



Enabling Grids for E-science

# Improving Quattor's Accessibility

*C. Loomis (CNRS/LAL)*

*Quattor Workshop (London)  
11-13 March 2009*

[www.eu-egee.org](http://www.eu-egee.org)



Information Society  
and Media



- **Quattor's Learning Curve**
- **Improving user experience.**
- **Improving developers experience.**

- **Quattor has a steep and difficult learning curve both for users and for developers.**
- **Some of this is linked to the comprehensive and complete nature of the quattor toolkit.**
- **But, much is linked to:**
  - Inadequate documentation.
  - Treating quattor as an all or nothing affair.
  - Organic development.
  - Evolutionary leftovers.

- **Although it is called the “quattor” toolkit. The name “quattor” is very seldom seen in the individual tools.**
  - ccm-fetch, aii-shellfe, panc, ...
  - ncm-cdispd, ...
  - EDG::WP4::CCM::, CAF::, LC::
  - /etc/ccm.conf, /etc/ncm-cdispd.conf, ...
- **Do the acronyms—CCM, AII, NCM, etc.—really serve any useful purpose now?**
- **This complexity is an unnecessary barrier for both users and developers.**

- **Documentation**
  - Lack of a good overview document.
  - Incomplete.
  - Often outdated.
  - Difficult to relate to a particular release of a tool.
- **Tutorials**
  - Often outdated.
  - Often treat full site management use case.
  - Hides incremental ways of integrating quattor into a site.
- **Again, this creates an unnecessary barrier for both users and developers.**

- **In some cases, have similar tools for the same function (e.g. CDB, SCDB, ...). Not necessarily bad, but...**
  - Must clearly document different strengths and weaknesses. (I.e. give the appropriate use cases.)
  - Have good justification for multiple tools.
- **Multiple APIs:**
  - For example, `get_tree()`, `get_element()`, .....
  - Need mechanism for marking deprecated elements.
  - And mechanism for marking public/private methods.

- **Coding Practices**
  - We have a set of coding practices for perl.
  - General agreement on using them.
  - Agreement to use LC, CAF libraries.
  
- **Nearly completely unenforced.**
  - Means that they are not rigorously followed.
  - Makes understanding and modifying code more difficult.
  - Makes common ownership of code difficult.

- **Pan:**
  - Eclipse-based editor.
  - Code checker.
  - Debugger.
  
- **Components:**
  - Lots of duplicated code for skeleton of component.
  - Cannot run component as standalone script.
  
- **Others?**



- **The community will not solve all of these issues overnight!**
- **What needs to be done:**
  - Agree on a list of problems.
  - Evaluate the impact of fixing each of the problems.
  - Implement them along with other developments.
- **Automation**
  - Can help in understanding what problems exist.
  - Prevent regression of things that have been fixed.