



# **An Easy Way to Access Grid Resources**

R. Menday, B. Hagemeyer, C. Cacciari, G. Fiameni, M. Melato, A. Curtoni, S. van den Berghe

<http://www.a-ware.eu>

# A-WARE Project

<http://www.a-ware.eu>



*Contract number*

IST-034545

*Type of project*

Specific targeted research project

*Project coordinator*

CINECA

[a-ware@cineca.it](mailto:a-ware@cineca.it)

*Partners*

CINECA, Italy

Fujitsu Laboratories of Europe, UK

Research Centre Jülich, Germany

NICE S.R.L., Italy

AIRBUS FRANCE SAS, France

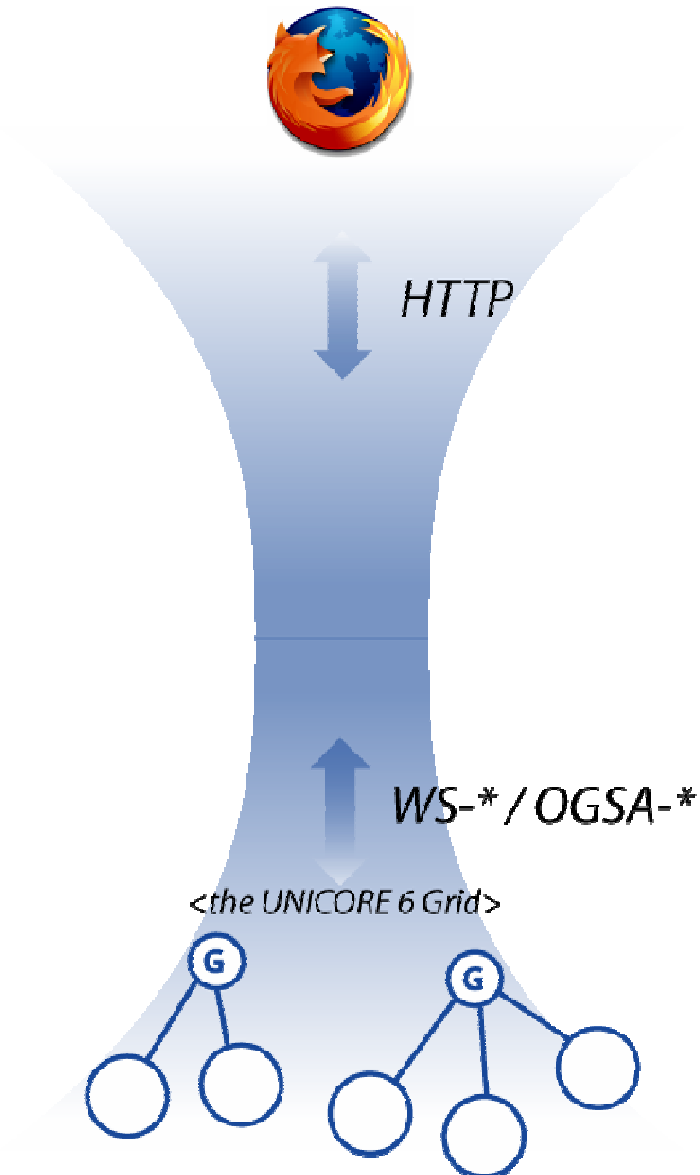
*Project start date*

1 June 2006 – 24 months



**AIRBUS**

# Web access to Web Services middleware



Web based access to the Grid

Bridging HTTP and Web services

Platform for high-level services



# Web-services based Grids

In A-WARE the primary Grid fabric is UNICORE 6.  
(<http://www.unicore.eu>)

Uses regular WS standards.

- Including WS-RF (Web Services Resource Framework)

OGSA (Open Grid Services Architecture)

- Domain interfaces covering atomic Grid functionality
- Formats for Job description, Information models, etc

WS-Security, SAML, etc

# An Easy Way to Access Grid Resources



Simplifying user experience

- Goal-driven. Focus on problems.
- Workflow.
- Easy Grid access.



