



Enabling Grids for E-scienceE

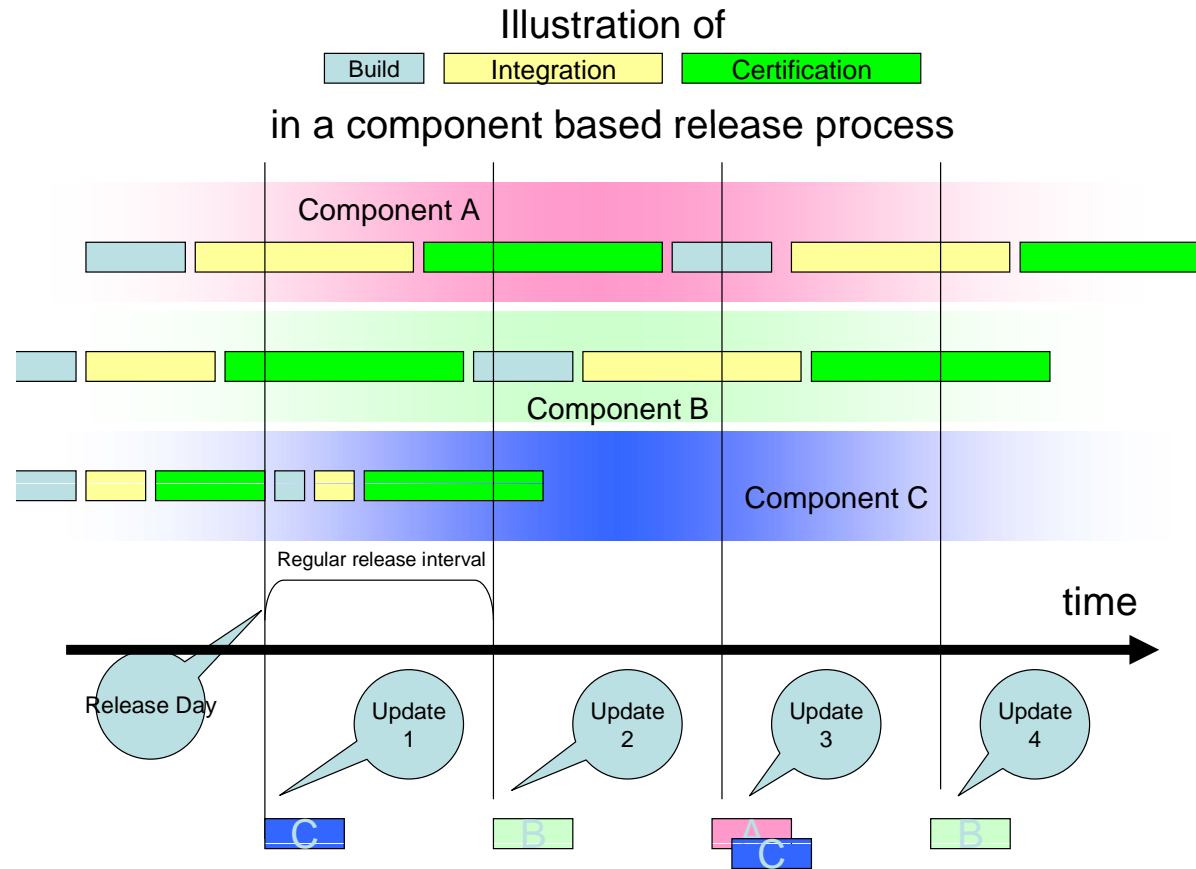
The future of the gLite release process

Oliver Keeble
SA3 CERN

www.eu-egee.org



- **A ‘gLite release’ is something we do every week.**
 - There are no more ‘big bangs’
 - Process operates on patches
 - gLite 3.1 services are being introduced gradually
 - Updates are released to PPS and production
 - Typically including up to 5 patches
 - Currently maintaining 2 releases
 - gLite 3.0 == SL3 == VDT 1.2
 - gLite 3.1 == SL4 == VDT 1.6 + ETICS build system
 - New gLite 3.1 services are being handled using as much of the existing process as possible
 - Addition of a ‘preparation’ phase before the formal process
 - Then the result is plugged into the process like any patch
 - Almost done!



