

# OpenCloudware

## The Cloud application lifecycle management platform

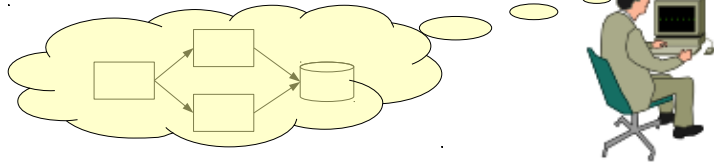
*Delivering a Cloud Platform for Building, Maintaining and Operating  
Enterprise PaaS Distributed Applications*

**Thomas Debru, UShareSoft**  
**Alexandre Lefebvre, Orange Labs**

# What is OpenCloudware?

- Co-funded collaborative R&D project, 18 partners
- Modular software framework for building a IaaS-agnostic Dev to PaaS enterprise-grade Cloud platform
- Enables to easily build, generate and operate enterprise distributed applications for deployment on any Cloud
  - Their modeling, assembly and build
  - Their deployment and operation (PaaS)
  - On multiple infrastructures (Multi-IaaS)
- 3 years (Jan 2012 – Dec 2014)
- Supported by the French FSN (Fonds National pour la Société Numérique)
- Co-labeled by the Minalogic, Systematic and SCS Pôles de Compétitivité

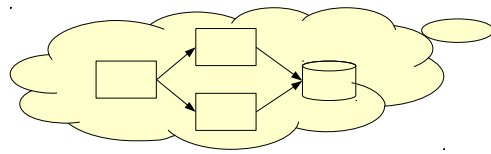
- Large Corps
  - Bull
  - France Télécom – Orange
  - Thales Communications & Security
  - Thales Services
- SMEs
  - ActiveEon
  - eNovance
  - eXo Platform
  - Peergreen
  - Linagora (prev. Petals Link)
  - UShareSoft
- Academic
  - Armines/Ecole des Mines de Nantes
  - IRIT – INP Toulouse
  - Télécom Paris Tech
  - Télécom Saint Etienne
  - Univ. Joseph Fourier
  - Univ. Savoie - LISTIC
  - Inria (Grenoble, Nice, G5K)
- Open Source organisation
  - OW2



OpenCloudware Self Service **Portal**

The user wants to deploy a distributed enterprise application (JavaEE, LAMP, ...)

First step: account creation and login (role-based access control)



## Use case

vApp  
component1  
component2  
component3  
SLA  
bindings

OpenCloudware Self Service **Portal**

**Template Gallery:**  
AaaS, JavaEE, OSGi,  
Services

End to end Cloud **Modeling**  
Meta Data and Modeling Tools

The user builds its virtual app using OpenCloudware tools. S/he expresses elasticity at the right level.

The output is a vApp model (OVF++)

# Use case



```
vApp
component1
component2
component3
SLA
bindings
```

OpenCloudware Self Service **Portal**

**Continuous Build,**  
Maven, Hudson ...

**Service Plan  
Builder (vApp++  
Creation &  
Configurations)**

**Template Gallery:**  
AaaS, JavaEE, OSGi,  
Services

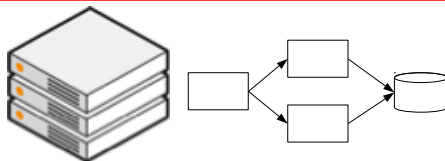
**PaaS Orchestration  
Service Logic Install  
& Deployment**

Sky Computing ( **Provisioning** )

VM Placement      VM Multi-Cloud Provisioning

**Multi-Cloud IaaS Controller (API)**

**Federation Layer**



End to end Cloud **Modeling**  
Meta Data and Modeling Tools

The vApp is built,  
and  
OpenCloudware  
instanciates the  
user vApp on the  
infrastructure.

