Portability and Interoperability in Clouds: contributions from the mOSAIC Project

Project mOSAIC: Open-Source API and Platform for Multiple Clouds
http://www.mosaic-cloud.eu

Beniamino Di Martino
Project Coordinator - Second University of Naples
beniamino.dimartino@unina.it
mOSAIC european FP7 project: main facts

- **Project acronym**: mOSAIC
- **Project full title**: Open-Source API and Platform for Multiple Clouds
- **Grant agreement no**: 256910
- **Funding Scheme**: STREP
- **Call**: FP7-ICT-2009-5  **Obj**: ICT-2009.1.2
- **Cost**: 3,705 Meur (EC financing: 2,85 M)
- **Duration**: 32 months
- **Start**: Sept 1st 2010.  **End**: April 28th 2013
- **Web site**: http://www.mosaic-cloud.eu
mOSAIC Partners

Second University of Naples – It (Prj Coordinator)

IeAT – Ro (Sci Coordinator)

European Space Agency - Fr

AITIA - Hu

Tecnalia - Sp

Terradue - It

XLAB - Slo

University of Lubljiana - Slo

Brno University of Technology - Ck
Portability and Interoperability issues

The process of developing, deploying, executing cloud applications is strongly influenced by the specifics of the cloud providers.

**Application Programming Interfaces**

- Syntactical differences
- Differences in programming models
  - Object oriented
  - REST based
  - Event driven
- Differences in API semantics
  - Different functional abstractions (especially at PaaS level)
  - Linked to application domains (especially at SaaS level e.g. enterprise patterns)
Portability and interoperability issues (cont.)

Resources and services

- Different resource semantics (especially at PaaS: e.g. stores)
- Different resources’ configurations and templates
- Different linkages of resources and configurations to provided services
- In order to interoperate, resources and services need to be retrieved and accessed; a Resource/Service Catalogue is needed, where the resources and services are (semantically) described, together with their groundings
Portability and interoperability issues (cont.)

Non-functional requirements and service levels

✓ Differences in semantics of Service level offerings and their level
✓ Mismatch between nonfunctional requests and offers
✓ no linkage of provided services and resources with service levels (especially at PaaS and SaaS)
✓ No standard or common KPIs and mechanisms to measure them
Portability and Interoperability: main contributions from the mOSAIC project

- An Agnostic, vendor neutral, API at PaaS level and an Open Source Platform, with adapters to most notable Cloud Providers’ APIs
- A Cloud Agency for multiCloud Services brokering, SLA monitoring and dynamic reconfiguration
- A machine-readable Cloud Ontology
- A Semantic Engine, for discovering Cloud API components, resources and Services, driven by functional and Application domain concepts, Cloud Patterns and inference rules
- A Dynamic Semantic Discovery Service, for discovering Cloud providers’ resources and services, mapping, aligning and composing them
mOSAIC API Architecture

Lowest Layer: Native resource protocol (Web service, RPC, etc.), or a native resource API provided as a library by the vendor for a certain programming language. No uniformity.

Driver API: Wraps the native API, providing the first level of uniformity: all resources of the same type are exported with the same interface. Thus exchanging, for example, an Amazon S3 with a Riak key-value store is just a matter of configuration.

Connector API: depending on the programming language, provides abstractions for the cloud resources, suitable for the programming paradigm. This is where we provide the second kind of uniformity for the programming paradigms, as all the implementations of the connector API in object oriented programming languages will have similar class hierarchies, method signatures, or patterns.

Cloudlet API: Even thought the developer already can access cloud resources, he or she must restrict himself or herself to a cloud compliant programming methodology, which we provide (integrated with all the layers already mentioned) that we call Cloudlet, as similar with the existing Java Servlet technology that provides standard programming components in J2EE environments.
mOSAIC API’s Layers

- Application components
- Support for components
- For different languages
- Reference API
- For same service
Semantic technology for portability - interoperability

To define a common, machine readable, dictionary, able to express resources, services, APIs and related parameters, SL requirements and offers, and related KPIs

To support code portability, by allineating and reconciliating different APIs and resources

To bridge the gap between the domain related functionalities and cloud resources and services

To support interoperability, by matchmaking Service interfaces

To support (semantic based) Cloud vendors’ resource and services discovery, and integration within an agnostic API
Semantic technology for portability - interoperability

To support Brokering, Negotiation and Service level Agreement, by matchmaking nonfunctional user requirements and provider offers.

To support dynamic resources reconfiguration, by monitoring SL parameters and reacting with applying heuristic rules.
Semantic technologies in the mOSAIC project

A Cloud Ontology able to provide a common definition of concepts related to Cloud domains and to describe Cloud components like infrastructures, platforms and services.
mOSAIC Ontology: Top Level and Standards/Proposals

NIST
mOSAIC Ontology: Top Level and Standards/Proposals

OCCI
mOSAIC Ontology: Top Level and Standards/Proposals

SLA@SOI
mOSAIC Ontology: Top Level and Standards/Proposals

- IBM/Oracle
- Azure/ Google/AWS/
- OpenNebula/
- Openstack/
- et. al.
IEEE P2302 – “Intercloud” Standard for Intercloud Interoperability and Federation (SIIF)

mOSAIC Cloud ontology is being included in the Standard
Semantic Engine

Offers a browsable catalogue of Cloud Services, Appliances and Cloud Patterns, represented in agnostic way.

Maps vendor specific resources, resources configurations, services, appliances and patterns among them and to their equivalent agnostic representation.

Offers semantic search and inferencing over resources, services, appliances, Cloud and application patterns.

Suggests proper resources’ configurations tailored to SL specifications.

Represents the different APIs semantically, vendor agnostically, and independently from programming model.
Dynamic Semantic Discovery Service

Discovers Cloud providers’ resources and services, Maps and allineates among them and suggests how to compose them; Supports the mOSAIC API developer and maintainer to develop new drivers and connectors to wrap and compose the discovered services.
Cloud Agency

Cloud Agency is a multi agent system (MAS) that accesses, on behalf of the user, the utility market of Cloud computing to manage always the best resources configuration that satisfies the application requirements.

Interoperability
Running Cloud Agency through CA-GUI
starting/stopping VMs, loading and attaching VM images, deploying and executing applications
CA GUI: SLA Monitoring

Visualization of performance indexes

Creating triggers on resource parameters
Monitoring the resource

Monitoring Host: 10.254.20.123
O.S. Linux x86

CPU usage graph:
- cpu_user: Percentage of CPU utilization that occurred while executing at the user level.
Setting up reconfiguration triggers

Provider: google
Resource type: Storage
Parameter: writetime

Rule list:

Add rule  Delete rule
Adding new rules
Thanks for your attention!

beniamino.dimartino@unina.it
http://www.mosaic-cloud.eu