- **Service Introduction**
- **Platforms**
- **ETICS**
- **Patches & Savannah**
- **Timetabling**
- **Release mechanisms**

- **Service introduction**
 - Certification is the **end** of the process
 - Certification begins after meeting acceptance criteria
- **Service preparation can involve**
 - Experimental services
 - “Incubation”
 - eg the UCY/AMGA experience
 - PPS
 - TROTJPWG
 - Separation of certification and testing
 - Clusters of competence
- **We now have Hydra**
- **... and new platforms**

- **More platforms**
 - Debian, 64 bit SL4
 - SL5 (=gLite3.2??)
 - the 'extended platform'
 - python 2.5
 - gcc 4.1
- **Porting and the release tools**
 - Some tools are based on rpm/yum
- **Process**
 - alignment of various platforms
 - management of the overheads – one change -> 4 patches?
- **metapackages and groups**
 - 64bt WN
- **glite-version and metapackages**

- **Source (or nosrc) rpms**
 - Produced by ETICS
 - currently there is an infrequent 'source build'
 - What about as input to the process?
 - These do not actually have to be buildable
- **Signed rpms**
 - build or certify time?
- **Regression / deployment tests**
- **Our build definition is currently very confusing**
 - Some packages in the release are not there
 - Some packages there are not in the release

- **Patch recycling**
 - Much history and state is accumulated in a patch
 - Recreating this without any kind of API to our tracking system is painful
 - But this must be made clear as reading the 'date submitted' can leave very misleading impressions about certification rate
 - can we do this in PPS??
- **Bug states – 'ready for review', 'fixed' in particular**
- **What other savannah changes are needed**
 - Patch locking
 - Integration with ETICS?
 - ETICS currently has no concept of a patch
- **Patch rejection**
 - Why do we so rarely reject patches from PPS?
- **Patch priorities**
 - What does the 'priority' field currently mean?

PERCEPTION

- **Developers**
 - Too slow
- **Sites**
 - Too much
- **Latency and Throughput**

THROUGHPUT

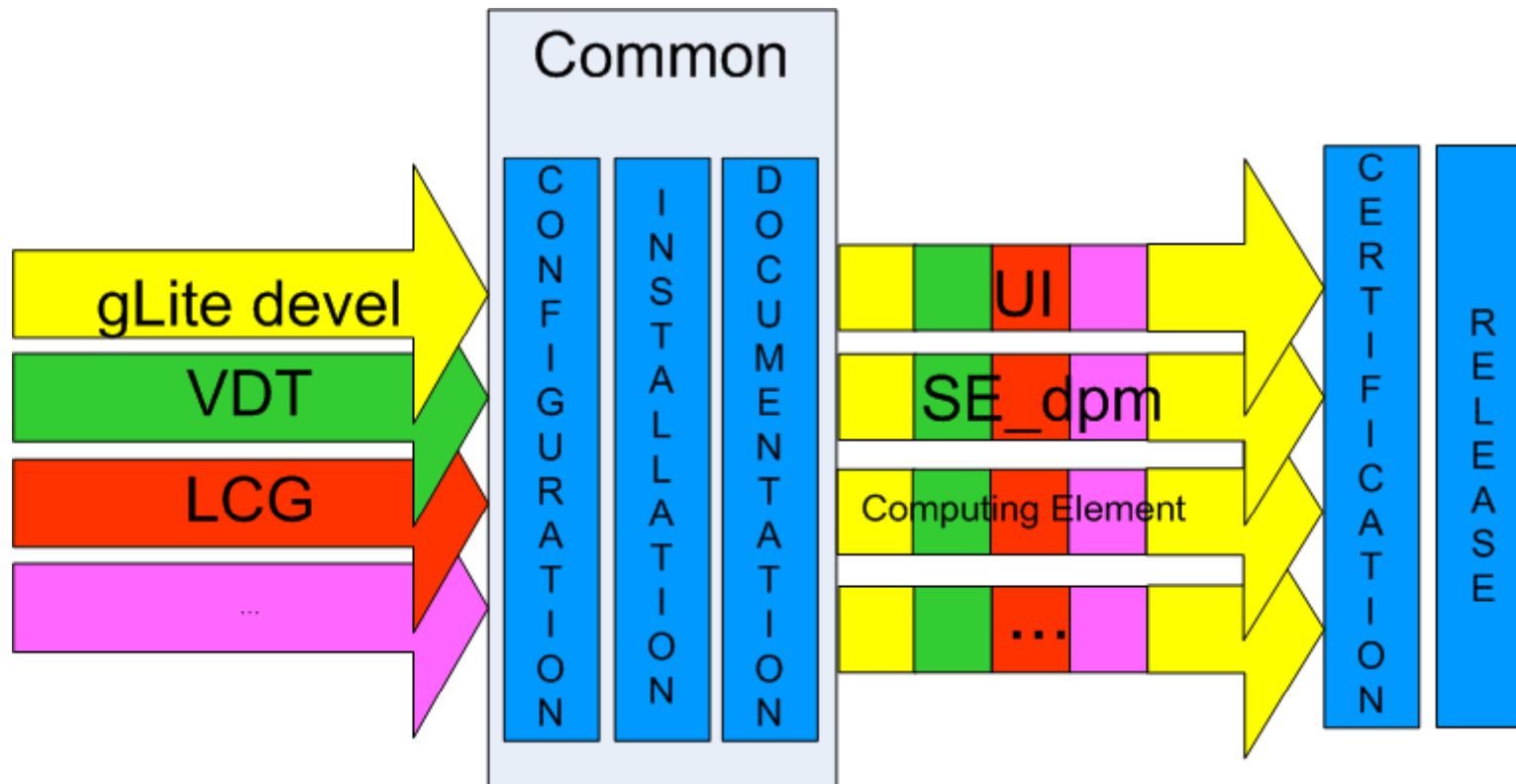
- **Limitation of patch flow**
 - can we say the pipeline is full?
 - can you miss your slot? 5 slots, 10 projects
- **Scaling**
 - Patches are very 'parallelisable'