## Use case



vApp  
component1  
component2  
component3  
SLA  
bindings

OpenCloudware Self Service **Portal**

**Continuous Build,**  
Maven, Hudson ...

**Service Plan  
Builder** (vApp++  
Creation &  
Configurations)

**Template Gallery:**  
AaaS, JavaEE, OSGi,  
Services

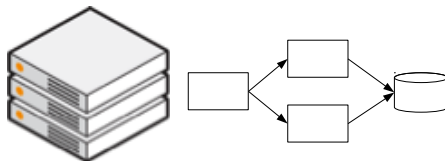
**PaaS Orchestration**  
Service Logic **Install  
& Deployment**

Sky Computing ( **Provisioning** )

<b>Billing Logs</b>	VM Placement	VM Multi- Cloud Provisioning
---------------------	-----------------	------------------------------------

**Multi-Cloud IaaS** Controller (API)

Federation Layer



**Billing**

End to end Cloud **Modeling**  
Meta Data and Modeling Tools

The users needs to know how much it will cost.

The OpenCloudware billing component shows the cost for the empty running vApp.

## Use case



vApp  
component1  
component2  
component3  
SLA  
bindings

OpenCloudware Self Service **Portal**

**Continuous Build,**  
Maven, Hudson ...

**Service Plan  
Builder** (vApp++  
Creation &  
Configurations)

**Template Gallery:**  
AaaS, JavaEE, OSGi,  
Services

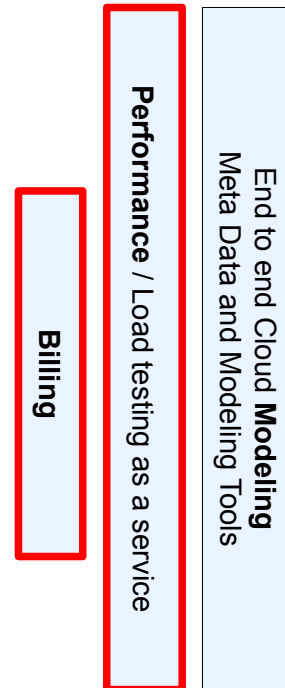
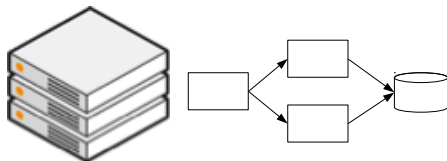
PaaS **Orchestration**  
Service Logic **Install  
& Deployment**

Sky Computing ( **Provisioning** )

<b>Billing Logs</b>	VM Placement	VM Multi- Cloud Provisioning
---------------------	-----------------	------------------------------------

**Multi-Cloud IaaS** Controller (API)

Federation Layer



The users uses  
OpenCloudware  
Performance  
testing tools.

The  
consequences can  
be seen through  
the billing tool,  
and is given a cost  
per hour for a  
given workload.



# Use case



vApp  
component1  
component2  
component3  
SLA  
bindings

OpenCloudware Self Service **Portal**

**Continuous Build,**  
Maven, Hudson ...

**Service Plan  
Builder (vApp++  
Creation &  
Configurations)**

**Template Gallery:**  
AaaS, JavaEE, OSGi,  
Services

**PaaS Orchestration  
Service Logic Install  
& Deployment**

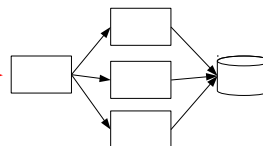
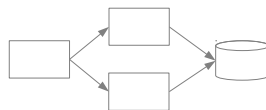
**Self-Management  
PaaS Runtime  
Self-Controller**

Sky Computing ( **Provisioning** )

