

# Post EMI WLCG Middleware Lifecycle Management

short summary of the longer text:

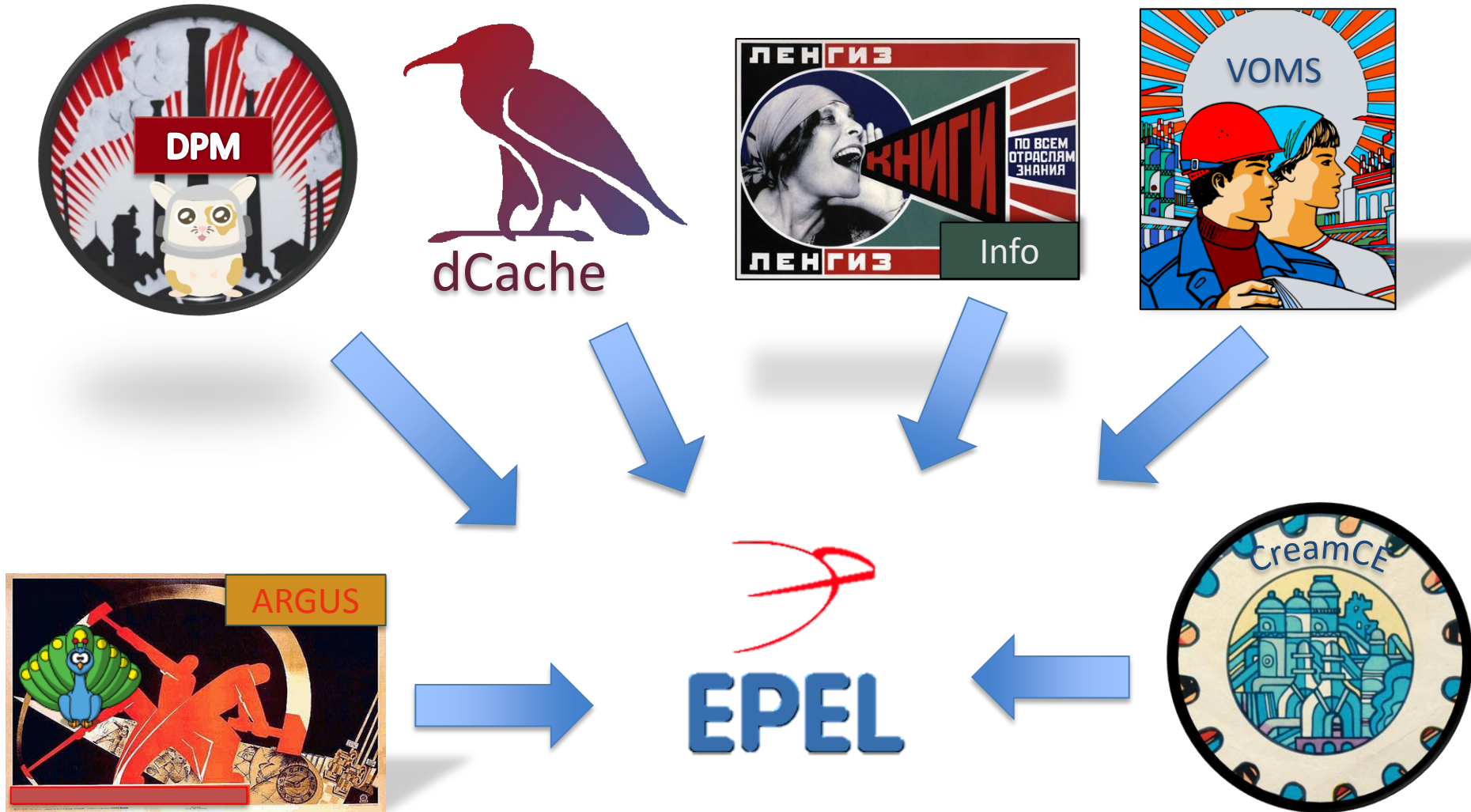
<https://indico.cern.ch/getFile.py/access?contribId=12&sessionId=1&resId=2&materialId=paper&confId=155073>

Markus Schulz

# Situation after EMI

- Most middleware will be quite stable
  - no major functional changes expected
  - most software in EPEL (or EPEL “like” packaging)
- No (very little) funding for central roles
- Products critical for WLCG supported by Product Teams located at HEP institutes
  
- → minimize central control
- → as close as possible to current approach
- → reuse existing structures
- → move responsibilities close to needs
  - batch system support, tarballs, etc.
- → you need it, you do it
  - or at least contribute to it