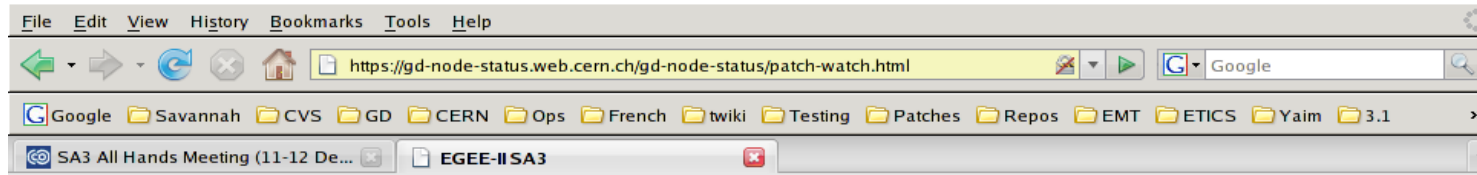
LATENCY

- **Release 'windows of opportunity' to PPS and prod**
 - How to reduce this effect?
- **Can we release configuration asynchronously?**
 - just for fast-track?
 - not appropriate for clients
 - yaim split has already improved responsiveness
- **Independence promotes low latency**

• **Process**

- 'Integration points' introduce delays and extra dependencies
- Important changes can be delayed by trivial ones
- Doesn't change throughput, but affects latency







EGEE-II: SA3 Activity

Patch Watch



Total:65

Queue Priority:High

#1370, 81days	#1369, 81days	#1389, 72days	#1491, 37days	#1517, 28days	#1544, 16days
---------------	---------------	---------------	---------------	---------------	---------------

Queue Priority:Normal

#1132, 234days	#1190, 182days	#1245, 143days	#1257, 135days	#1261, 126days	#1350, 94days	#1349, 94days
#1352, 93days	#1371, 81days	#1373, 80days	#1395, 71days	#1403, 65days	#1402, 65days	#1409, 60days
#1423, 58days	#1420, 58days	#1458, 53days	#1457, 53days	#1466, 46days	#1472, 44days	#1474, 43days
#1496, 35days	#1500, 32days	#1501, 31days	#1502, 30days	#1515, 29days	#1514, 29days	#1513, 29days
#1512, 29days	#1518, 28days	#1516, 28days	#1522, 24days	#1521, 24days	#1527, 23days	#1531, 22days
#1530, 22days	#1541, 17days	#1540, 17days	#1539, 17days	#1536, 17days	#1546, 16days	#1545, 16days
#1543, 16days	#1542, 16days	#1552, 15days	#1551, 15days	#1555, 14days	#1558, 10days	#1561, 8days
#1569, 3days	#1568, 3days	#1572, 2days	#1571, 2days	#1576, 1days	#1575, 1days	#1574, 1days

