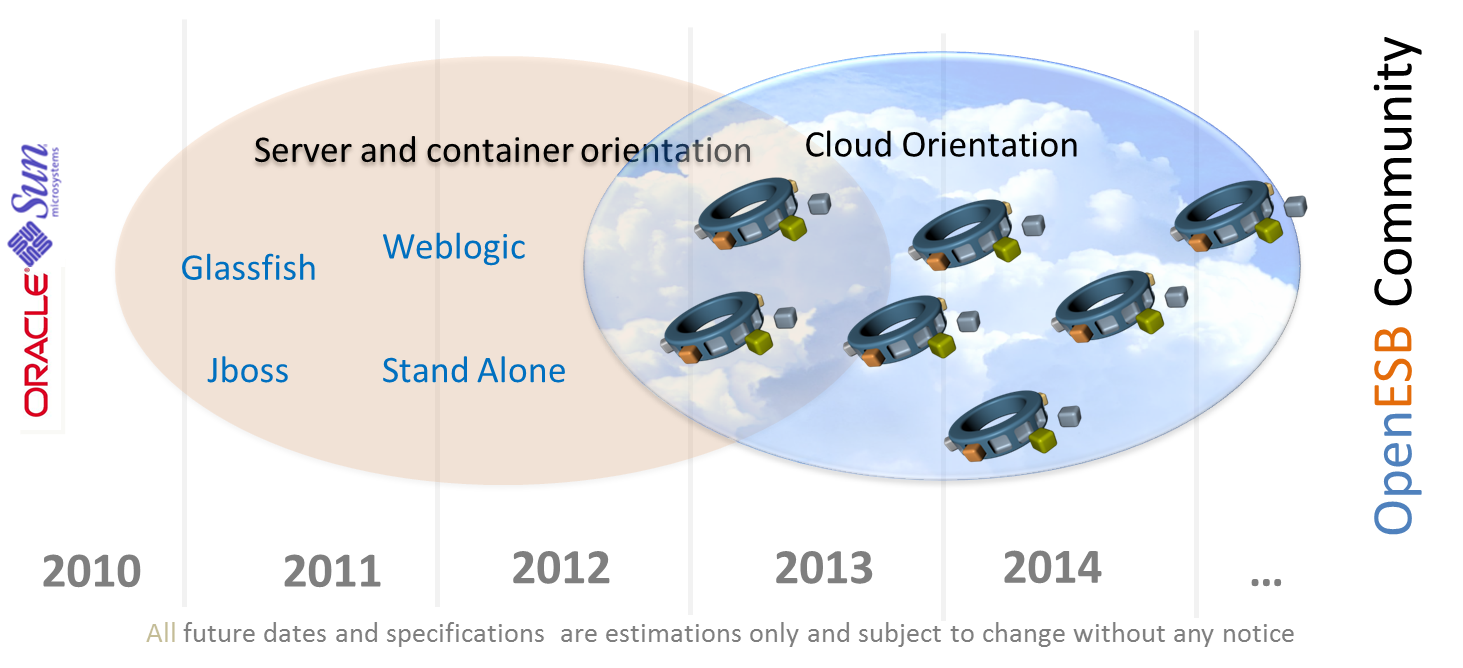
|  |  |  |
| --- | --- | --- |
| OpenESB community project | | |
|  |  |  |
| Tittle | New Monitoring tool | |
| Initiated by | Paul Perez Pymma consulting: | |
| Contact | [paul.perez@pymma.com](mailto:paul.perez@pymma.com) | |
| Starting Date | 20/01/2012 | |
| Planed for: | Estimation : Mid 2012 | |
| Version | 0.1 | |

|  |  |  |
| --- | --- | --- |
| Action by | Date | Comment |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Context

From one side, since many months, OpenESB community teams work on new platforms for OpenESB. Next OpenESB generations will be run on additional containers like GFV3, JBoss and JVM. It is easily understandable that the classical Glassfish v2 console cannot be used on other platforms. Decoupling the monitoring tools from the platform where OpenESB runs is a necessity to manage multiplatform deployments. On the other side, current OpenESB deployment API is not ready for cloud deployment where scalability and high availability is provided by infrastructure and not by container capability. Next OpenESB generation must be able to support cloud deployment.