Fill gap between :

Portal solutions &

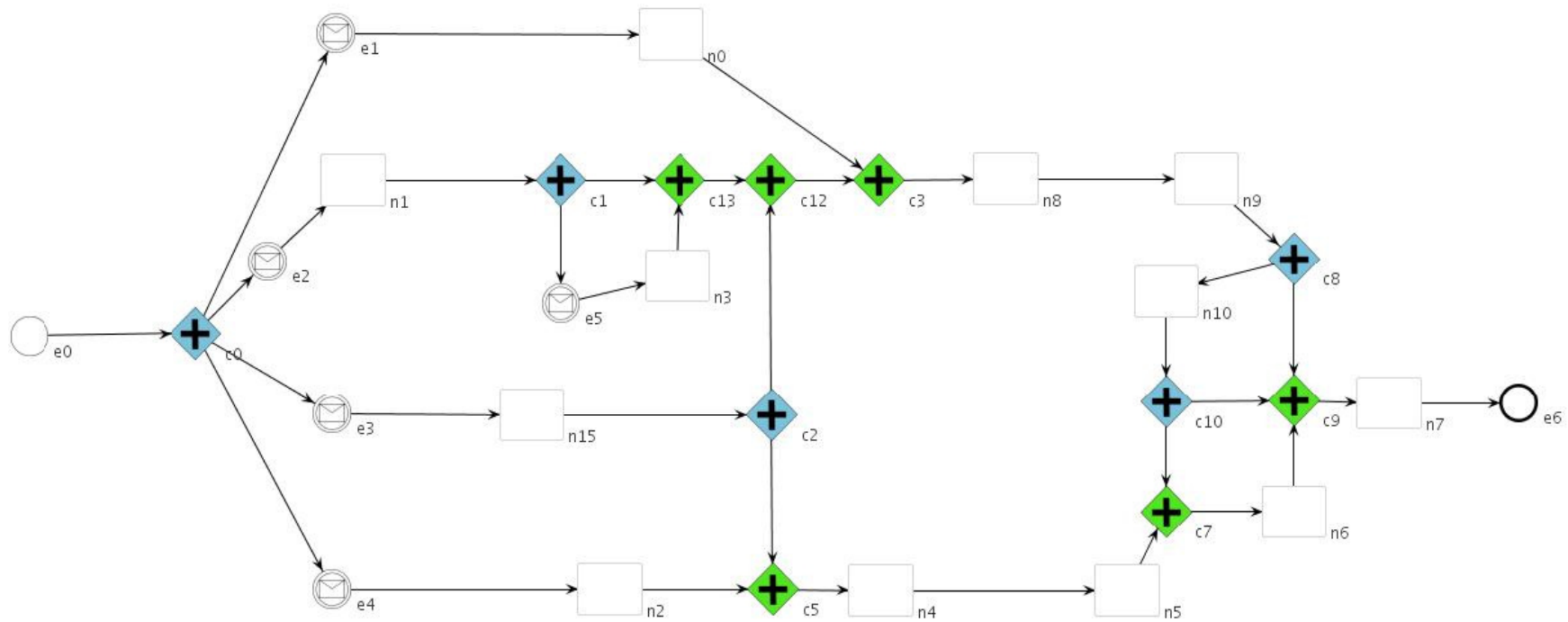
... middleware

Integrating Web services based Grids (UNICORE 6,  
...)



# Use-case Driven

Industrial use-case provided by Airbus





# General Approach

Take the large part of the complexity coded into Grid clients.

- Re-locate to a server
- 'contextualise' access for each user.