Queue Priority:Low

Queue Priority:On Hold



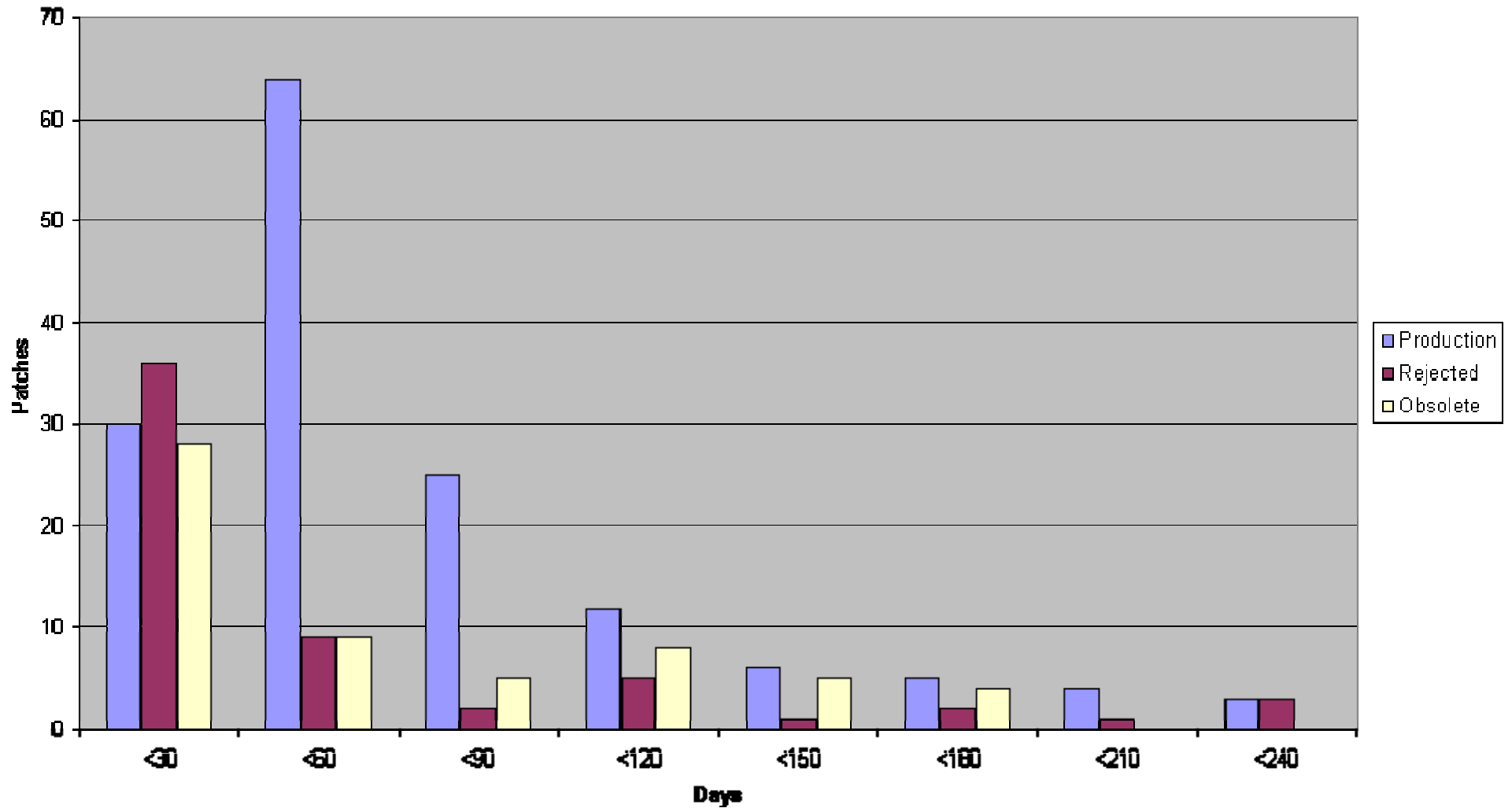

EGEE-II is a project funded by the European Union under contract number [INFSO-RI-031688](#)



Page updated: 2007-12-07

Done

Patch Latency



- **Different approach to clients?**
 - The requirements are pretty different for the clients
 - We can distribute them like experiment software
 - All porting efforts are concentrated here
 - Are currently made available as tarballs too
- **Externals**
 - Delayed repositories
 - DAG, jpackage
- **'Useful stuff'**
 - SLCS, ... MPI, gridice?
 - hierarchical release
 - RESPECT

