



Enabling Grids for E-science

SA3 observations

Oliver Keeble

SA3

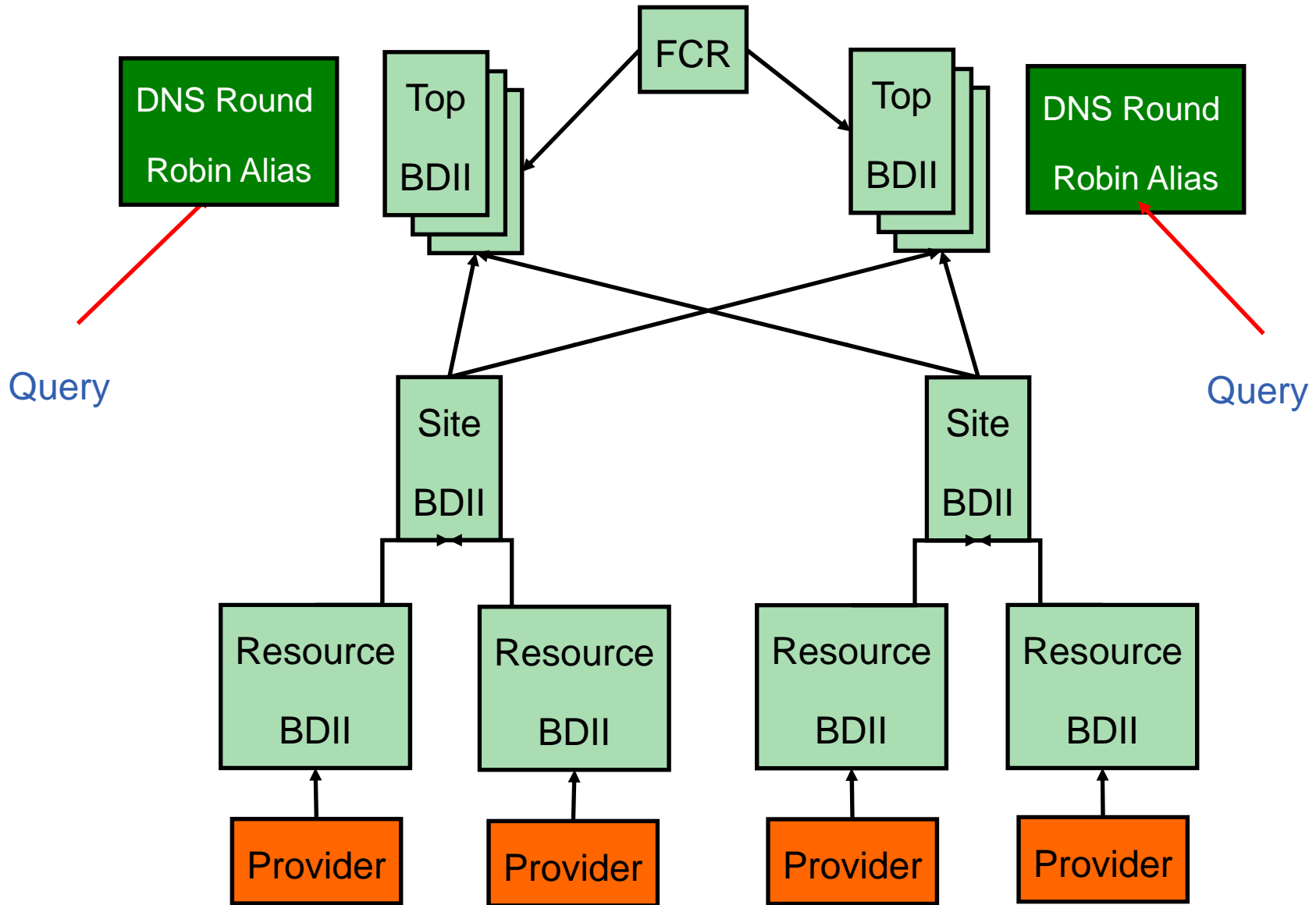
CERN

www.eu-egee.org



- **“The goal of the SA3 activity is to manage the process of building deployable and documented middleware distributions, starting by integrating middleware packages and components from a variety of sources” .**
- **The project produces an integrated middleware stack called gLite for which SA3 performs the following functions;**
 - Build
 - Integration
 - Configuration
 - Certification & Testing
 - Release
 - Interoperation
 - Information System

- **BDII implements the Information System**
 - has to scale with infrastructure (ie no. of clients & services)
- **BDII inherits an architecture from globus MDS**
 - Scaling to current infrastructure was not envisaged
 - First limit was reached at 4 sites!
 - Today, the we have over 250 sites in EGEE and have done simulations with double that size
- **Build the scaling in from the beginning**
 - It can be very difficult to anticipate
 - eg, how many IS queries do you expect a job to make?



- **Many incremental changes have kept it going**
 - parallelisation, ldapadd, multiple db instances, indexing, deployment advice, round-robin DNS
 - moving to a completely new IS now would be extremely disruptive
 - at the moment we have no API
 - evolutionary approach necessary
 - next... client side caching?
- **Our IS is used for a number of purposes, thus it is not optimised for its primary purpose, service discovery**

- **EGEE interoperates with OSG**
 - we share an enormous amount of middleware
 - including the security infrastructure
 - it's still hard work!
- **Interoperability is much more easily achieved**
 - unifying processes to achieve interoperation is harder
- **ARC and UNICORE have shown that commitment and perseverance are necessary ingredients**
- **Many ways...**
 - Mutual deployment of clients
 - Gateways
 - Common standards

- **gLite is a large scale integration project**
 - unites various contributor projects over which we have varying degrees of control
- **Each input project has its own configuration scheme**
- **We need to present a uniform interface for the administrator**
- **YAIM is**
 - bash
 - modular
 - extensible



<https://twiki.cern.ch/twiki/bin/view/EGEE/YAIM>

- **Tensions**
 - Timely or tested release
 - Stability or functionality
- **Deployment**
 - Automatic deployment of clients
- **Change rate to support production**
 - High, and peaky

- **Release process**
 - Incremental
 - Keep independent things independent
- **Integration points and release model**
 - docs
 - meta rpms
 - config
- **We have a responsibility not a choice to release a component**
- **Software can arrive with no previous deployment history**
- **There is often great time pressure to release**
- **Cannot choose components only on the basis of stability**
- **Conditions for integration (eg use of externals) cannot always be enforced**

- **Certification**

- coverage

- deployment environments

- *batch system support*

- use cases

- *documentation / architecture*

- multiplatform

- one change now requires parallel certification steps

- **Testing**

- Certain things can only be tested at the production scale