Portability and Interoperability in Clouds: Agents, Semantics and Volunteer Computing can Help - the mOSAIC and Cloud@Home Projects

HPC’2012 – June 29th 2012 – Cetraro (IT)

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:** 30 months
- **Start:** Sept 1st 2010. **End:** Feb 28th 2013
- **Web site:** http://www.mosaic-cloud.eu
mOSAIC Partners

Second University of Naples – It (Prj Coordinator)

Institute IeAT – Ro

European Space Agency - Fr

AITIA - Hu

Tecnalia - Sp

Terradue - It

XLAB - Slo

University of LUBLjiana - Slo

Brno University of Technology - Ck
Position in The FP7

Within Objective 1.2, a total of 195 proposals were received, of which 28 were selected. Together with one project from Objective 9.4.
Project acronym: Cloud@Home

Project full title: Cloud@Home: a New Enhanced Computing Paradigm

Funding Scheme: MIUR PRIN

Call: MIUR PRIN 2008

Cost: 500 Keur (MIUR financing: 250K)

Duration: 30 months


Web site: http://cloudathome.unime.it/
Cloud@Home Partners

• University of Messina (Coordinator)
• Second University of Naples
• University of Catania
• University of Pisa
• University of Sannio
(Some) 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 (expecially at PaaS level)
  - Linked to application domains (expecially at SaaS level e.g. enterprise patterns)
(Some) portability and interoperability issues

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
(Some) portability and interoperability issues

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, Interoperability and Semantic technologies in 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 Services brokering and SLA monitoring and resource reconfiguration
- A Cloud Ontology
- A Semantic Engine, for finding mOSAIC API components and resources, driven by functional and Application domain concepts, patterns and rules
- A Dynamic Semantic Discovery Service, for discovering Cloud providers’ resources and services, aligning them with mOSAIC API components and resources
mOSAIC Approach

The mOSAIC project aims to develop an open-source platform that enables applications to negotiate Cloud services as requested by their users.

The platform will implement a multi-agent brokering mechanism that will search for services matching the applications’ request, and possibly compose the requested service if no direct hit is found.

Using the Cloud ontology and Semantic Engine, application developers will be able to specify their requirements (functionalities and resources) and service level requests.
mOSAIC Approach

Cloud-application developers will be able to postpone their decision on the procurement of Cloud services until runtime.

End-user applications will be able to find best-fitting Cloud services to their actual needs and efficiently outsource computations.

mOSAIC will facilitate competition and cooperation among Cloud providers, who, in return, will be able to reach customers they could not reach before.
mOSAIC Architecture and components

An API

Cloud-based language- and platform-independent API
Extends the existing language- or platform-dependent API capabilities with composite features based on patterns

A framework

Semantic engine
Cloud ontology & Semantic representation of Cloud resources
Applications’s needs in terms of SLAs and QoS requirements

Cloud agency
Dynamic Semantic Discovery Service

Application Tools

An open-source platform

a proof-of-the-concept prototype ready to be tested, exploited or extended by its users
include instances of the APIs for two programming languages and application tools

Proofs of validity through the use cases and applications
mOSAIC Components

Cloud-enabled applications
- mOSAIC's proof-of-the-concept applications
  - Earth Observation applications
  - Intelligent maintenance system
  - Model exploration service
- mOSAIC PaaS and IaaS
  - Application support
    - API implementations
      - Java cloudlets
      - Python cloudlets
      - Java connectors
      - Python connectors
      - Demo applications
    - Application tools
      - Eclipse plug-ins
      - Frontends (cmdl, web)
      - Network backends
      - Configuration tools
      - Portable Testbed Cluster
    - Service discoverer
      - Semantic engine
        - Semantic query builder
        - Pattern builder
        - Reasoner
        - Maintainer
        - Search engine
        - Ontologies
- mOSAIC’s platform components
  - Application service components
    - SLA
    - Network
    - Benchmark
  - Application support components
    - Deployable COTS
      - Drivers
  - Platform’s core components
    - Register & Discover
    - Packager & Deployer
    - Provisioner & Monitor
    - Operate & Maintain
    - Scheduler & Scaler
    - Interoperability support
      - mOS

Hosting services supported by mOSAIC
- Amazon
- IBM CloudBurst
- Rackspace
- GoGrid
- Hostko
- CloudSigma
- CHS
- Others

Deployable services support
- Eucalyptus
- OpenNebula
- DeltaCloud
- OpenStack
- HDFS

User community developed applications
- Information extraction
- Analysis of structures