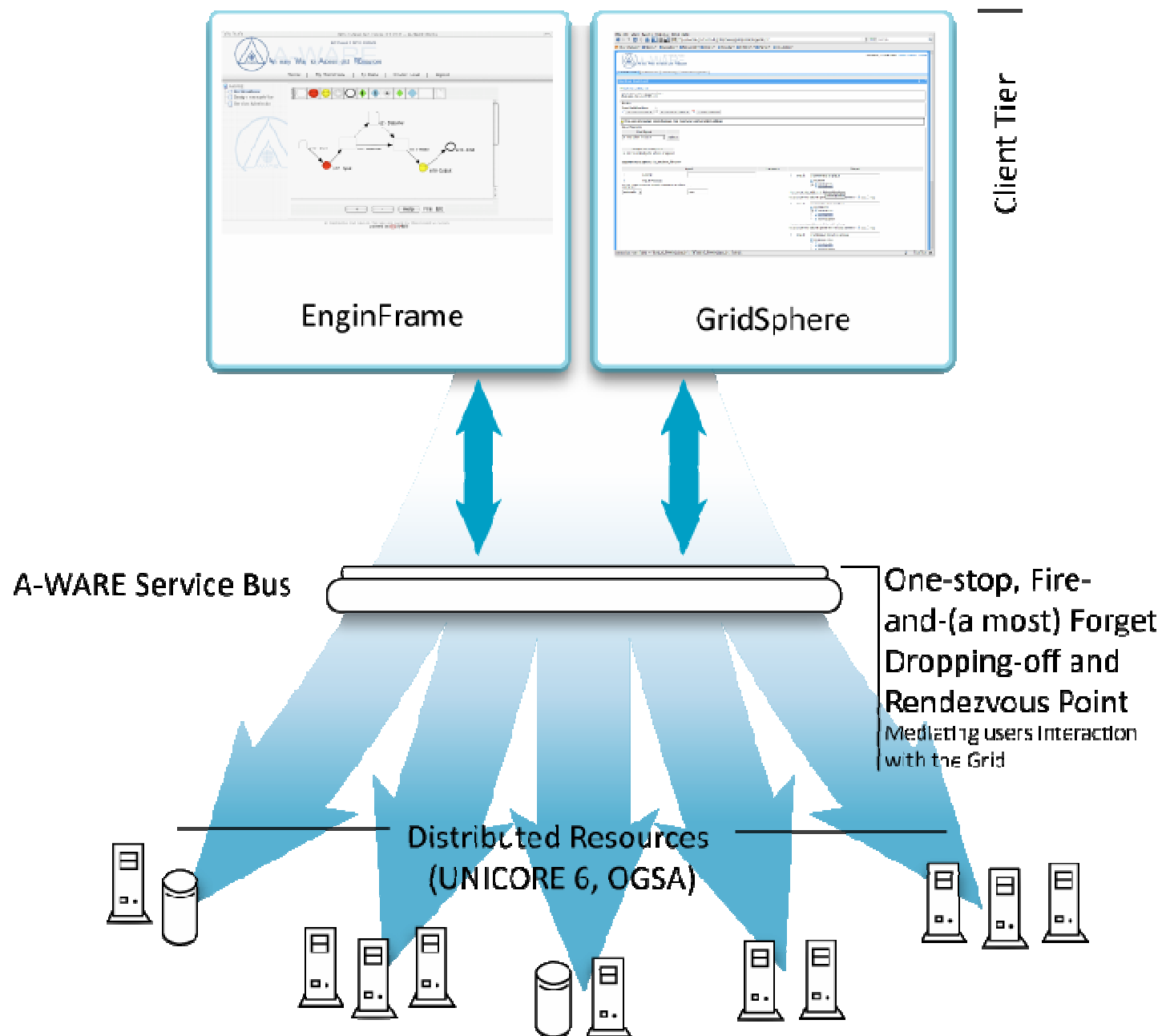
Deploy Portal based solutions around this.

Always-on.

Extensible.

Basis for useful high-level services.

Service Bus platform.





# Java Business Integration (JBI)

Service-oriented computing.

Integration environment.

Component architecture. Normalised Message Router.

Implementation : Apache ServiceMix

Provides an number of 'off-the-shelf' components.

We use JBI to build the A-WARE Service Bus ...



# A-WARE Service Bus

Mediation layer virtualising distributed (Grid) resources together with integration of non-Grid (Local) resources.

Probably located 'near' the user.

## High-level Services Framework

- Workflow
- Search / Query
- Brokering
- Data management
- etc ...

## Platform Services

- Logging, Registry, Monitoring



Homogeneous access to Grid and non-Grid resources through a high-level set of abstractions which drive the ASB 'runtime'.

Manages complexity for user:

- Security complexity
- WS and WSRF complexity
- Un-reliability of distributed resources
- Aligning small, and not-so-small differences



# WSRF <-> Goal-driven

WSRF : system-level protocol framework (?)

-> Quite verbose

Mismatch between this, and goal-driven, user-requirements.

ASB bridges this gap through higher-level formulations



# Normalised Execution Environment

Actions expressed abstractly (script, transfer, etc ... ).

ASB 'grounds' and manages the semi-autonomous execution of these Actions.

Service-oriented virtualising environment.

- 'programmable' Façade over the Grid
- High-level services construction toolkit
- Exposes applications as services
- Interoperation role too

Workflow leverages this ....



# Workflow

ASB architecture specifically designed to support multiple orchestration strategies.

The 'world' not orchestrated directly.  
But rather ...

- High-level, goal-driven services on ASB
- Orchestration of these services
- 'buffer' between the abstract and the 'real'



# 'Kinds' of Workflow

Two kinds :

- Write and run once
- Write, deploy and use multiple times

(consequence of this is that from a consumers point-of-view workflows can become 'atomic' ASB actions themselves, leading to a possibility for workflow of workflows)

BPEL the second kind ...



# BPEL

Business Process Execution Language.

Initially targeting BPEL in A-WARE.

Using the Apache ODE BPEL engine  
... which comes JBI-enabled



# **BPMN**

Business Process Modelling Notation

Modelling processes / workflow in abstract terms

Workflow Designer Application (WDA)

Process of 'grounding' onto executable workflow (BPEL)

- BPMN2BPEL

## **Workflow Repository Service (WRS)**

Repository for the storage and retrieval of workflow definitions.