Billing Logs    **VM Placement**    **VM Multi-Cloud Provisioning**

**Multi-Cloud IaaS Controller (API)**

**Federation Layer**



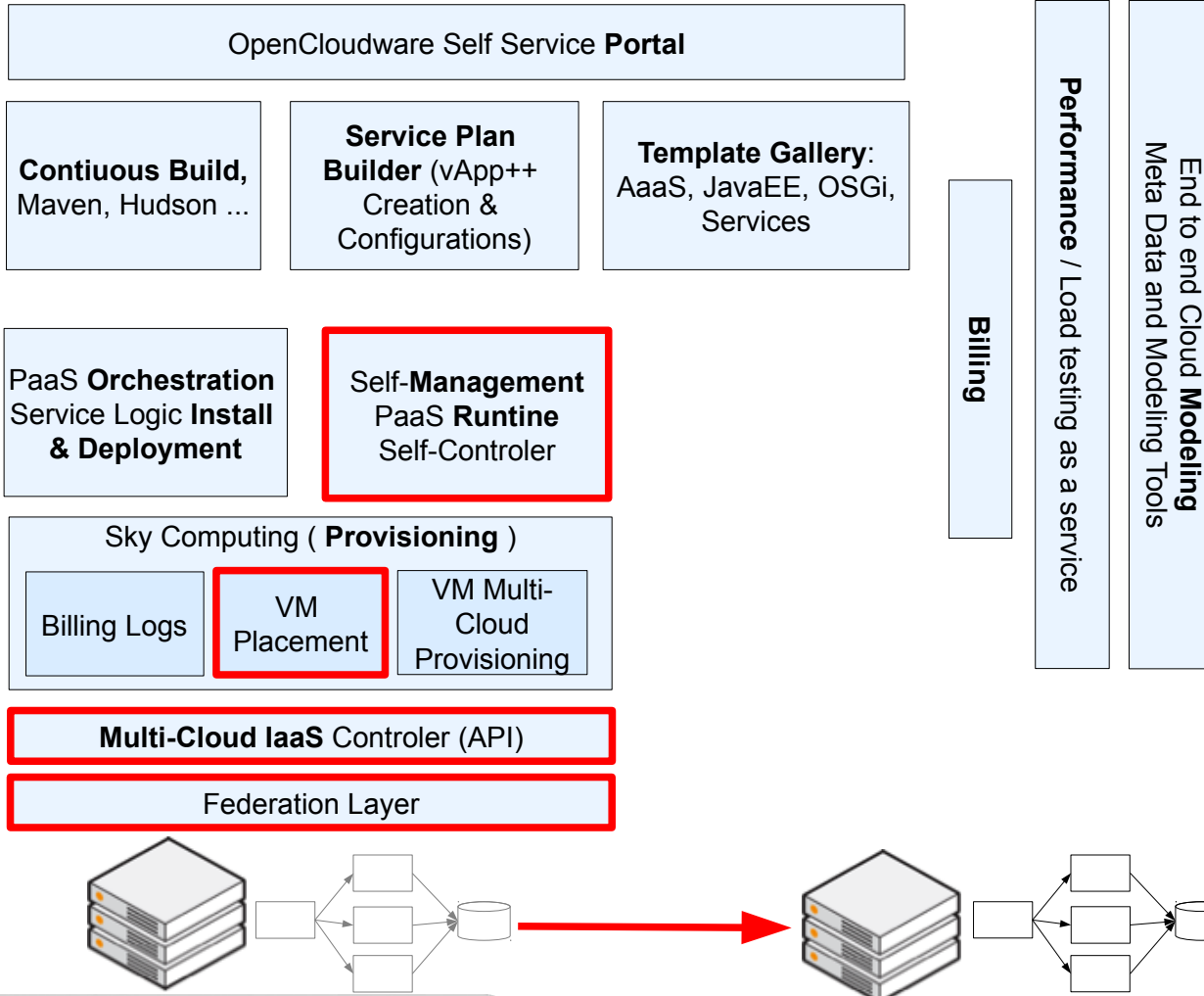
**Performance / Load testing as a service**  
 End to end Cloud **Modeling**  
 Meta Data and Modeling Tools  
**Billing**

While the performance testing tool is running, the user observes elasticity (the application goes beyond what 2 nodes could have done).

