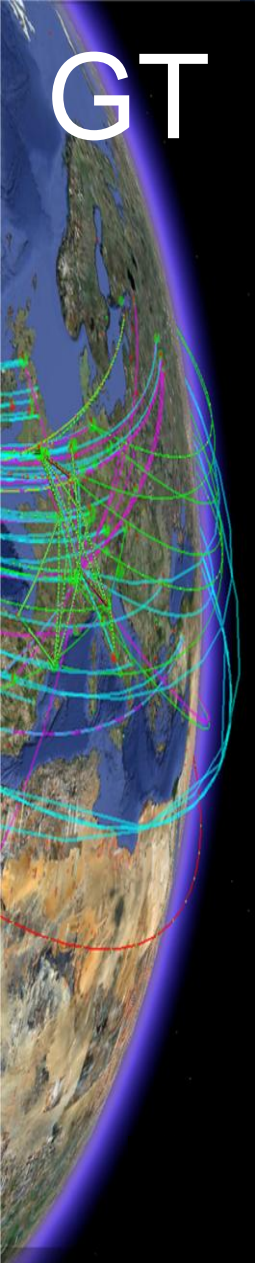
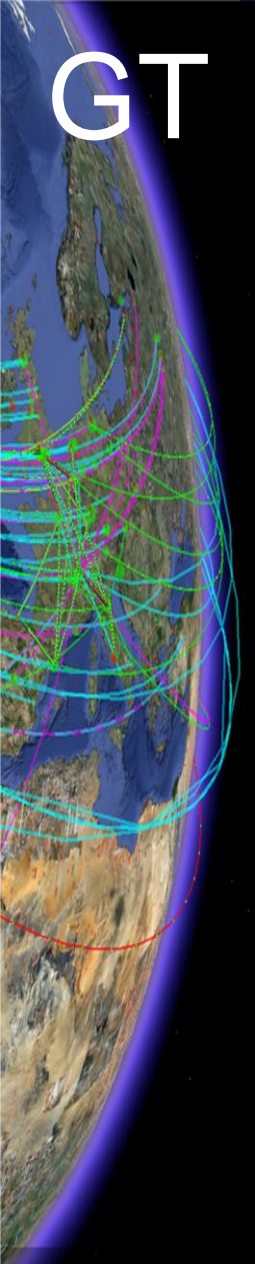


Ops TEG WG5

Oliver Keeble
On behalf of WG5



- WG5 is split into
 - Middleware distribution
 - Middleware deployment
 - Middleware configuration
- Feedback assembled on the twiki page
 - Classified
 - works well
 - top problems
 - strategic direction
 - saving manpower
- <https://twiki.cern.ch/twiki/bin/view/LCG/WLCGTegOperationsWG4WG5>
- Note: PT = Product Team



| Impact | Improvement |
|--------|---|
| 5 | Increase responsiveness of release process |
| 4 | Resolve repository proliferation |
| 3 | Cleaner expression of middleware dependencies |
| 2 | "Single channel" for release announcements |
| 4 | Maintain application area releases |
| 3 | Alternative client distribution strategies (eg CVMFS) |
| 4 | Closer involvement of experiments in pilots |

Embrace EPEL releases by PTs

- Advantages
 - More responsive releases
 - Easy for PTs to manage
 - Resolve repo proliferation
 - Good packaging
 - Source rpms
 - Converge on unique source for software and release news
- Disadvantages
 - Single platform
 - Restrictions (eg oracle linking)
 - Testing?

Expand Validation

- Capitalise on validation opportunities
 - All along the release process
 - Pre-certification pilots
 - Staged rollout
 - <https://twiki.cern.ch/twiki/bin/view/EMI/EMIBetaAcceptanceTesting>
 - NB EMI/EGI project timescales
- OSG has a “validation testbed”
 - An infrastructure isn’t enough
 - Needs to be used proactively

Explore AA release and CVMFS

- Application area release
 - Currently in AFS
 - `/afs/cern.ch/sw/lcg/external/Grid/`
- Pursue cvmfs
 - Will experiments incorporate this stuff naturally?
 - Is an independent grid client release via cvmfs useful?

| Impact | Improvement |
|--------|--|
| 5 | Improved documentation |
| 4 | Improved packaging (reducing need for subsequent configuration actions) |
| 4 | Simplification of middleware configuration (config file consolidation, files in standard locations, graceful config re-reads...) |
| 3 | Reinvest in yaim (cleanup, rollback...) What effort is available for this? |
| 2 | Improve quattor integration |
| 2 | Support puppet natively (ie templates supplied by middleware providers) |

PTs concentrate on low level config

Improvement

Improved documentation

Improved packaging (reducing need for subsequent configuration actions)

