OSCi Domain 2 presentation
Massively distributed services

The aim of this document is to describe Domain 2 functional scope as well as related scientific and technical challenges.

Introduction

Cloud computing is a hot topic these days and enterprises as well as public administrations are more and more considering to migrate into the cloud in order to reduce costs and computer material based on the so called “on-demand” and “pay-as-you-go” mechanisms.

Some obstacles however still prevent many organizations to participate in this paradigm shift. Most issues are related to quality of service (QoS) management, integration and interoperability:

- QoS issues deal with the possibility to specify quality of service thanks to SLA (Service Level Agreement) contracts and to ensure that those contracts are satisfied. This requires enforcement mechanisms to take place at the infrastructure (IaaS) and platform (PaaS) levels. It allows adaptation of the platform to user's requirements such as security, reliability or performance.
- Integration issues deal with the possibility to smoothly connect in-house applications with services and applications deployed in public clouds. This is necessary since organizations do not want or are not able to outsource legacy or strategic applications and data. This integration has also to comply with previous QoS requirement, meaning that connectivity should also satisfy security, reliability or performance specification.
- Interoperability issues deal with the need for organizations to be independent from any cloud provider solutions, e.g. to avoid lock-in situations. They should be able to easily move from one to another as well as having the possibility to mix multiple solutions i.e. to build a “cloud of clouds”. Moreover, applications that are being built and operated in the cloud should be able to interoperate regardless of the underlying vendor-specific cloud infrastructure/platform.

In order to address these requirements we propose to set up an infrastructure that combine ESB and virtualization technologies in order to build a new generation PaaS (Platform as a Service) allowing to federate clouds of services deployed either inside or outside (i.e. over Internet) organizations’ networks. This paves the way to an open infrastructure that mixes private and public clouds sometimes called “hybrid cloud”.

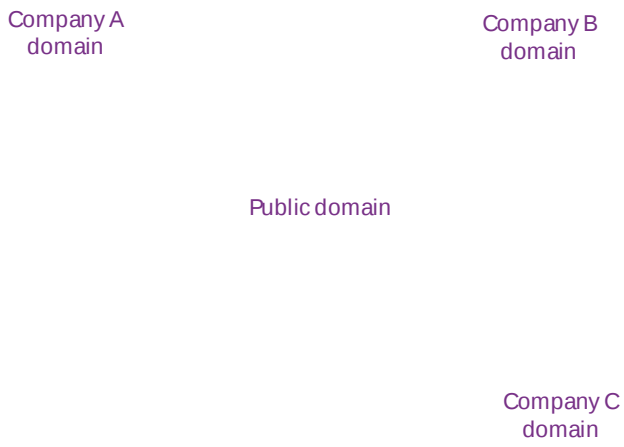
In the above figure, originating from the SOA4All[1] European project, information systems’ from 3 organizations are connected to a public cloud. Each information system constitutes a private cloud called domain made of one or many DSB nodes. These domains manage their local QoS policy in terms of

[1] http://www.soa4all.eu/. Petals Link and INRIA participate to this project that aim, among other, to define a federated distributed service bus based on Petals ESB and INRIA/ProActive.

security (service access rights, confidentiality), performance, and reliability. They are connected thanks to an interoperable, firewall friendly, federated protocol based on ProActive/OTM technology.

Main advantages of such architecture are:

- The possibility to add dynamically new domains as private clouds. Such feature allows to:
  - Manage technical and policy heterogeneity: each domain can use its own technology.
  - Interoperability: in order to join the federation, domains have to comply with the federation API. Rules have to be defined in order to specify how QoS policies should be combined and propagated from one domain to another.
- Easy service deployment: thanks to the ESB based distributed service container, one can easily connect and deploy services that becomes potentially available to all stakeholders.
- The mix of public and private clouds allows managing End to end QoS by using federation protocol. This constitutes the major competitive advantage regarding existing solutions, to be able to reuse existing applications and combine them with external services.



Scientific and technical challenges

Main scientific and technical challenges that we have to address are listed below.

Elasticity

Actual solution doesn't address elasticity at the infrastructure level: each domain manages a fixed set of Petals' nodes in charge of a group of services. Scalability may be achieved by using classical clustering solutions provided by the ESB and based on duplication of services and load-balancing allocation. Main drawback of this solution is that it is quit static.

Main idea would be to use a virtualization technology that transparently manages infrastructure resources thanks to a resource manager that can shrinks or extends a collection of virtual machines (by using an hypervisor approach for example).

By combining such technologies, the goal is to achieve “on-demand” service scalability PaaS.

Relative technologies: open-source IaaS such as ProActive, Open-Nebula, OpenStack, Apache Tashi, Xen cloud Platform or Nimbus.

Interoperability

The main challenge is to design interoperable protocol and API in order to be able to federate clouds IaaS providing from different vendors. Indeed nowadays clouds IaaS are silos that are not able to work together, even worse there is no standardized API. This produce vendor lock-in since users has to tailor their application and packaging for each solution.

Cloud infrastructure services as well as platform services, e.g. for process management or for the integration of security functionalities, should (1) provide interfaces that are based on open standards and (2) adhere to transparent QoS criteria. As opposed to the IaaS level for which already a number of standardization activities exist, standardization for the PaaS level is lagging behind.

The high sensibility of the OSS community for open standards and the transparent OSS development process fosters the evolution of an interoperable IaaS and PaaS landscape.

