What do Web Services bring to Future Grid Architectures?

Roger Menday, Philipp Wieder
Forschungszentrum Jülich, Germany
{r.menday, ph.wieder}@fz-juelich.de

Outline

• Introduction
• Web & Grid Services
• Project GRIP
• Architectural evolution
• Summary
Motivation

**Status Quo:** “I need a Grid Service” -> “I deploy a Grid System”

**Future:** “I need a Grid Service” -> ”I deploy a Grid Service”

- **Usage:** Deployment time vs. usage time ratio is unacceptable (days vs. hours?)
- **Boundaries:** Virtual Organisations (VOs), within VOs (different CAs, ...), Grid systems (including protocols, ...)
- **Market:** Move from batch model to a service/market-oriented approach (Web Services, OGSA, Business Grid, ...)

Next Generation Grid Wishlist

- Transparent and reliable
- Open to wide user and provider communities
- Pervasive and ubiquitous
- Secure and provide trust across multiple administrative domains
- Easy to use and to program
- Persistent
- Based on standards for software and protocols
- Person-centric
- Scalable
- Easy to configure and manage

*So do Web Services help to fulfil this?*
Challenges

- **Integration of Web Services**: Lots of talk about paradigm shift, but this maybe reflects the truth.
- Currently: Layered, often closed architectures, custom protocols. **Move to: standard protocols and web services**
- With WS/GS: Distributed, open (the O in OGSA) architecture, with **standardised Grid protocols**, expressed using the language(s) of web services

**New challenges:**

- Service detection, orchestration, dynamic federation, semantic grid ... and still the old: languages, interoperability, ...

Web Services

*Designed for loosely-coupled distributed computing*

- **WSDL**: service description
- **SOAP**: service invocation

*Extensible SOAP Header to carry additional information*

- **Routing**: WS-Routing
- **Community security**: SAML, WS-Trust, WS-Policy, XKMS

... and then there's **OGSI**

(not really a WS specification - builds on the standard SOAP & WSDL specs)
Open Grid Services
Infrastructure/ Architecture

OGSI also makes web services **stateful** - useful for the kind of services which make up a Grid

OGSI also allows construction of service interfaces from other interfaces - **interface inheritance**

OGSI provides **service data** - remote instance variables for web services

---

Status of standardisation work

Most work taking place at the Global Grid Forum

Some interesting working groups:

**OGSA**: documentation of requirements, functionality, priorities, and interrelationships for OGSA services

**CMM**: define a Common Management Model and a set of OGSI porttypes for the standardised management of resources and services

**OGSI-Agreement**: agreement negotiation for the usage of services according to policy

**OGSA-Sec**: grid service security framework

**GridIR**: information retrieval system on the OGSA Grid - document collection management, indexing/searching, query processing
The Grid Interoperability Project

... to realise the interoperability of UNICORE and Globus and to work towards standards for interoperability in the Global Grid Forum

- Development of an interoperability layer between the two Grid systems
- Interoperable applications
- Contributions made to the Global Grid Forum
- UNICORE towards Grid Services

www.grid-interoperability.org
www.unicore.org

Partners

Two year project funded by the E.U. with the following partners:

- Forschungszentrum Jülich (DE)
- Pallas (DE)
- Deutschen Wetterdienst (DE)
- Fujitsu (UK)
- University of Manchester (UK)
- University of Southampton (UK)
- ICM (PL)
- Argonne National Laboratory (US)

Project completes end 2003.
UNICORE today

- Full control over the jobs through a graphical user interface.
- Multi-system and multi-site jobs with UNICORE synchronising the jobs and staging data
- Secure & co-operates well with firewalls
- Abstraction of system functions, commands, and user actions to achieve system and installation independence. Software Resources. Plugins
- Retain full administrative autonomy at participating centres

What about the cons ??

- Can’t really use it as the basis for exotic Grid applications/services.
- Lack of delegation -> some restrictions

Technical points and issues

- Vertically integrated architecture
- Security based on X509 certificates and ssl. No delegation
- Java based, although Perl sometimes used for target systems
- Abstract Job Object (AJO)
  - Carries a workflow of jobs
  - Jobs described in an abstract form
  - Workflow can also contain some control constructs
Authentication

NJS – Network Job Supervisor

TSD – Target System Interface

NJS <-> TSI Protocol

Indicates SSL transport

USite – UNICORE site

UNICORE Architecture

UNICORE Protocol Layer

UPL

NJS

IDB

UUDB

TSI

Client

Gateway

Multi-site jobs

Abstract

Incarnation

Non-abstract

Authorisation

Architectural Options
Security ...

Biggest influencing factor on the design of Grid architectures?

Providing a delegation mechanism for a resource to access another resource on behalf of a user is both challenging and controversial.

Lots of web service security specifications. Some nice characteristics:

- Message-level security
- Multiple signatures on a document describing a workflow
- SOAP header is extensible to support security, message routing, policy, etc.

Service Virtualisation and Job Abstraction

Increasing the separation between the user and the command line

Service-oriented thinking, not batch

UNICORE well placed already - software resources

Mechanism for this:

- Derive a new ‘software resource’ porttype
- Use operation extensibility of OGSi to parametrise the request of the software resource
A "peer-to-peer" Grid

- Fading distinction between client and server
- Attractive in theory, but not very infrastructure friendly in practice

LANs, Virtual Organisations, etc ...

**UNICORE**

Layered. (a UNICORE 'grid' is a collection of Gateway servers - a UNICORE Usite is a 'mini-grid' under the control of one Gateway?)

It's 'traditional' 3-tier architecture fits better for deployment in a production environment.

- *(Physical)* grouping of Vsites arranged into Usites.

**Globus**

dispenses with 'infrastructure' components. It operates on a Vsite-to-Vsite basis, and assumes that each machine is accessible.

- *(Logical)* grouping of Vsites to create VO structures.
Proposal

What we would like is the best of both worlds

With web services this is easier to achieve in a interoperable and maintainable manner.

i.e. an ‘overlay’ network infrastructure, which gives us the infrastructure friendliness of the UNICORE architecture, with the logical purity of the Globus architecture.

A possible evolution

- A Grid comprising of distributed services
- Virtualisation - the UNICORE software resource concept maps to application specific web services
- Interoperability between different Service providers
- Dynamic higher level services built out of other services. Job workflow, for example, but also viewing security, authorisation, etc, as services, from which other aggregated services can be built
Summary
Summary

- OGSI is a natural direction for UNICORE project to take, and a sound move for the future
- With the advent of OGSA, we view interoperability in a broader sense and not just interoperability with Globus
- A Grid composed of services from multiple (including non-UNICORE) services
- Service aggregation to build the functionality needed

Forms the basis of the current work in GRIP.

Questions, comments, suggestions?

www.grid-interoperability.org
www.unicore.org