



The role of integrated middleware distributions



Oliver Keeble &
Markus Schulz



Outline

- The way middleware is provided is being reexamined
 - Funding structure is changing
 - Technology is changing
- In this context, the effectiveness and purpose of integrated distributions is being looked at
 - What distributions are and why we have them
 - What we think they do
 - What they do
 - Their place in enabling grids
 - The future



Integrated Distributions

- We know many distributions
 - gLite
 - ARC
 - VDT
 - LCG
 - UMD
 - ...
- Also of course linux distributions
 - distrowatch.com lists 319 OSS distros
 - Specialisation
- Distribution has independently varying components
 - Each component at a unique version
 - Defined relationship between them
 - **eg can install together**
 - Components of heterogeneous origins





gLite

- gLite currently provides
 - ~30 metapackages
 - ~350 packages
 - ~10 patches per month
- Change is generated at the component level
- 'User' view is at the service level
 - This happens at the end of the release chain
 - Resulting releases are not targeted to specific services

Patch #	Description
1648	sl4/i386 New torque 2.3.0-snap.200801151629.2cri and Maui 3.2.6p20-snap.1182974819.8
1708	R3.1/SLC4/i386: glite-AMGA_oracle metapackage
1782	VOMS Admin Server 2.0.14.1 & VOMS Admin Client 2.0.7.1 & VOMS Admin Interface 2.0.2.1
1787	VOMS server configuration update (multiple bug fixes)
1802	New version of lcg-info to support multiple BDII endpoints in LCG_GFAL_INFOSYS
1854	New yaim to fix the bug #36982 in WMS patch 1726
1874	Fix for rpm conflicts in gLite 3.1 update 25

Priority	Service	Version	Details
Normal	glite-TORQUE_client	3.1.4-0	Details
Normal	glite-AMGA_postgres	3.1.6-0	Details
Normal	glite-LB	3.1.1-1	Details
Normal	glite-VOBOX	3.1.13-0	Details
Normal	glite-VOMS_oracle	3.1.11-0	Details
Normal	glite-WMS	3.1.2-0	Details
Normal	lcg-CE	3.1.16-0	Details
Normal	glite-AMGA_oracle	3.1.1-0	Details
Normal	glite-TORQUE_server	3.1.4-0	Details
Normal	glite-VOMS_mysql	3.1.11-0	Details
Normal	glite-WN	3.1.15-0	Details
Normal	glite-UI	3.1.15-0	Details



Why middleware distros?

- Reasons to distribute middleware as a distro;
 - simplify availability of services and updates
 - provide integrated services
 - identify endorsed products at specific versions
 - meta packages, config, docs
 - [promote interoperability]
 - With standards this is no longer necessary
 - [allow co-installation of services]
 - not a use case any more
 - [allow co-installation of clients]
 - use a common SDK

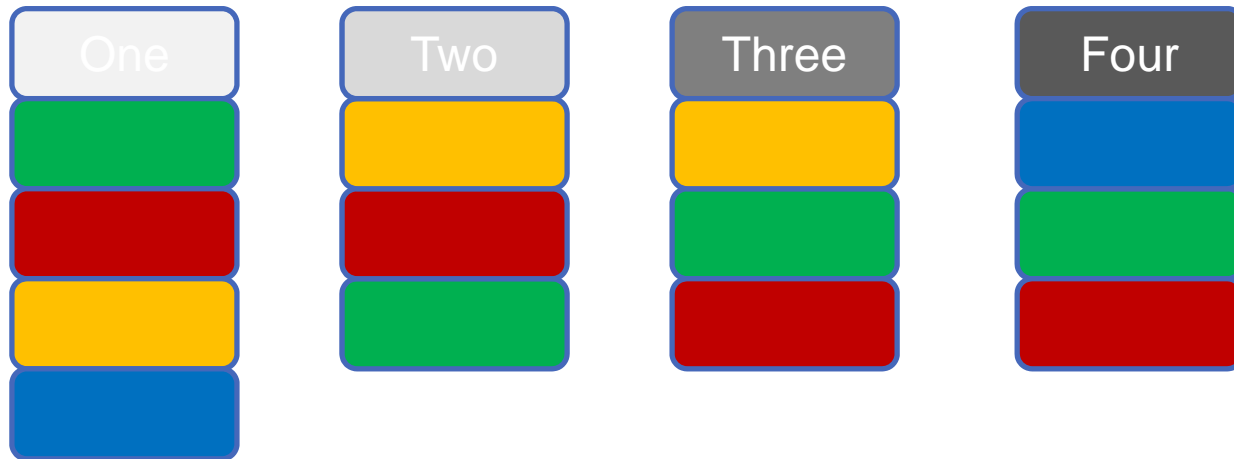


Extending the concept

- What is installed on a node comes from many distributions
 - middleware+externals+RHEL+jpackage+DAG+...
 - Middleware distribution contains much more than is on the node
- Some distributions are evolving into appliances
 - linux distros for specific tasks, eg firewalls, routers etc
- gLite has the concept of node-types
 - In EGI speak this is the 'product' of the 'product team'
 - Should the 'node-type' become the distribution?
- The release team could be integrated too
 - Development, test & release all in one team
- Advantages
 - Freer variation of dependencies – greater reactivity
 - Releases genuinely targeted by service
 - Accountability associated with change generation
 - Maps onto likely distribution of effort in the future
 - More efficient certification and change management