Software platform support

Infrastructure support
- Cloud Agency
  - MTP
  - Mediator
  - Meter
  - Archiver
  - Tier agents
- Agents for Cloud Agency
  - Broker
  - Vendor agents

Clouds adaptors

Open source

mOSAIC’s proof-of-the-concept applications
- Earth Observation applications
- Intelligent maintenance system
- Model exploration service

API implementations
- Java cloudlets
- Python cloudlets
- Java connectors
- Python connectors
- Demo applications

Application tools
- Eclipse plug-ins
- Frontends (cmdl, web)
- Network backends
- Configuration tools
- Portable Testbed Cluster

Service discoverer
- Semantic engine
  - Semantic query builder
  - Pattern builder
  - Reasoner
  - Maintainer
  - Search engine
  - Ontologies

Platform’s core components
- Register & Discover
- Packager & Deployer
- Provisioner & Monitor
- Operate & Maintain
- Scheduler & Scaler
- Interoperability support
  - mOS

Application service components
- SLA
- Network
- Benchmark

Application support components
- Deployable COTS
  - Drivers

Hosting services supported by mOSAIC
- Amazon
- IBM CloudBurst
- Rackspace
- GoGrid
- Hostko
- CloudSigma
- CHS
- Others

Deployable services support
- Eucalyptus
- OpenNebula
- DeltaCloud
- OpenStack
- HDFS

User community developed applications
- Information extraction
- Analysis of structures
mOSAIC Key features and technologies

Vendor agnostic API
Open source PaaS
Cloud resources and services brokering
Cloud Agency
SLA negotiations and monitoring
Cloud Ontology
Semantic Engine
Dynamic Semantic Discovery Service

Component-based applications
Multiple Clouds
Long time running applications
Event driven, asynchronous programming model
mOSAIC API

Concepts:
   in public D1.3/Sept 2011 & papers

Implementations:
   In Java, available at:
   http://www.mosaic-cloud.eu  -> <For Developers> box
   https://bitbucket.org/mosaic/
   Guide in mosaic-api / mosaic-mvn / doc
   In Python, in September 2012
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 though 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

- Component
  - Cloudlet API
  - Connector API
  - Interoperability API
- Component
  - Cloudlet API
  - Connector API
  - Interoperability API

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) resource and services discovery
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
Semantic technologies in the mOSAIC project

The Semantic Engine:
overcomes syntactical differences representing and annoting the API semantically, independently from programming model.

offers a catalogue of functionality related to Cloud domain, representing specific services in agnostic way.

offers semantic full text search with the use of semantic thesaurus.
Semantic technologies in the mOSAIC project

The Semantic Engine:
Links together services and resources and maps them with grounding implementation.
Helps to express non functional requirements and supports construction of SLAs depending on concepts related to patterns and heuristic rules.
Semantic technologies in the mOSAIC project

✓ A Dynamic Semantic Discovery Service, for discovering Cloud providers’ resources and services, aligning them with mOSAIC API components and resources. Together with Semantic Engine, the discovery service helps to enrich the catalogue of services and automatically classify them, abstracting and annotating them.

✓ Support to mOSAIC’s Cloud Agency to express brokering policies and to find best fitting provider according to SL requests. Semantic based rules can be defined in Cloud Agency to express Service level monitoring and reconfiguration rules.
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.
Vendor Agents

- The overall goal of the Vendor Agents (VA) inside the Cloud Agency is to mediate the relationship of their clients with the specific cloud providers they are connected to.
- VAs create a separation layer between the Cloud Agency and the Cloud Provider and hide the user applications and other agents from the details of the cloud provider, the resources they use and the infrastructure they run on.
- Vendor Agents provide *resource provisioning* and *resource management*.
Vendors' specifics addressed

- **The resources types they provide:** compute and storage resources are quite common. But they are sometimes complemented with load balancers, relational databases, map-reduce, elastic IPs, etc;

- **The operations on resources** including the way they are created, destroyed, related with each other, etc;

- **The resource characteristics:** CPU, RAM, prices and the quality of the services

- **Interaction mechanisms:** there are various API types which are available depending on the cloud provider like REST, SOAP or language libraries.

