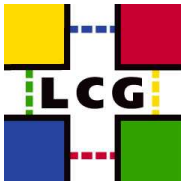
Thoughts on EGEE architecture

Jeff Templon

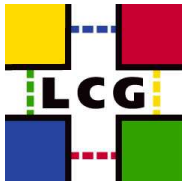
NIKHEF

templon@nikhef.nl



Outline

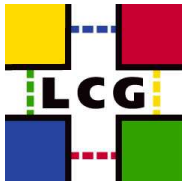
- A bit of history
- How to make integration easy
- How to make sure services can interoperate
- How to encourage developers to stay on track
- How EGEE can succeed in deploying a service grid



History

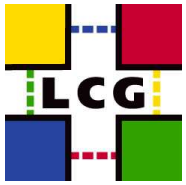
- DataGrid was started with a vague idea of what was needed*
- The original ATF started immediately with an API document
- Middleware development proceeded more or less independently
- At some point the "architecture" had lost enough relevance that people quit talking about it
- ATFng took a very different approach – investigate service invocation sequences for typical use cases, requiring explict exposure of all external calls – another way to "document, debug and fix the API"

*I wasn't there!



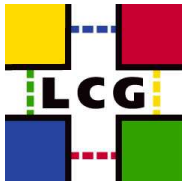
How to make integration easy

- The issues most frequently noted in connection with problems integrating EDG software
  - Installed software dependencies
  - Configuration dependencies
  - Service dependencies (accessibility)
  - Service dependencies (ill-defined component interactions)
- Dependencies must be
  - Discovered
  - Tracked
  - MINIMIZED!
    - Number of dependencies
    - Version dependence of remaining dependencies



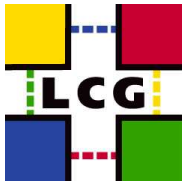
How to make sure services interoperate

- A service needs to have
  - A client API (how will a user use the service?)
  - A protocol (how will requests be sent over the net?)
- These must be PUBLISHED (not like GRAM)
- The client API standard means that Joe User can easily discover how to use the service, and can use a competitor's solution should one show up (just relink or maybe not even that!)
- The protocol standard means that clients and servers can talk to each other no matter whose actual implementation is involved



How to encourage developers to stay on track

- If a question is ill-posed, the answer is usually not satisfactory either
- Middleware developers need a clearly defined question from which to work
  - Give them an API which has been worked out via ATF-style service invocation sequence diagrams
  - Give them a definition of what the result of various invocations must be
  - Give them any known performance requirements
- At this point, the middleware is free to be creative while still providing the end result we want



How EGEE can succeed in deploying a service grid

- Need a culture of dependency-minimal deployment
  - Applications too – "paratrooper programming" is missing
- A strong ATF group needs to do early work (service sequences) to define APIs and protocols
  - Try to use standard, published, existing protocols where available and appropriate
  - Publish them!
  - Define process (review by all concerned parties & approval/rejection) for API evolution
- Middleware people must be BOUND by the API
  - Deviations are not deployed. It has to be this strict – deployment of deviating, unapproved software must be the exception rather than the rule.
- We know enough about Grids to do it this way!