Open Source SOA Universal Middleware

The open and flexible middleware solution realized and managed by Engineering Group

www.eng.it
The universal middleware for the development of extensible SOA/BPM solutions and the governance of services

- Objectives
- Overview
- Governance
- Architecture
Objectives

Open source solution supporting the realization of various types of distributed middleware:
- Integration
- Orchestration
- Modular applications
- Container of innovative services.

Business view
- **Innovation** – a business opportunity rather than a technological interest
- **Change** – improving the existing code and fostering innovation:
  - The existing applications will grow through the integration of new services
  - The new code will be more widely integrated into consolidated applications.

Technological view
- Play **an active role in transformation and innovation processes**.
- Supporting a **constantly evolving ecosystem of flexible application services**.
Core components (service bundles) for the integration and management of services; for the definition of business rules and the realization of application modules in SOA architectures.

By parametrizing each component, the solution gains a considerable modularity, allowing to meet even the most complex requirements.
It acts as a **SOA dorsal enabling the integration and cooperation of applications**, providing messaging, routing and transformation services, in a consistent and secure way.

It allows advanced users to define a **homogeneous solution for the virtualization of heterogeneous and decentralized services.**
Middleware layer for the **communication among distributed and heterogeneous systems**, adherent to different security and reliability specifications.

**Various use modalities:**
- within an application architecture (even an existing one)
- as communication services and wrapper supporting the ESB / BPM middleware
Overview / Orchestration Middleware (BPM)

- Business Modeling
- Analysis / tuning

Business

Process Modeling

- Standard BPMN
- Business vision
- Technical extension
  (services, security)

Connectors

BPM

Metadata & Historical / real time info

- Various installation modalities
  - as a stand-alone server
  - as an application service
- independent from the BPM Engine
- use of existing authorization policies (i.e. LDAP or DBMS)
Definition of processes, allowing the management of **collaborative services through human tasks**

- **User Notifications** by e-mail, sms, web services...
- **Various use modalities**:
  - web worklists, which can be recalled by any web application
  - API Java and Web Services

- Real-time iteration between users/processes and external systems
- Iteration through web forms and/or PDF documents
Overview / eForm lifecycle middleware

- Automatic generation of documents (also in PDF) and management of their entire life-cycle through processes

- Creation of forms
- Integration of forms with external document management systems
- Status management through specific business components (i.e. digital signature, certified electronic mail)
- More value to business knowledge and management of a company policy repository.

The use of a system based on BPM and of a rule engine does not increase the system complexity: it provides it with more flexibility and makes analysts’ activities easier.

- The use of rules that are externally defined by the application components facilitates the maintenance and adaptability of the services to new business requirements.
The Web is no longer a collection of static pages of HTML that describe something in the world. Increasingly, the Web is the world – everything and everyone in the world casts an "information shadow," an aura of data which, when captured and processed intelligently, offers extraordinary opportunity and mind bending implications. (O'Reilly & Battele, 2009)

…. From the **Internet of things** (IoT), a neologism referring to the extension of the internet to the world of objects and to the user-object interaction model.

**The challenge consists in realizing a solution that implements:**

- **Virtualization of information systems** – No longer just hierarchical information structures. Each service is considered as an application made of data to be exported, integrated and shared.
- **Mashup of user services** – It supports the integration of all functionalities within a service, instead of using simple links to services suppliers, as happened with the “2.0” applications.
- **Independence from devices** – An application/a service must exploit at best the device on which it is installed: geo-referencing, camera/barcodes, resources (i.e. address book, documents, photos).
- **User-system interaction, based on the models used by the best-known social networks.**
## Governance / Projects definition

<table>
<thead>
<tr>
<th>User Interface</th>
<th>Processes &amp; Rules</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architect/Developer</td>
<td>Analyst</td>
<td>Architect/Developer</td>
</tr>
<tr>
<td>User Interface Design</td>
<td></td>
<td>New Components Design</td>
</tr>
<tr>
<td>Widget Research from Provisioning Registry</td>
<td>Component Research from Provisioning Registry</td>
<td></td>
</tr>
<tr>
<td>Ajax UI Development</td>
<td>New Components Development</td>
<td></td>
</tr>
<tr>
<td>Business Logic UI Development</td>
<td>Publishing Configuration</td>
<td></td>
</tr>
<tr>
<td>Publishing Provisioning Registry</td>
<td>Technical Completion BPMN</td>
<td></td>
</tr>
<tr>
<td>Proxy Configuration</td>
<td>Rules Definition (rules engine)</td>
<td></td>
</tr>
<tr>
<td>Release Application (WAR)</td>
<td></td>
<td>Publishing Provisioning Registry</td>
</tr>
</tbody>
</table>

Governance
Governance / Relationship among the involved actors

GOVERNANCE PORTAL

- Administrative Governance
- Services Governance
- PM
- Knowledge Management
- Infrastructure Governance
- Application Governance
- Security Governance