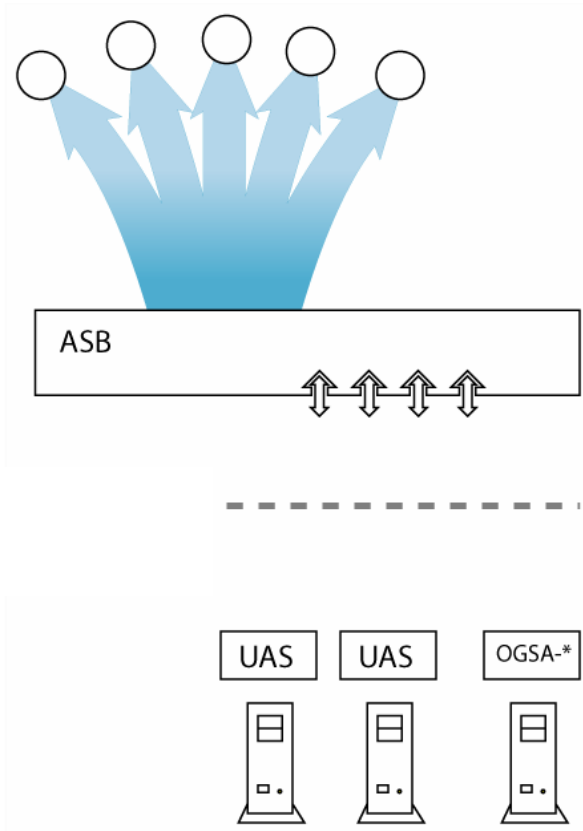
# Client Perspectives

## Communication with the ASB

Encourage a rich ecosystem of client options

Largely HTTP based currently

- RESTful HTTP
- SOAP over HTTP
- Work in progress
  - ATOM / ATOM Publishing Protocol
  - RDF representations
  - XMPP (IM) connectivity
  - etc





# Design

The screenshot shows the 'Data Mapping' configuration window in the A-WARE portal. It is titled 'From message e17 to task n10\_Addition'. The interface includes a 'Preselect tasks using data flow object' section with 'From' and 'To' dropdowns. The 'From' dropdown is set to 'Message' and 'Addition' (with path '/xs:add\_out'). The 'To' dropdown is set to 'Message' and 'Hello' (with path '/xs:heLlo'). There are 'add\_out' and 'hello' buttons below the respective dropdowns. At the bottom, there are 'Add mapping', 'Clear mapping', and 'Done' buttons.

# Mapping

The screenshot shows the 'Mapping' interface in the A-WARE portal. It displays a workflow diagram with nodes: 'e16 Start', 'e17 Input', 'n10 Addition', 'n11 Hello', 'e18 Output', and 'e19 End'. The nodes are connected by arrows, with 'e17 Input' feeding into 'n10 Addition', which then feeds into 'n11 Hello', and finally 'e18 Output'. The interface includes a toolbar with various icons and a 'Help | File | Edit' menu at the bottom.

# Binding

The screenshot shows the 'Service binding' configuration window in the A-WARE portal. It displays a workflow diagram with nodes: 'e16 Start', 'add', 'hello', 'e18 Output', and 'e19 End'. The 'add' and 'hello' nodes are highlighted. A 'Service list' panel on the right contains a list of services: 'airbusTest.localPT0', 'addAndHello.e9\_Incoming()', 'airbus.moveD010', 'hello.hello()', 'airbus.saverf0()', 'add.add()', 'airbus.paam()', and 'airbus.anadd()'. The 'hello.hello()' service is selected. There are 'Bind task' and 'Done' buttons at the bottom.

# Submission

The screenshot shows the 'Service Submission' configuration window in the A-WARE portal. It displays a workflow diagram with nodes: 'e16 Start', 'e17 Input', 'add.in', 'e18 Output', and 'e19 End'. The 'add.in' node is highlighted. A 'Service Submission' panel on the right contains a list of services: 'example.e17\_Input0', 'e17\_Input\_data\_outElement', 'add.in', 'a', and 'b'. The 'add.in' service is selected. The 'a' and 'b' fields are set to '2' and '3' respectively. There are 'View XML', 'Hide', and 'Use edited XML' buttons. Below the XML editor, there is a 'test' button and a 'Submit' button.



# Summary

- Making the distributed world ‘seamless’ (and easier)  
(ASB as mediator)
- General platform for user-focussed, high-level services
- Pluggable ...
  - above** many clients
  - below** multiple Grid fabrics & other resources
  - middle** pluggable workflow, query, etc ...