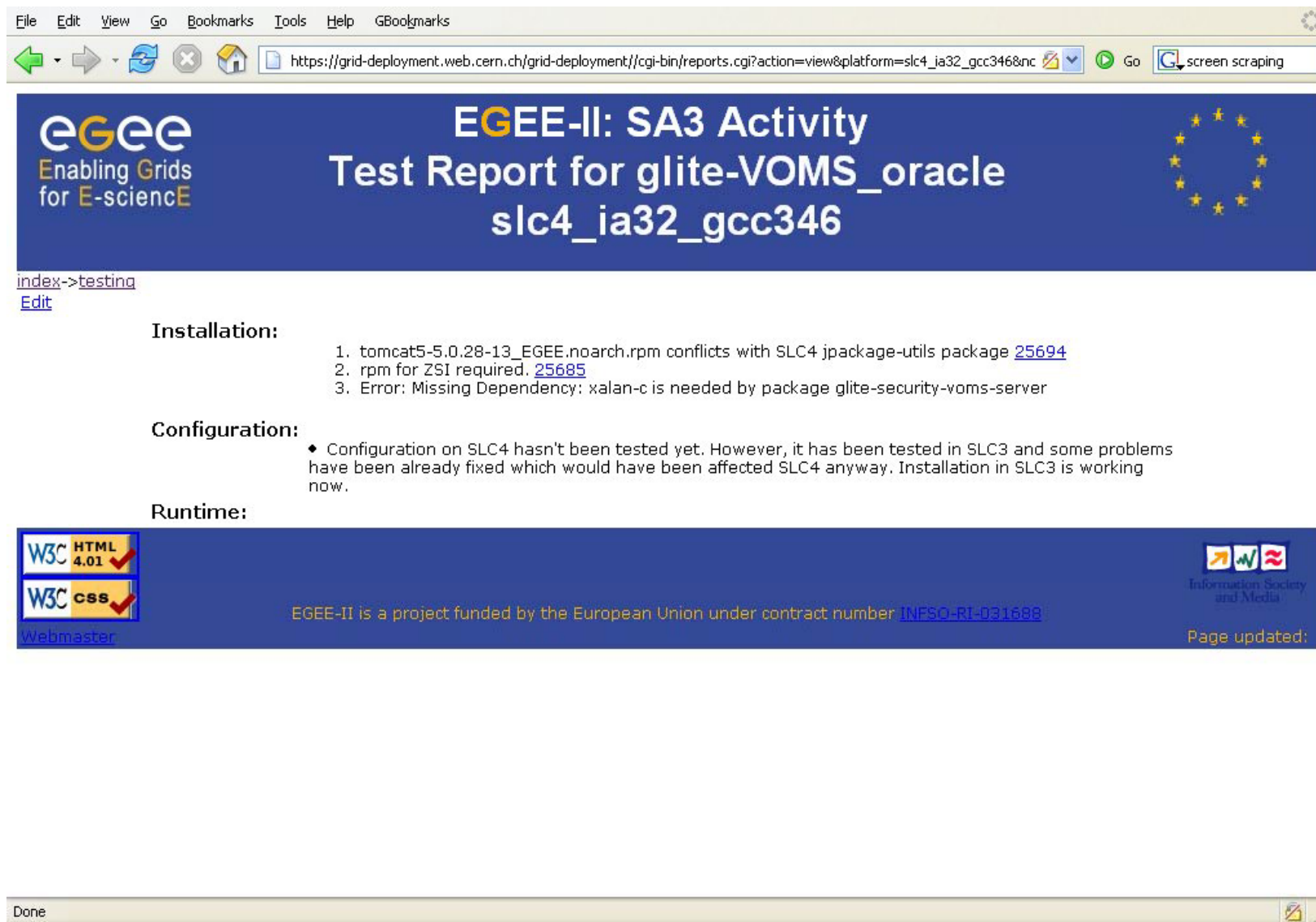
[index](#)

Node	slc3	ia32	qcc323	slc4	ia32	qcc346	slc4	x86	64	qcc346
1029		X			X				X	
1031		X			X				X	
1040		X			X				X	
1044		X			X				X	
1047		X			X				X	
1062		X			X				X	
1075		X			X				X	
1089		X			X				X	
1093		X			X				X	
1096		X			X				X	
1107		X			X				X	
1112		X			X				X	
1113		X			X				X	
1114		X			X				X	
1116		X			X				X	
1117		X			X				X	
1120		X			X				X	
1121		X			X				X	
1123		X			X				X	
1128		X			X				X	
1132		X			X				X	
1133		X			X				X	
1138		X			X				X	
1140		X			X				X	