# Independent Product Teams





# Integration

- EPEL as integration point
  - Test and Stable Repository
- WLGC/UMD repository for non EPEL packages
  - Packages that can't live in EPEL
    - Commercial
    - Versioned Meta-Packages
- Tools
  - PTs' choice during development
  - Fedora tools for release
    - needed for EPEL



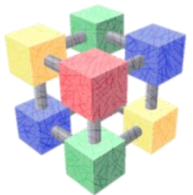
# Versioning

- Versioned Meta Packages
  - dependencies are not managed individually
  - version++ indicates changes
- Versions can be discovered and tracked
  - Resource info provider
    - encoded by public key
  - Local detailed logs by info provider for site admins
- Improved WLCGBaselineVersions page
  - recommended and lowest accepted versions
  - time line for required upgrade



# Testing and Verification

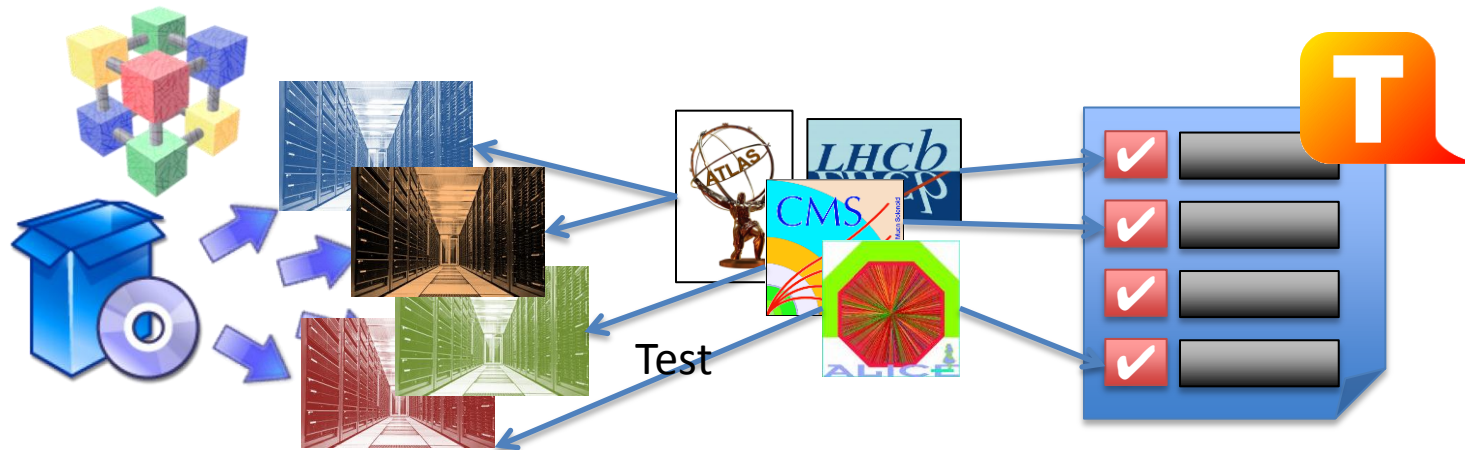
- PT independent tests
  - functional and regression tests
  - testing against production
  - site <-> PT coordination for resources
    - SEs, early pilots, test nodes,.....
- Pilot services
  - coordinated by WLCG-Ops Team
    - needs PT-Site-Experiment coordination



# Staged Rollout

- **Staged Rollout**

- coordinated by EGI and WLCG-Ops Team
  - via web page with fields for sites and experiments
- this is the critical step
- participating sites cover SEs and experiments
  - Active experiment sign off ✓



# Repositories

- EPEL Stable
- EPEL Test
- UMD/WLCG non EPEL material
  - used also for rollback
  - highest priority and protected
- UMD/WLCG History
  - anything ever released







- Staged Rollout catches most problems
  - but not all.....
- Using UMD/WLCG Repository and RPM tools
  - UMD/WLCG History
- This should happen only rarely
- Coordination:



# Configuration Management

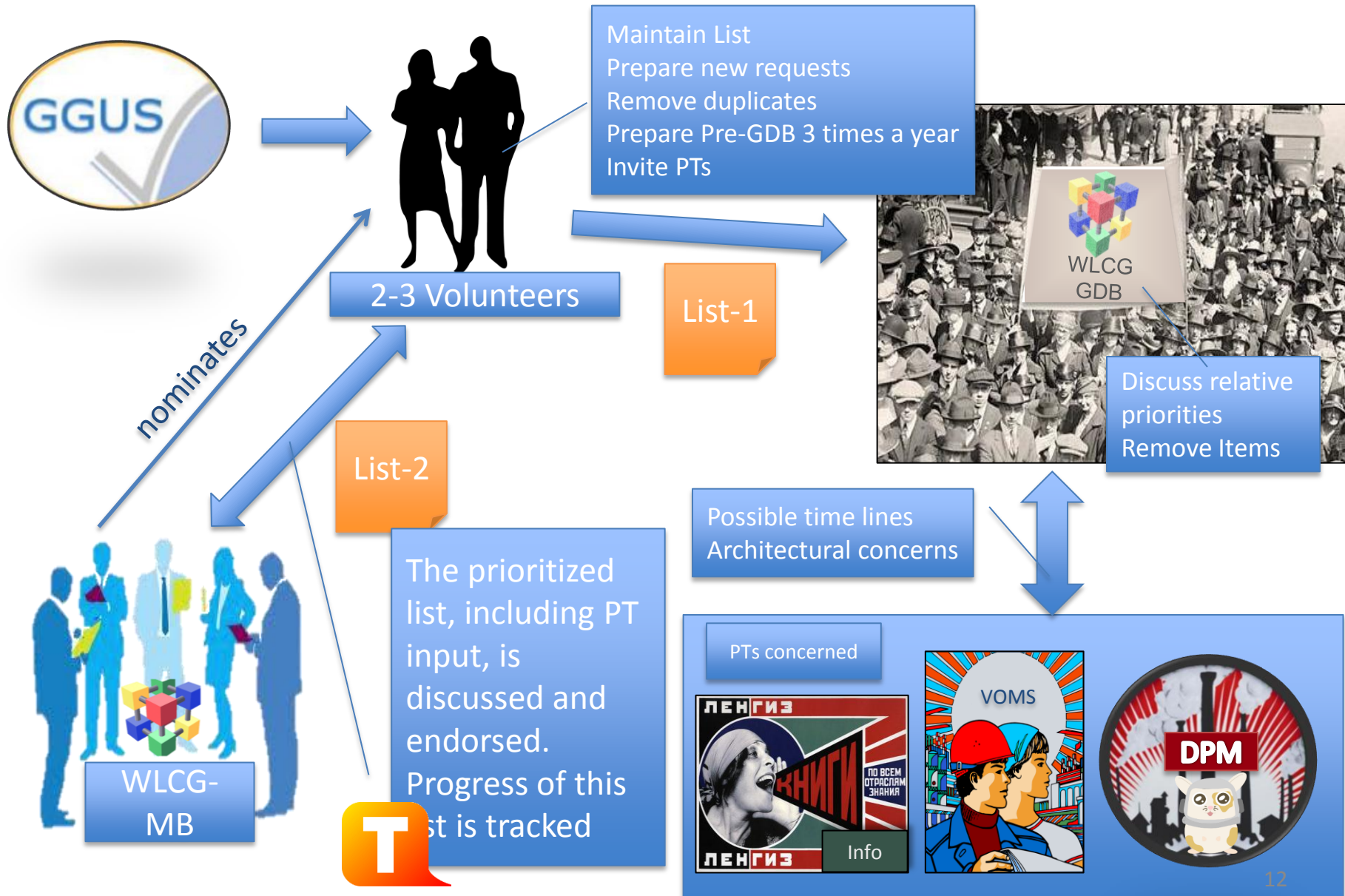
- Keep it simple
- Offer what you use
- Initially YAIM
- Long term:
  - Generic documentation (must)
  - PTs' preferences, like Puppet (may)
  - Sites should collaborate ( Chef, Quattor, Puppet..)



# Problem and Change Request Tracking

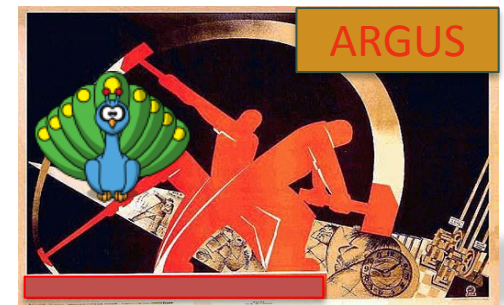


# Change Request Prioritization



# Expectation Management

- PTs are independent and **not WLCG financed**
- WLCG can express needs and priorities
- PTs' have their priorities
  - and less resources



# Open Questions

- Where can users, site managers and PTs exchange information?
  - planned changes, progress
  - documentation
  - FAQs, ops tools (exchange)
- Between library and supermarket
  - “one stop shop” preferred
  - simple navigation
  - some level of uniform presentation

# Is ScienceSoft the Answer???





# Open Question: WN/UI

- Who?
  - integration, configuration
- How?
  - RPMs, tarball, CernVMfs, Application Area.....
- First discussions on tarballs started
  - GridPP, ATLAS, CERN
  - Commitment needed
- Who will be the WN/UI PT ???