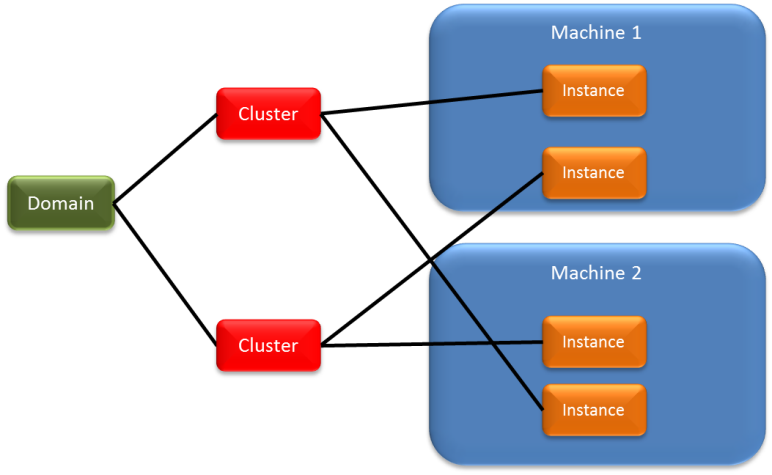
Today OpenESB runs on JBoss, GF3 and standalone JVM but in Alpha version

# Project

It is the reason why we would like to design and implement a new API implementation for deployment and monitoring and an independent tool for developers but mainly for production and support teams.

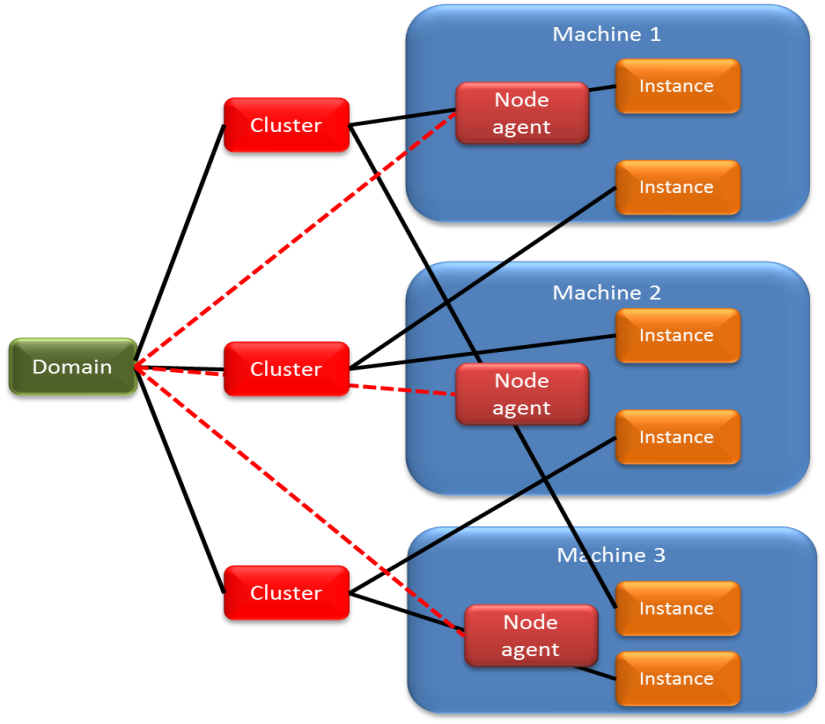
# Architecture

At present scalability and high availability rely on Glassfish hierarchy domain, clusters and instances.



As explain below this hierarchy cannot be used for heterogeneous container. So if we don’t use a Glassfish infrastructure, however we can work with 4 independent instances.

Majority of OpenESB components are ready for multi instances configuration and can work in this context without modification. Ex: multiple File BC can work together on the same directory by using internal locking.

However, it is not possible to let the instance completely independent for many reasons. The first is the difficulty to deploy the same configurations and the same applications on multiple instances. Glassfish Clusters relies on Node Agent to deploy an application on multiple instances and synchronise the instances when a parameter has been modified.

Node agents are Domain delegates which synchronize instances with the domain. For more details on Glassfish V2 Node agent, please have a look on Glassfish Administrator guide.