# Use case



vApp  
component1  
component2  
component3  
SLA  
bindings



Billing

Performance / Load testing as a service

End to end Cloud Modeling  
Meta Data and Modeling Tools

Testing is over.  
The user needs a public IP, modifies the vApp description and redeploys.

He then adds data to the DB, tests the performance with this public instance.

Latency changes, which indicates OpenCloudware has moved VMs to IaaS closer to users.

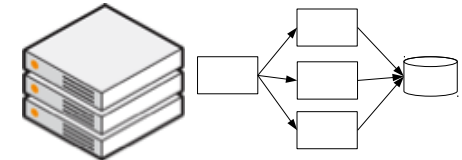
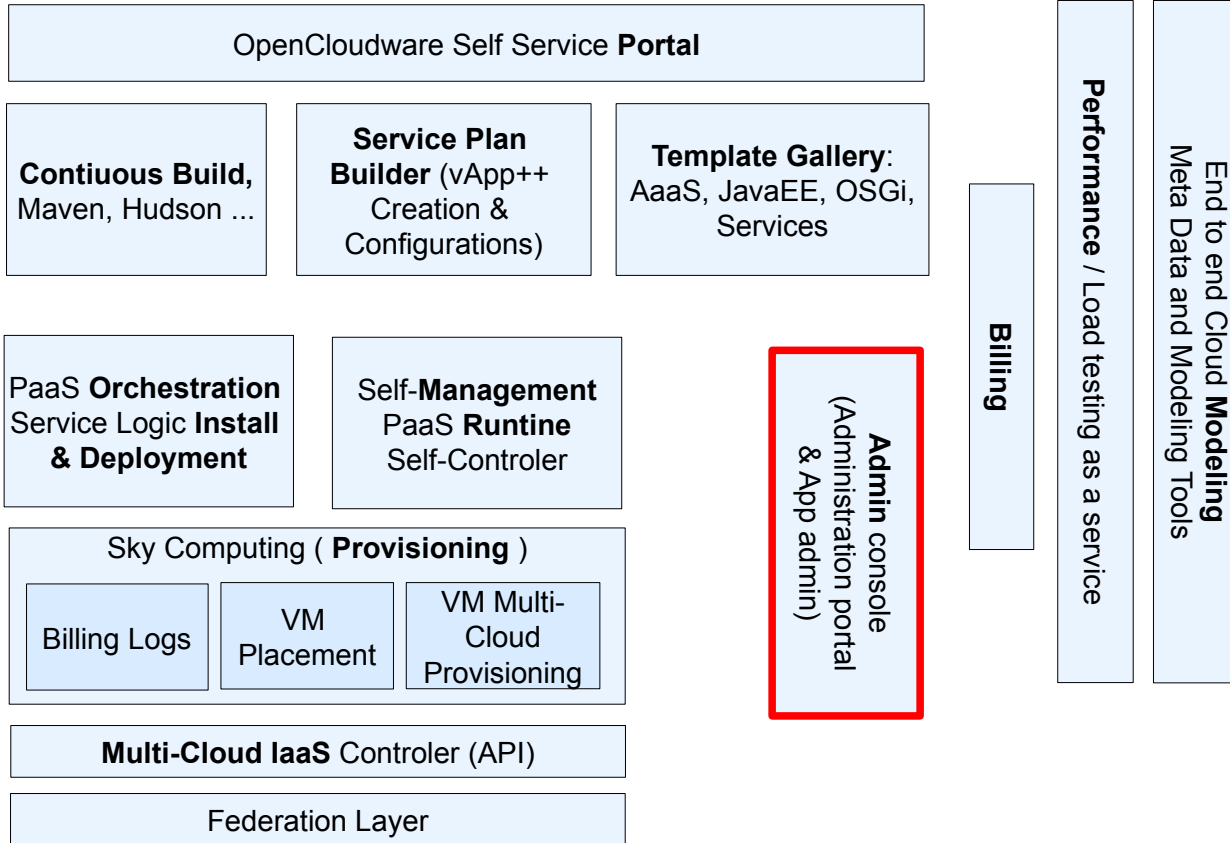
# Use case