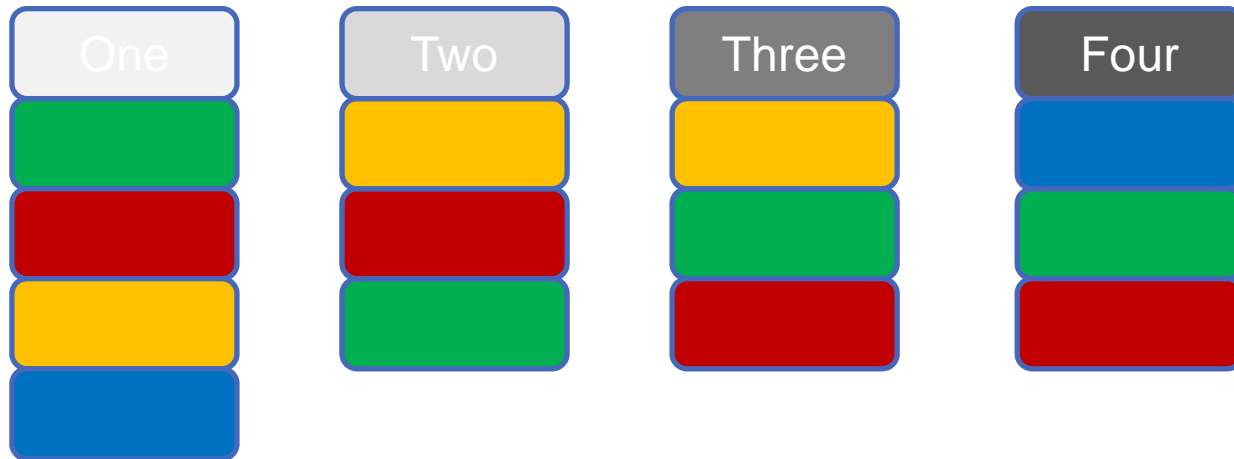
Current situation



- Release timetable was dictated by the slowest component
- 11 certifications



Independent distributions



•7 certifications



Disadvantages...?

- [testing resources needed by product team]
 - But a team should know the environment in which its product will be deployed
- [multiple versions of libraries etc on infrastructure]
 - this we have now!
- [compatibility at the protocol level must be assured]
 - A requirement now
- Will system testing be effective?
- Common layers have to be implemented differently
 - By common conventions rather than by a single team
- How are 'internal releases' handled?



Future context

- EGEE-III project stops next year
- Future directions are being discussed for 'the EGI era'
 - gLite consortium is being set up
 - EGI.org has an 8 person middleware unit
 - UMD
 - Harmonise the ARC, gLite and UNICORE stacks
 - Release strategies under negotiation
 - Less centralised effort will be available
- All are efforts to produce something sustainable
 - Must be decentralised



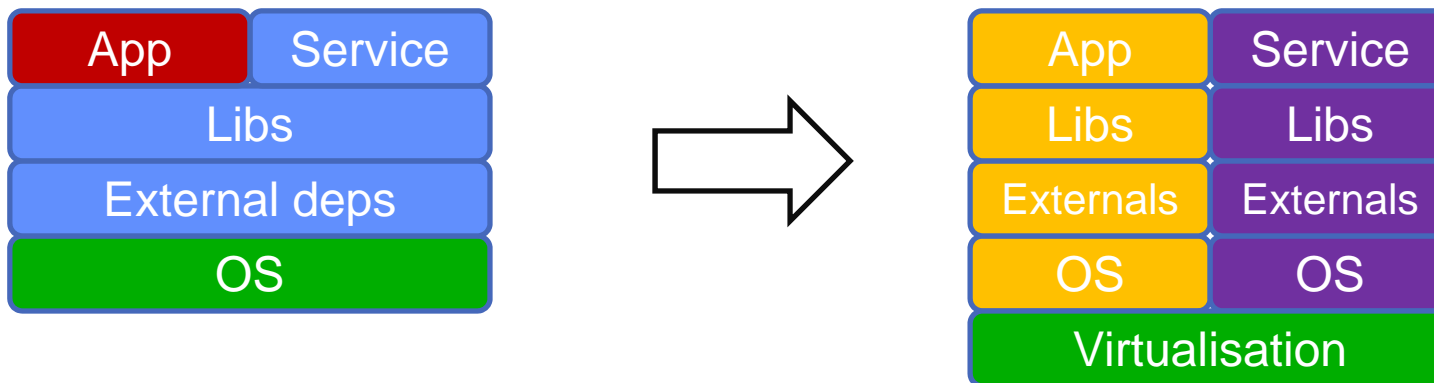
Virtualisation

- At the moment, distribution is still split between OS and middleware
- Applications can be bigger than the OS, next step is distribution of the whole thing via VMs.
- An appliance is a full OS/middleware/middleware stack
- Particularly applicable to clients/applications
- Services would be harder
 - Distribution via product team
 - How to update a running service?
 - Using local package managers?
 - From shared filesystems?



VMs and applications

- Distribution is via the application community
 - Based on inputs from m/w providers and others
- Each site maintains a library of VMs, supplied by their VO's.
 - VMs are triggered on an incoming job
 - Matching could be via current mechanisms
 - Or are managed cloud-style with pilot agents in them
- Decouples fabric from applications





Conclusions

- Distributions are being reexamined
- Likely to reduce in functional scope
 - Specialised distribution for each service
- Likely to extend vertical integration
 - Virtual appliances will come with complete OS
- Coherence will have to be maintained by shared conventions and standards
- This approach maps well onto possible funding models for further maintenance and development



Extras



Where it came from

- Initial motivation for linux distros
 - Everything had to install on the same machine
 - Integrated, pre-compiled binaries
 - Package Management & Installation
 - Didn't want to track hundreds of different packages
- Initial motivation for middleware distros
 - Similar to the above
 - Have inputs from many sources which have to co-install
 - Built using the same tools
 - apt/yum/rpm



What makes it a distro?

- Monolithic software releases
 - Everything is updated at once
 - Nothing works until everything works
 - Important changes can be slowed down by unimportant ones
- It is extensible