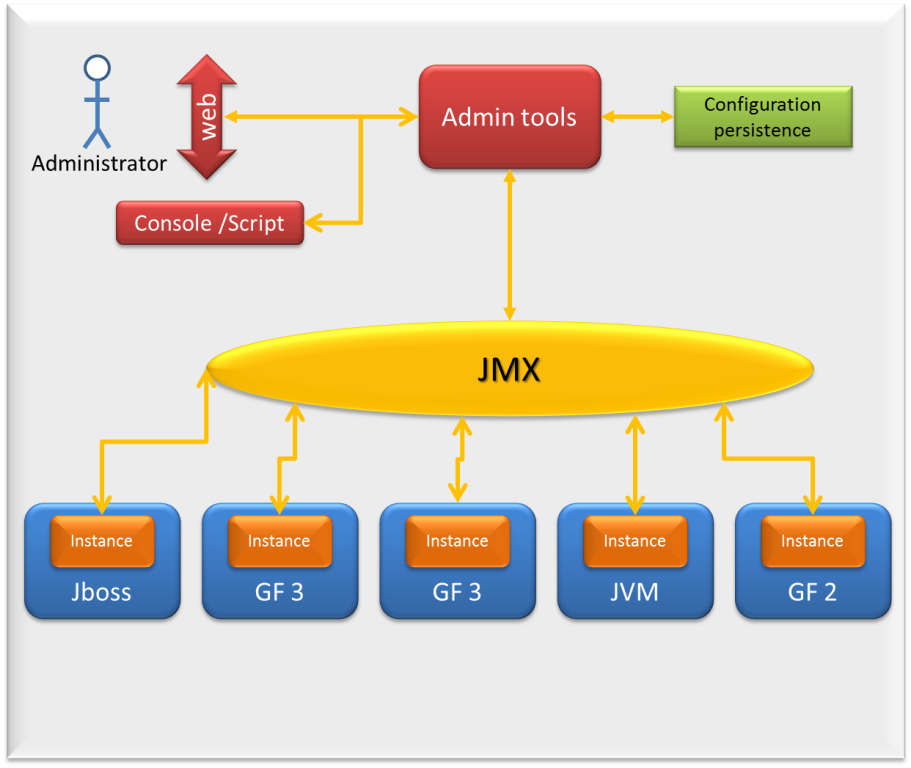
The second issue comes during application deployments. It is not possible for a production teams to deploy the same application N times on M machines. In deep, without cluster infrastructure OpenESB administrator are forced to deploy manually the same application on each OpenESB instance. No production team would agree to deal with this overload and no management will sign any budget for such complex tool.

# New administrative tools

If we want to promote OpenESB in the cloud architecture, we have to design and develop a new administrative tool in place of GF V2 console. In this chapter, I’ll try to summarize the architecture and the main features of this tool. Everyone is free to comment, criticize the document and feel free to propose and impose your point of view.

## JMX

By design, JBI on which rely OpenESB uses JMX to administrate the implementation and the component. (See JBI specifications Chapter 6). Communications between the administrative tools and JBI entities are defined by JMX framework and since JBI framework and component embedded JMX, we don’t have to use container (JBoss,GF) JMX implementation. So we decouple a bit more OpenESB from Containers



## Tools client

For the admin tools, I propose to develop both a web interface for Human administrator work and a console (Asadmin like) to run script. May be more sophisticated interface can be designed like WS or else. Please if you have any ideas send us it to us.

### Web interface

As explain above, we have to provide deployment, administrative and monitoring features comparable to GF V2 console. This pushes us to provide a war package to run it on any application server. What framework can be uses to develop the web interface has not been fixed yet ( your proposition are welcome).

### Console

It is a Java standalone application able to run command line or script.

## Configuration Persistence

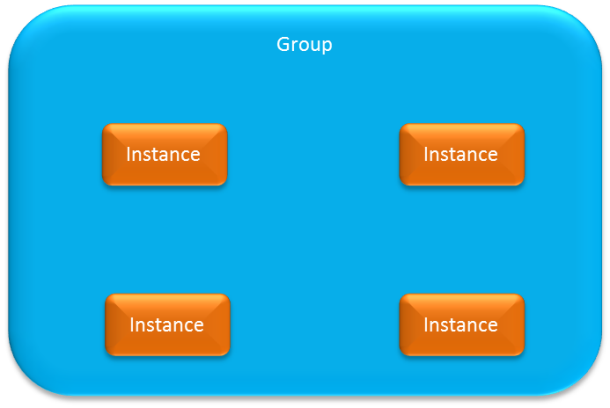
Configuration persistence is the way to keep instance configuration somewhere. I propose to use a classical xml file. It does not allow distributed access but there is no need for that AFAIK. Using xml file allows us to define an xml schema for the configuration file.