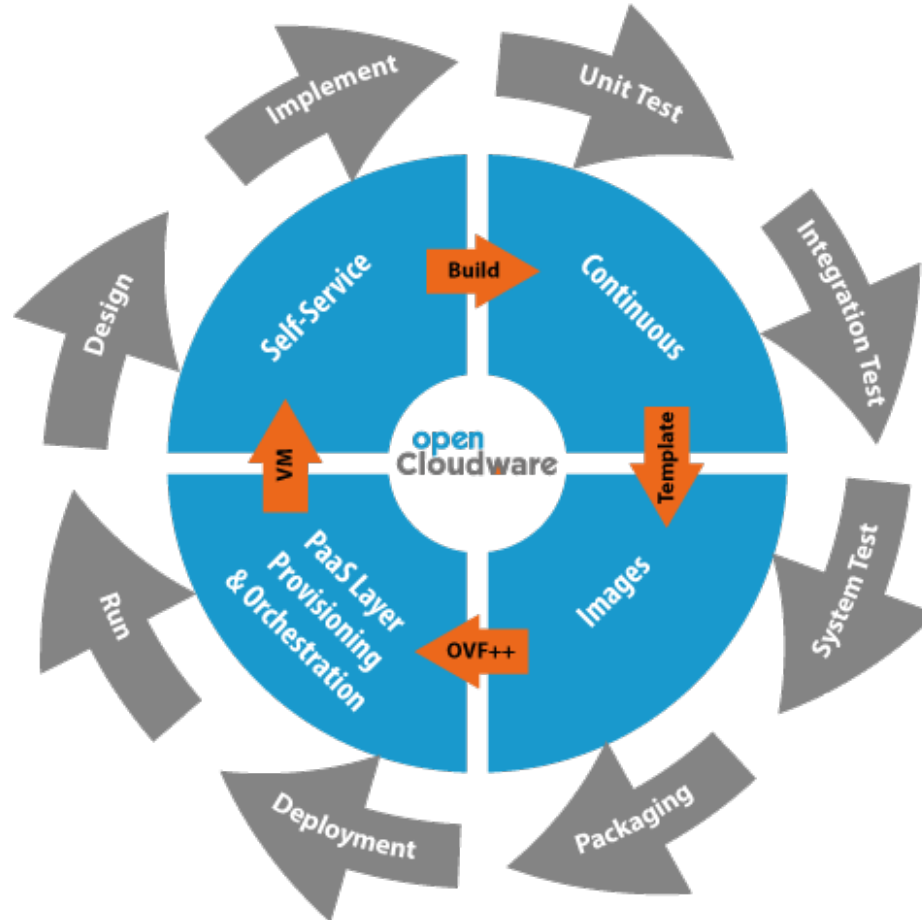
vApp  
component1  
component2  
component3  
SLA  
bindings

The application goes to production.

OpenCloudware monitoring tools provide the necessary information.



# End-to-end Platform Dev to Run



*OpenCloudware Software as a Service*

# Project Target

- Produce a component-based middleware platform
  - open platform for cloud software engineering
  - for collaborative development of distributed Cloud applications
    - targeting primarily enterprise JavaEE - OSGi virtual appliances but extensible to other middleware
  - for IaaS-agnostic Cloud Server application provisioning and deployment, orchestration and operation
    - self-service management, elasticity, green IT optimisation
  - with interoperable execution on multiple major Cloud IaaS (incl. hybrid)
- The OpenCloudware platform will be available online as a SaaS

