One-Stop, Fire-and-(almost) Forget Dropping-off and Rendezvous Point

R. Menday, B. Hagemeier, B. Schuller, D. Snelling, S. van den Berghe, C. Cacciari, M. Melato

UNICORE Summit 2006
2006-08-30

Björn Hagemeier
b.hagemeier@fz-juelich.de
Motivation

- An easy way to access Grid resources (A-WARE)
- Portal solution (easy)
- Domain-specific work assignments
- Automatic grounding of workflow tasks
- Partial, flexible interactivity
Agenda

- **Introduction**
- **Fabric Layer**
  - Atomic services
  - Roctopus
- **Higher-level Services**
  - Workflow
  - Orchestration
  - Notification and interaction
- **JBI**
  - Communication infrastructure
  - Flexible messaging
  - Integration of UAS
- **Summary**
Introduction

- **Unicore**
  - Vertically-integrated, stovepipe
- **Recent extensions to SOA**
  - Loose coupling
  - Stovepipe construction toolkit
  - Fabric layer

- **Higher-level services**
  - Orchestration
  - Non-fabric
  - Workflow, business process or service chain
OSFAAFDOARP explained

- **One-stop**
  - single facade to the grid user
  - drop-off and rendezvous

- **Fire-and-forget**
  - work engine orchestrates workflow
  - 'rendezvous' on completion

- **'almost'**
  - information during runtime
  - participation in execution
Architecture

- Portals
- Workflows
- WOS
- Fabric
Grid Fabric Layer

- **Unicore Atomic Services (UAS)**
- Positioned on JBI bus
- Accessed by (JBI) binding component using Roctopus API

  ➞ Easy programming
  ➞ Simple configuration
  ➞ Other backends possible
Roctopus

- Support for multiple backend infrastructures, e.g. Unicore 5 & Unicore 6
- Hides implementation and configuration details
- Small set of interfaces
- Model of resources resembling REST
Workflow Orchestrator Service (WOS)

- **BPEL as starting point, possibly others**
- **Rule-based reaction to workflow status**
  - Notification
  - Selection of resources (Policies)
  - Corrective reaction to failures
Position of the WOS

- In front of Gateway, tightly coupled to portal
- Completely behind gateway
- Portal in front, WOS behind gateway
Functionality

- Workflowing
- Scheduling
- Brokering
- Negotiating
- Integrating

- Informing
- Interacting
- Securing
- Mediating
- Transforming
Scheduling

- **Static**
  - Completely predefined and authorized by client or user

- **Dynamic**
  - Description of work without resource assignment
  - Automatic assignment of resources according to requirements

- **Hybrid**
Brokering

- Selection of resources
- Match requirements
- Respect user's policies
- Changes of resources during runtime closely tie brokers and schedulers
Informing

- Static and dynamic information
- Filtering
- Transports
  - Email
  - RSS feeds
  - SMS
  - Instant messaging
- Information about
  - Status changes
- User preferences
Interacting

• **Input from user during execution**
  - Approve dynamic resource selection
  - Adjust parameters of execution
  - Monitor progress
Java Business Integration (JBI)

- **Normalized Message Router**
- **protocols and transports**
  - REST
  - WS-*
  - Embedded

- **Multiple implementations of standard**
  - ServiceMix
  - OpenESB
ServiceMix

- JBI implementation
- Many existing components
  - Transport bindings: Email, Jabber IM, RSS/Atom feeds
  - BPEL, Drools
- Several SOAP bindings
- Simple to use API
- Everything is on the bus
Domain Specific Languages (DSL)

- **High-level user view**
  - execution of entire workflows
- **Low-level technical view**
  - execution of atomic services
- **Domain experts can**
  - understand, validate, modify and develop DSL descriptions
- **Mapping down to executable workflows**
Business Process Execution Language (BPEL)

- BPEL and WSRF
- Difficult mapping from BPMN to BPEL
- Deploy once, run multiple times
  ➞ BPEL
  ➞ WSDL of services
  ➞ Deployment descriptor
Rules

- Message Routing
- Initiating status messages
- Drools directly supported by ServiceMix
- Rules for orchestration
Technologies

- Portals (JSR 168)
- BPMN
- BPEL
- JBI (JSR 208)
- Roctopus
- UGS
Summary

• **What's OSFAAFDOARP?**

• **Considered architectural approaches**

• **Flexible support of functional requirements through JBI**
  - BPEL support
  - Rule engine

• **Work assignments in terms of DSL**
  - specialists can work in their domain of knowledge
  - *canned* workflows