# The group

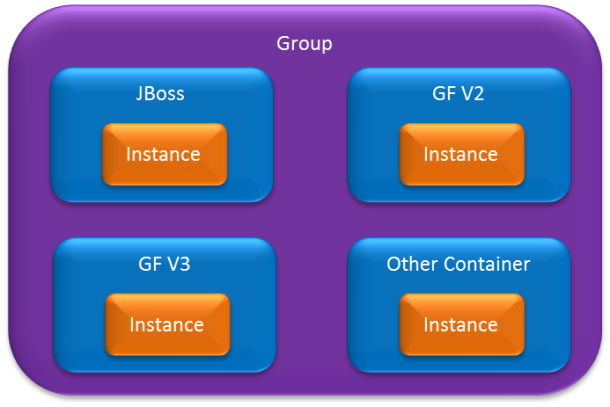
This architecture is simple and efficient but it does not solve the multi instances deployment issue. How, an administrator could deploy the same application on many instances. How can he be sure that the instances running an application ‘X’ have the same configuration? Moreover, if we need more scalability, how can we set up a new instance and deploy the application ‘X’ on it easily.

In order to create a link between instances working on the same context, I propose to create a new entity for our configuration and deployment process. It is the ‘Group’.

The group is a virtual instance of OpenESB with a virtual configuration. By default an OpenESB instance is seen as “standalone” but if we want to associate it with other instance it must belong to a group.

By definition, the instances of the group have the same configuration then host and run the same applications.

Group must be a facility for support and production teams. Modification on configuration and deployment are made at the group level and applied to the instances of the group.

The main difference between the group and the cluster is that the monitoring tools address the modification directly to the instance and not through a node agent. So we don’t rely on container infrastructure to propagate the modification. Since the group can contain OpenESB instances hosted in different containers. Again, container infrastructure and configuration has few impacts on the group management. But some container basic parameters which have a huge influence on instances are beyond the scope of the tool. Ex: Memory size available to run the instance cannot be managed at the administrative tools level. That will be a drawback if we compare to the Glassfish console V2. May be a future version can address this issue by managing instances and containers.

Group has two main functions first it simplifies deployment to many instances the second is to synchronise or propagate instances configuration. How the monitoring tools will play its role and what technologies or API we will use it not fix yet.

Groups configurations are stored in the Configuration persistence. As it exists in many application server, the configuration I can be stored in a XLM file. I’m not a XML schema expert but Configuration schema can look like that:

<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://openesb.org/openesb/schema/"

xmlns:tns="http://openesb.org/openesb/schema/"

elementFormDefault="qualified">

<xsd:complexType name="InstanceComplexType">

<xsd:sequence>

<xsd:element name="name" type="xsd:string"/>

<xsd:element name="description" type="xsd:string"></xsd:element>

<xsd:element name="adminAddress" type="xsd:string"></xsd:element>

<xsd:element name="adminPort">

<xsd:complexType>

<xsd:sequence/>

</xsd:complexType>

</xsd:element>

<xsd:element name="adminLogin" type="xsd:string"></xsd:element>

<xsd:element name="adminPassword">

<xsd:complexType>

<xsd:sequence/>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="GroupComplexType">

<xsd:sequence>

<xsd:element name="name" type="xsd:string"></xsd:element>

<xsd:element name="login" type="xsd:string"></xsd:element>

<xsd:element name="password" type="xsd:string"></xsd:element>

<xsd:element name="Instances" type="tns:InstanceComplexType" maxOccurs="unbounded"></xsd:element>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="Groups">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="Group" type="tns:GroupComplexType"></xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

# What next

What are the next tasks to carry out this project?

|  |  |  |
| --- | --- | --- |
| Name | Description | Comment |
| Publish this document | First step of course |  |
| Get community feedback | Put together feedbacks, comments new ideas… |  |
| Validate project | Get Agreement from OpenESB committers |  |
| Name Architect and project manager | Find the “poor” guys that will deal with the project ;-) |  |
| Create project detailed specifications | Write a precise document on specification architecture technologies used … |  |
| Resource evaluation | Find the good guys for the project (you may be) |  |
| Specification validation | Specification get a Go NoGo from the community |  |
|  |  |  |

This document is a first draft to initiate the project. Please send your feedback on