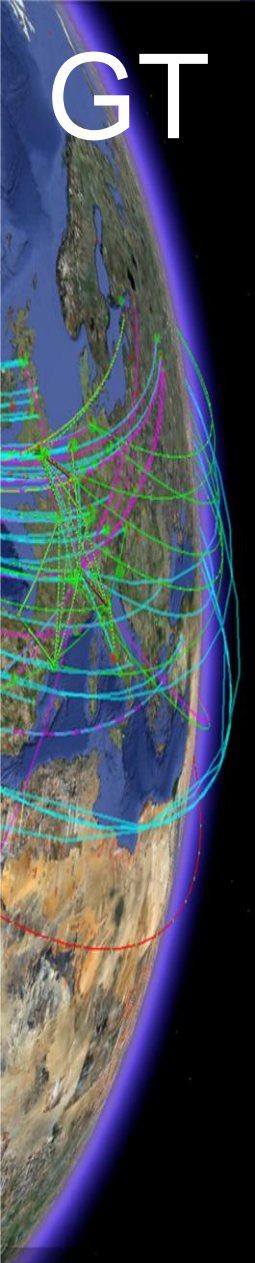
Simplification of middleware configuration (config file consolidation, files in standard locations, graceful config re-reads...)

- Who will own this?
 - Specific actions have to be identified
 - Generalities are not enough

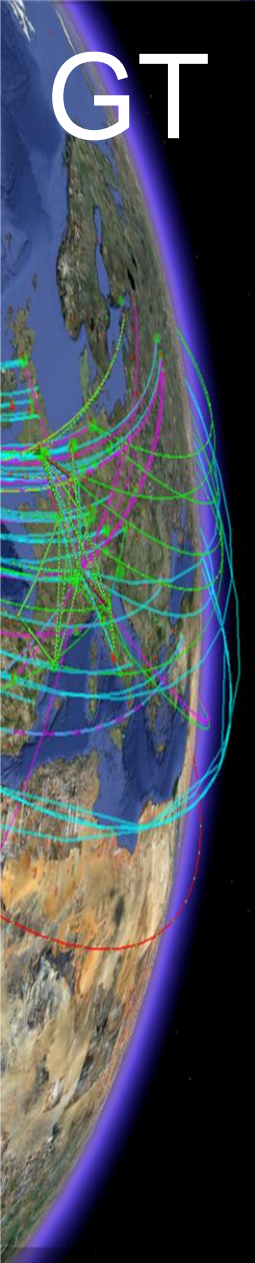
For config management: “Do Nothing”

- Don't invest further in yaim unless a long term maintainer is found
 - Currently with INFN until end of EMI
 - SL6
 - Needs changes for EPEL adaption
 - Needs support for rollback
 - Needs to play better with fabric management
- If the PTs focus on native config, config management engines can look after themselves.
 - Better to be done by admins than devs

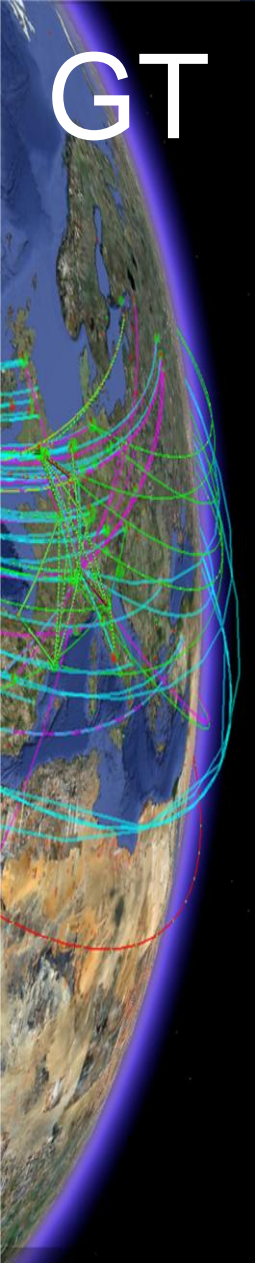
- Slides from previous talks, ignore



- A number of comments on the large investments sites make in configuration management
 - Middleware very diverse
- Simplification of the underlying configuration mechanisms
- Integration of middleware configuration with other systems (particularly quattor)
 - Mixed experiences
 - What's the direction?
- Choice of tools for the future
 - Integration with fabric/system management
 - Support by providers
 - Yaim/puppet
 - Opinion here is split



- How to better manage release validation
 - Atlas would like more opportunities for pre-release validation, pilots etc
 - CMS are concerned about the time they spend supporting middleware deployment at sites
 - EMI sees staged rollout as necessary and working
- How to ease deployment at a site
 - Comes down to packaging, repository and configuration issues
 - Updates are still seen as risky



- **Reactivity**
 - Concerns about the responsiveness of middleware release processes
- **Interaction with application area**
 - AFS release still required
- **Dependencies**
 - Often cited as a problem
 - Is the feeling that deps are badly expressed in packaging?
 - Or that the middleware simply has too complex a dependency tree?
- **Repository proliferation**
 - EMI, UMD(EGI), gLite, OSG, ARC, EPEL, dCache...
 - Not only a 'policy' problem, certain combinations of repos result in failed installations