For example API standardization allows defining a set of functionalities that must provide any cloud technology wanting to become member of the federation.

Relative technologies: Amazon EC2 API has been adopted by many cloud technology providers; Libvirt is a common API for virtualization.

Governance & monitoring

Federation protocol allows managing a group of clouds by combining individual cloud policies based on a set of rules. Federation governance manages those rules as well as cloud life-cycle and SLA contracts. This information is stored in a distributed repository.

QoS enforcement mechanisms that ensure that SLA contracts are satisfied should be provided by clouds and managed at the federation level. This means that the QoS enforcement engine should provide also a standardized API. A monitoring layer is also available.

Relative technologies: NOSQL distributed databases such as Hadoop, Cassandra, and many others.

Security

At the security level, in order to take in charge user authentication requirement, a federated identity mechanism should be provided.

Relative technologies: Shibboleth, OpenID, Liberty Alliance

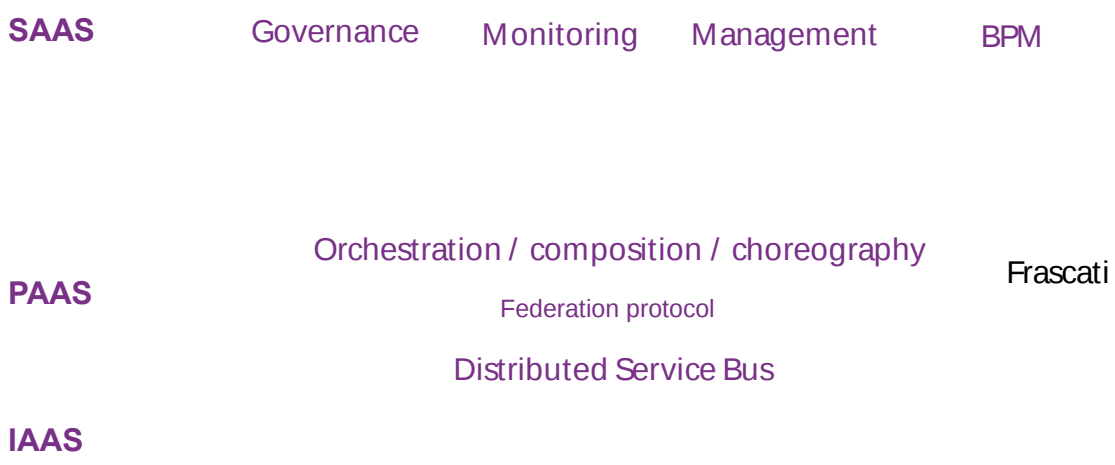
Multi-tenancy and multi-instances

Multi-tenancy refers to the partitioning of data and configuration for applications in order to support separate virtual instances of applications for different clients. Multi-instance goes a step further and provides separate software instances for different clients. Both multi-tenancy and multi-instances should be supported by the federated architecture and provided in the application programming model.

Relative technologies: virtualization at the platform level

Architecture

A first draft of the target architecture can be found below.



Partner involvement

This part is dedicated to describe the involvement of each OSCi domain 2 partners.

Petals Link

Petals link is interested to participate to the specification and development of domain 2 infrastructure as well as federation protocol.

We will leverage Petals ESB enriched with some SOA4All features.

Fraunhofer FOKUS

FOKUS is interested in contributing to the development of interoperable cloud solutions, especially on the level of platform services (PaaS), since these constitute the foundation for the realization of cross-

organizational value chains and service ecosystems. In this regard, FOKUS is interested in developing PaaS reference models and in identifying interoperability and standardization needs.

Moreover, the quality of cloud computing services, particularly when it comes to federation, should be measurable in terms of transparent criteria. FOKUS is interested in the identification and specification of such criteria that need to be defined depending on the deployment model (public, private, hybrid cloud) , the delivery model (IaaS, PaaS, SaaS) and the needs of a specific application domain (e.g. public sector). Based on the identified criteria appropriate certification mechanisms should be developed.

In addition, FOKUS proposes work in the direction of federated service management. The goal is not to provide interoperability on the level of APIs and data only, but on management processes as well, including issues such failure management, incident management, and change management, just to mention a few. In an advanced scenario, standardized and thus implementable management interfaces allow to stitch together a virtual data center by complementing I/P/SaaS offers by Management-as-a-Service.

Beihang University

Beihang University is interested in supporting the cross-domain application integration and interoperability on the levels of both SaaS and PaaS. In particular, we provide a solution to this issue by adopting service-oriented approaches. We propose a Development-as-a-Service to allow users to develop applications with online tools, deploy them into an elastic and scalable hosting environment to guarantee the quality of service and test them with monitoring.

In SaaS layer, we provide an online development toolkit supporting developers to perform development tasks, including business process modeling, service composition, testing, deployment, monitoring tools, etc. Once a new application is developed, developers need to deploy them into a runtime environment. The runtime environment is provided by platform services in the layer of PaaS that offers on-demand runtime middleware support so as to meet QoS requirements. It consists of software appliance management (e.g. scheduling and load-balancing), composite service engine, service container and etc.

INRIA

INRIA ADAM is interested in participating to the core distribution mechanism for the federated architecture as well as to the programming model for hosted end-user applications.

We will leverage the use of the FraSCAti service-oriented platform and integrate it in the federated architecture.

Activeon

USP