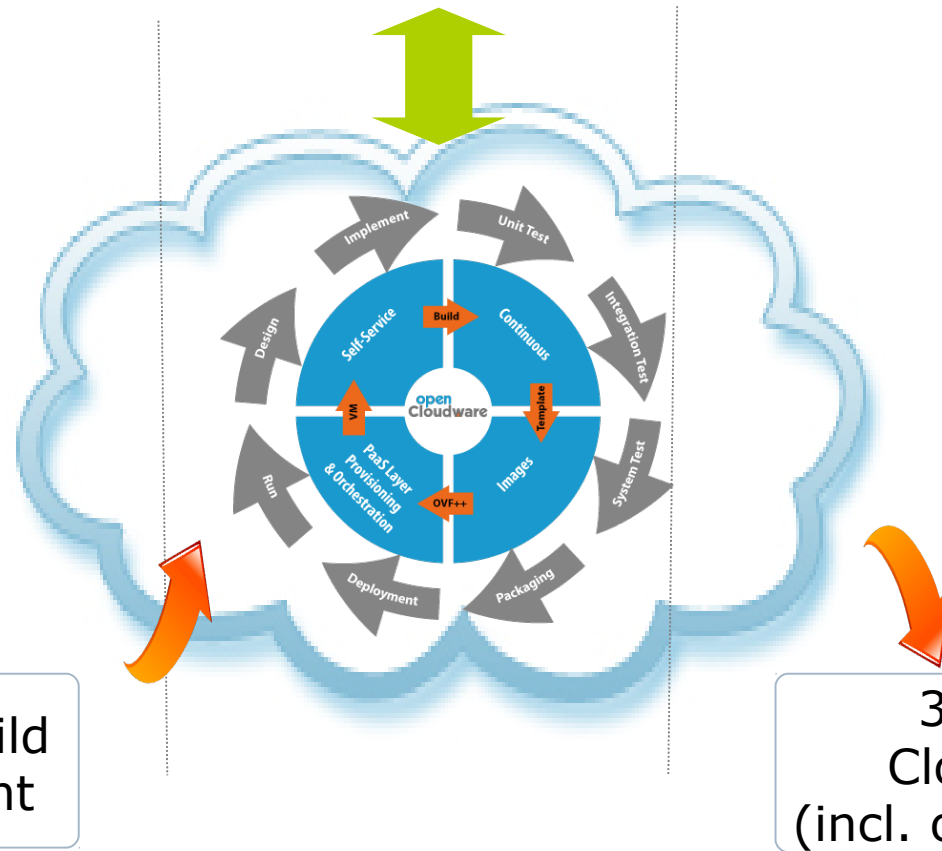
# Project Scope

OW2  
projects

OW2  
Open Source  
Cloudware  
Initiative

CompatibleOne

+ other collaborative  
projects



3<sup>rd</sup> party build  
environment

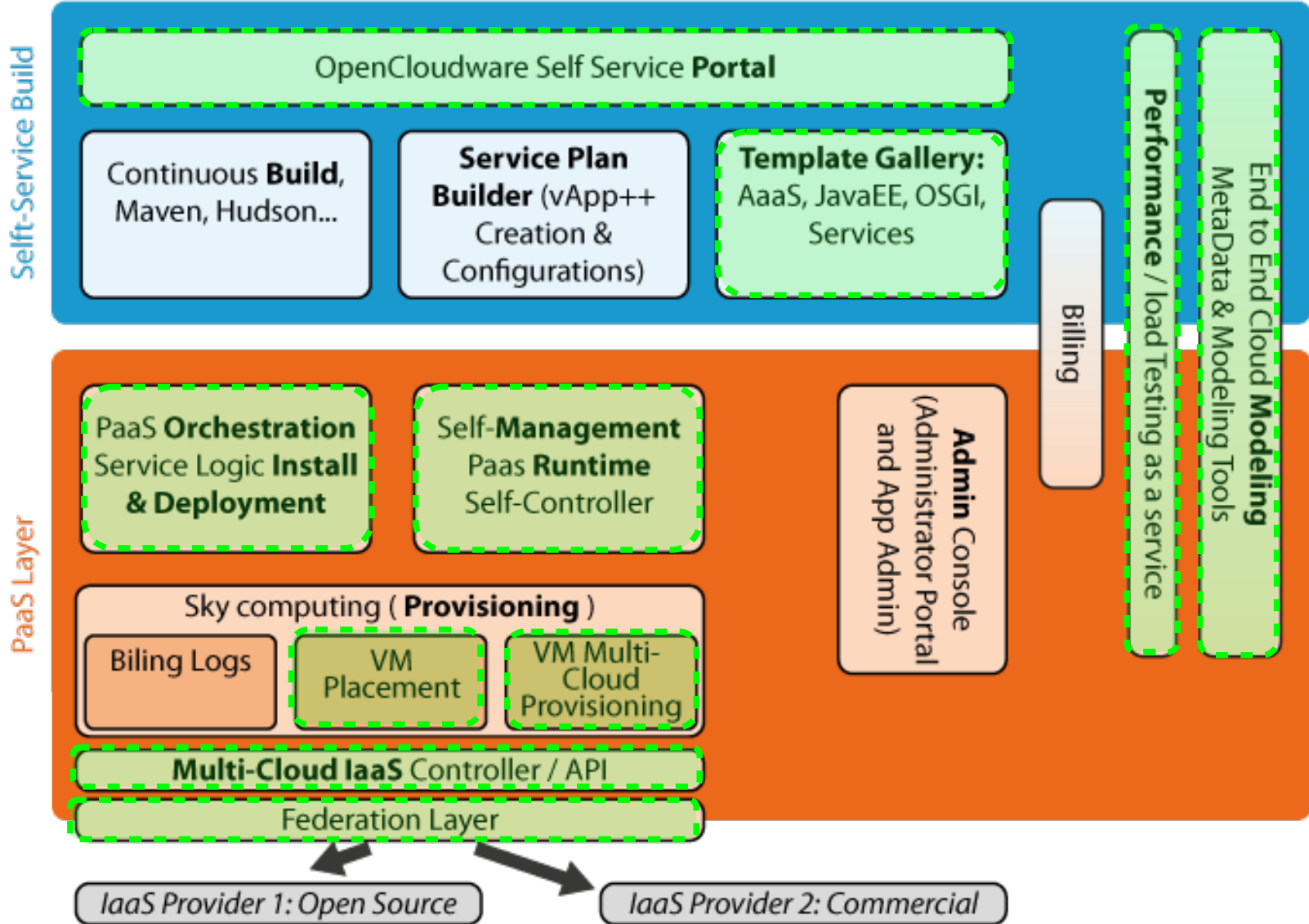
3<sup>rd</sup> party  
Cloud IaaS  
(incl. open source)

- Extensive re-use and development of **OW2 software**
  - PaaS: JavaEE, JASMINe, JORAM, Petals ESB
  - Security: AuthzForce
  - Performance: CLIF
  - Multi-iaaS: Sirocco, ProActive, Entropy
  - Portal: eXo
- Use of **other open source** software
  - Hudson, Maven, Jenkins, sunxacm, jclouds etc.
- UForge for cloud image template management and VM generation

- Ambitious project with several challenges
  - End-to-end Modeling
  - End-to-end Automation
  - Building multi-tier vApps
  - Autonomic management
  - Agnostic Multi-iaaS Portability
  - Security for identity and access, and application protection
- Links with
  - Many other open source cloud projects
  - Collaborative research projects (French, European)



# Where are we now?



# Q&A

## OpenCloudware

**A Cloud Software Platform for Building and Operating  
Enterprise PaaS to Multi-IaaS Applications**

**Thomas Debru, UShareSoft**  
**Alexandre Lefebvre, Orange Labs**