SERVICE CONTRACT MANAGER

CLIENT REFERENCE PERSON

SERVICE/APPLICATION USERS

OPERATIONAL STAFF
Governance / Tools

- Tools supporting the **efficiency of all the involved actors.**

- **Tools IDE**
  - Services Implementation
  - Registry
  - Rules Management
  - Process Definition & Implementation
  - Deploy & Versioning

- **Web console**
  - Administration
  - Management
  - Monitoring
  - JMX

- Metadata & Historical / real time info

- Analysts
- Architects
- Developers
- Managers
Governance / Monitoring & Management

- The information is available in **real-time**
- The information can be used **from different points of view**

- Monitoring
- Management
- Business Intelligence & BAM
Governance / Monitoring & Management

- Real-time evaluation of the process and service lifecycle
- Search for a service/process, also using relevant applicative information
- Navigation through correlated activities
- Visualization of interchangeable data (messages, documents)
- Errors management, also allowing to re-execute a blocked service or process.
It allows to share information with Business Intelligence systems, in order to quickly get dashboards monitoring one’s own operational and business indicators, even in real-time.

- Definition of one’s own KPIs
- Dynamic graphs
- Reports
- Drill-Down and Drill-Up
- Filters
- Aggregated documents

Spagic MetaDB data schema, shared with eBAM / SpagoBI
A new approach for innovative models

From the applications cooperation through ESB
Clear organizational separation of:
- Applications
- Integration infrastructure
- Specific governance services

To the outsourcing based on the XaaS pattern
Everything is based on distributed nodes to provide specialized services:
- SaaS Software (front-end & back-end)
- PaaS Platform
- CaaS Communication

Cross Governance
"Software that you write once and can use in binary form universally: in many different platforms, many different industries, and for many different purposes." Peter Kriens *(OSGi evangelist)*

**OSGi Objectives**
- Adherence to SOA
- Modularization
- Management of components life-cycle
- Dynamic runtime

**Spagic Universal Middleware Objectives**
Enterprise SOA Framework made of components allowing to realize a modular and configurable Universal Middleware around an OSGi kernel.
OSGi technology supports the dynamic management of bundles for the realization of modular application services.

**Fundamental Layer**

- *Execution Environment* – middleware execution
- *Modules* – software modularity and dynamicity
- *Life Cycle* – service lifecycle management
- *Registry* – service registry and search
- *Services* – service container
- *Security* – security management services
Services, connectors and modules are implemented into the Component Model pattern
- Each component can provide new services while referencing other ones
- The dynamic behaviour of the components is guaranteed by the OSGi Declarative Services.
Architecture / OSGi bundles development – reuse – deploy

- Reusing Jar Library
  - Jar Library
  - Transformation
  - Develop new component
  - New Requirement
  - OSGi Bundles
    - Implements Spagic Interface
    - Config Publishing rules
    - Deploy
    - Spagic Bundles
    - Transformations
    - Registration Spagic MetaDB
      - Spagic Service Manager...

Supported by Eclipse Visual Tools

www.spagic.org
Creative Commons Attribution-Share Alike 3.0 Unported License
Services interact with each other through messages
- **ECF Distributed EventAdmin Service**: Eclipse project supporting the communication among services distributed on different nodes.
Architecture / Message Bus & BPM

- **BPM layer**
  - **Start1**
  - **Process:**
    - **ConfirmOrder**
    - **CreditCheck**
    - **Router1**
    - **Parallel1**
    - **PrepareInputForStoreProcess**
    - **InvoiceGeneration**
    - **CrmNotifier**
    - **Parallel2**
    - **End1**

- **Message Bus layer**
  - **Normalized Message Router**
  - **OSGi Event Admin**

**Normalized Message Router**
- it manages the exchange of messages among components.
- it is internally based on the OSGi Event Admin Service Communication.
BPMN2 modelling
- Orchestration of application modules (OSGi Bundles) and external systems (OSGi Connectors)
- Deployment on BPM engine (JBPM OSGi bundles, Apache ODE BPEL Server)
Presentation Pattern: from MVC to MVP
- Clear separation of user services (VIEW & PRESENTER) and backend services
- Widget libraries (i.e. GWT)
- High browser compatibility
- Support to W3C ARIA accessibility specifications
- Support to internationalization
Technology architecture / SOA Application

- It implements the business logic through command patterns.
- Configuration of the interaction with the other components – Application Composition.

Application Node (es. JBoss o Apache Tomcat)

- Separation between the front-end and the sole backend layer
- We can switch from a remote modality to a VM one without modifying the code
Technology architecture / Business Rules management

Definition of rules supported by a specialized editor or by tables on a spreadsheet (Excel or OpenOffice).

It defines the set of “concepts” (entity, property, relations) of a specific domain (rule), supporting the requirements of flexibility and modularity of SOA applications and BPM processes.
A **listener** (Eclipse Event Admin) allows to gather the information that are produced by the middleware nodes and afterwards shared among the monitoring tools.