Done

- **Get the list of packages**
 - This can be “screen scraped” from the Savannah patch
- **Check that the packages exist**
 - This can be done automatically
- **Update the repository**
 - For certification
 - Create a subdirectory with patch number in patches
 - For PPS
 - Update the node lists with new versions from the patch
 - *May require manual intervention for new packages and node types*
 - Create new meta packages if needed
 - *Add all new packages to pps-updates*
 - For production
 - Same as pps but using different node lists

- **Certification requires testing**
 - Not discussed here!
- **Iteration cycles need to be small**
 - In order to make progress in the shortest possible time
 - One problem fixes shows up another
- **Communication is the biggest problem area**
 - Causing slow iterative cycles
 - Information lost
 - IM, e-mail, meetings, pair wise work etc.
 - *Every problem needs to be recorded.*

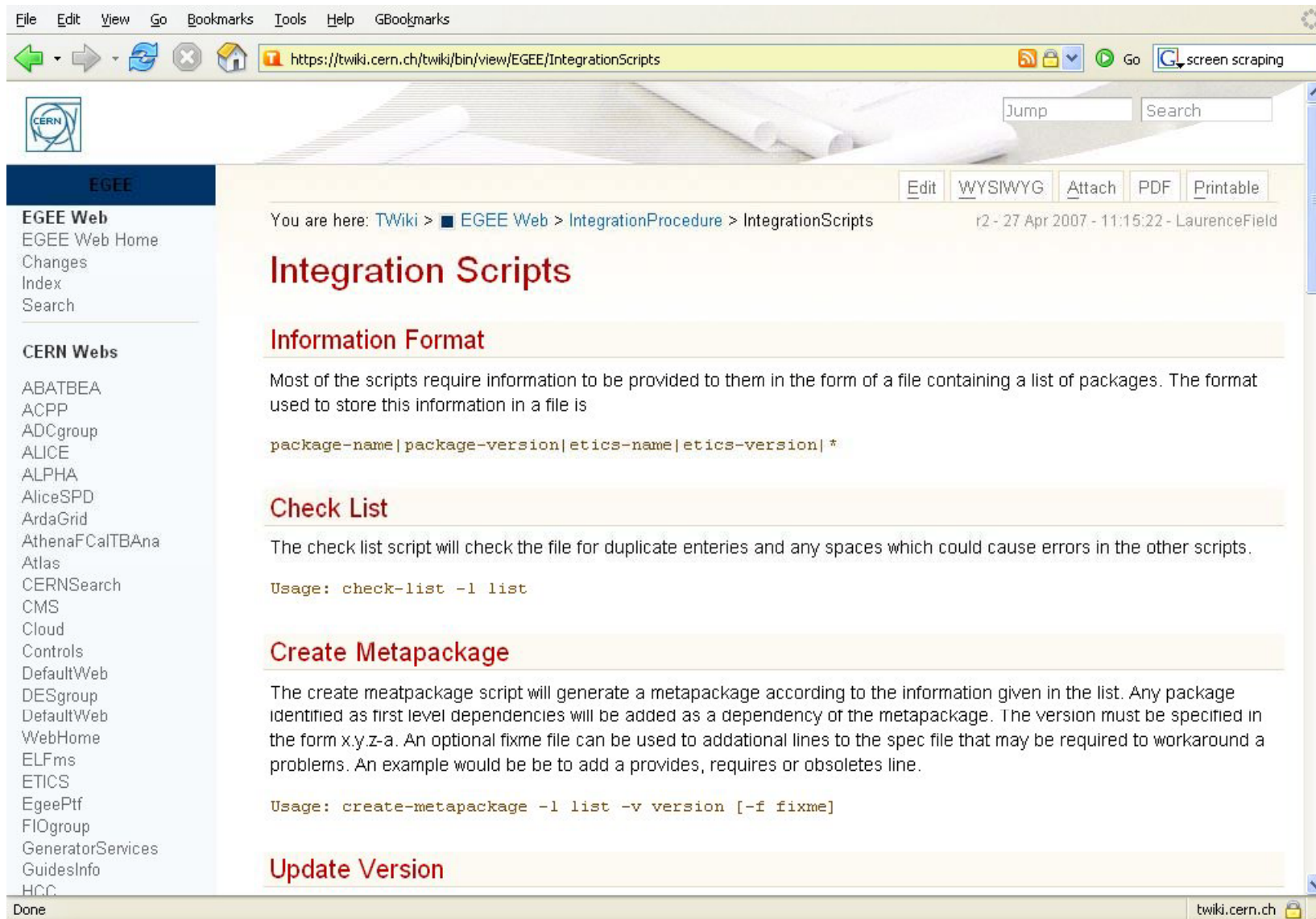


The screenshot shows a web browser window with the following content:

- Browser address bar: `https://grid-deployment.web.cern.ch/grid-deployment/cgi-bin/reports.cgi?action=view&platform=slc4_ia32_gcc346&nc`
- Page header: EGEE logo, "Enabling Grids for E-science", "EGEE-II: SA3 Activity", "Test Report for glite-VOMS_oracle slc4_ia32_gcc346", and the European Union flag.
- Navigation links: [index->testing](#) and [Edit](#)
- Installation:**
 - tomcat5-5.0.28-13_EGEE.noarch.rpm conflicts with SLC4 jpackage-utils package [25694](#)
 - rpm for ZSI required. [25685](#)
 - Error: Missing Dependency: xalan-c is needed by package glite-security-voms-server
- Configuration:**
 - ◆ Configuration on SLC4 hasn't been tested yet. However, it has been tested in SLC3 and some problems have been already fixed which would have been affected SLC4 anyway. Installation in SLC3 is working now.
- Runtime:**
- Footer: W3C HTML 4.01 and CSS validation logos, "Webmaster:", "EGEE-II is a project funded by the European Union under contract number [INFSO-RI-031688](#)", "Information Society and Media" logo, and "Page updated:".
- Bottom status bar: "Done"

A screenshot of a web browser window. The address bar shows the URL: https://grid-deployment.web.cern.ch/grid-deployment/cgi-bin/reports.cgi?action=index. The page content includes the EGEE logo, the title 'EGEE-II: SA3 Activity Integration Dashboard', and a list of links: Node Status, Patch Status, Porting Status, and Testing. At the bottom, there are W3C HTML 4.01 and CSS validation icons, a 'Webmaster' link, a funding statement 'EGEE-II is a project funded by the European Union under contract number INFISO-RI-031688', and the 'Information Society and Media' logo. The page is marked as updated.

https://grid-deployment.web.cern.ch/grid-deployment/cgi-bin/reports.cgi?action=testing grid-deployment.web.cern.ch



The screenshot shows a web browser window with the address bar containing `https://twiki.cern.ch/twiki/bin/view/EGEE/IntegrationScripts`. The page content includes a navigation menu on the left, a breadcrumb trail, and three main sections: 'Information Format', 'Check List', and 'Create Metapackage'. The 'Information Format' section contains a code snippet for package specifications. The 'Check List' section includes a usage example for the `check-list` script. The 'Create Metapackage' section includes a usage example for the `create-metapackage` script. The browser status bar at the bottom shows 'Done' and the URL `twiki.cern.ch`.

- **Software process is working**
 - Information and communication via Savannah
- **ETICS can be used as a “simple” build system**
 - The parallel builds are a definite advantage
 - glite 3.1 branch should be frozen
 - Developers should start doing remote builds themselves
- **Everything that we do needs to be automated**
 - Do it, understand it, do it again, then automate
- **Integration procedure has been defined**
 - And is in the middle of being automated
 - “move-patch” command should drive everything
 - Integration dashboard gives a good overview
 - Test reports have proven help information flow and communication

- **Move-patch certification**
 - Get_patch()
 - input: patch number
 - Output: list of require packages found from savannah page
 - Check_packages()
 - Input: list of packages corresponding to patch
 - Output: Found packages to stdout, missing to stderr
 - Update-repository()
 - Input: repository location, list
 - Output: updates the certification repository
- **Move-patch certified**
 - Update-savannah state()
 - Input: patch, state
 - Output: updates Savannah state page
 - Remove-patch()
 - Input: patch number
 - Output: removes patch from certification

- **Move-patch pps**
 - Get_patch()
 - input: patch number
 - Output: list of require packages found from savannah page
 - *Should fail if patch not in state “certified”*
 - Update-nodelist()
 - Input: patch list
 - Output: updated node lists in CVS and tag.
 - *Should warn if package can’t be found*
 - Check_packages()
 - Input: list of packages corresponding to patch
 - Output: Found packages to stdout, missing to stderr
 - Update-repository()
 - Input: repository location, list
 - Output: updates the certification repository