- **Security credentials:** usernames and passwords are widely used. But there are also specific keys which can even differ for accessing different resources types on the same provider.
Report “Advances in Clouds”

Report from the Cloud Computing Experts Working Group of the European Commission (DG INFSO, Unit Internet of Services, Software and Virtualization)

Presented in Brussels on May 2° 2012

Credit where credit is due

- Maximilian Ahrens, Zimory, Germany
- Prashant Barot, Oracle, Germany
- Francis Behr, Liberty Surf, France
- Dr. Ivona Brandic, TU Wien, Austria
- Dr. Marcus Brunner, NEC Europe Ltd., Germany
- James Clarke, Waterford Institute of Technology – TSSG, Ireland
- Dr. Stuart Clayman, University College London, UK
- Dr. Thierry Coupaye, France Telecom Orange Labs, France
- Dr. Peter Dickman, Google, Switzerland
- Dr. Margot Dor, ETSI, France
- Dr. Ake Edlund, KTH, Sweden
- Prof. Dr. Erik Elmroth, Umeå University, Sweden
- Andreas Ebert, Microsoft Corporation, Austria
- Jürgen Falkner, Fraunhofer IAO, Germany
- Ana Maria Juan Ferrer, Atos Research and Innovation, Spain
- Mike Fisher, British Telecom, UK
- Dr. habil. Alfred Geiger, T-Systems
- Andrés Gómez Tato, Fundación Centro Tecnológico de Supercomputación de Galicia, CESGA, Spain
- Prof. Dr. Yi-Ke Guo, Department of Computing, Imperial College London, UK
- Dr. Giles Hogben, ENISA, Greece
- Prof. Dr. Keith Jeffery, Science & Technology Facilities Council, UK
- Dr. Ricardo Jimenez-Peris, Universidad Politecnica de Madrid, Spain
- Prof. Dr. Frank Leymann, IAAS, University of Stuttgart, Germany
- Prof. Dr. Ignacio M. Llorente, C12G Labs, Spain
- Prof. Dr. Beniamino di Martino, University of Naples, Italy
- Burkhard Neidecker-Lutz, SAP Research, Germany
- Prof. Dr. Dana Petcu, West University of Timisoara, Romania
- Thomas Piwek, Deutsche Lufthansa AG, Germany
- Dr. Harald Schoening, Software AG, Germany
- Lutz Schubert, HLRS, Germany
- Prof. Dr. Theodora Varvarigou, SCES NTUA, Greece
- Dr. Yaron Wolfshal, IBM, Israel

Thank you for your contribution!
Cloud Standards Customer Council

Practical Guide to SLAs Webinar
Download the SLA Whitepaper

Melvin Greer, Senior Fellow and Chief Strategist, Cloud Computing, Lockheed Martin; Chair, CSCC Steering Committee

Cloud service level agreements are important to clearly set expectations for service between cloud consumers and providers. Providing guidance to decision makers on what to expect and what to be aware of as they evaluate and compare SLAs from cloud computing providers is critical since standard terminology and values for cloud SLAs are emerging but currently do not exist.

Amy Wohl, Editor, Amy Wohl's Opinions

Today, customers complain regularly that SLAs are just another form of vendor boilerplate, to the extent they exist at all, and that it is difficult if not impossible to get much modification. They also point out that they want the SLA because it will cause the provider to put some skin in the game, not because the penalties would solve their problems in the case of outages or other situations covered by the SLA. That doesn’t mean we don’t need SLAs; we do. It’s important we make it clear what is going on now versus what we would like to see/influence for the future and when we are hoping that future will occur.

The Cloud Standards Customer Council held a webinar to introduce the completed "Practical Guide to Cloud Service Level Agreements," on Tuesday, April 10, 2012.

If you missed any or all of the webinar you can download the deck or the entire webcast below.

Download webinar
Download PDF

Download the SLA Whitepaper

The Guide highlights the critical elements of a service level agreement (SLA) for cloud computing and provides guidance on what to expect and what to be aware of when negotiating an SLA. The guide articulates a set of requirements from a consumer’s perspective and identifies elements that need to be addressed via open standards through CSCC’s liaison partnerships with key standards development organizations.

Melvin Greer, senior fellow and chief strategist, Cloud Computing, Lockheed Martin; chair, CSCC steering committee, lead the webinar describing the rationale behind the development of the guide, the target audience and the intended benefits of the guide. A question and answer period will immediately follow the presentation.

Representatives from the following organizations developed the Practical Guide to Cloud Service Level Agreements, along with input and feedback from the general CSCC membership: Boeing, CA Technologies, cebe IT & KM, Cloud Perspectives, CloudOne Corporation, Ekartha, Fort Technologies, Hoboken Consulting Group LLC, IBM, Kroger, Lockheed Martin, Powersoft Computer Solutions Ltd., Second University of Naples, and Wohi Associates.
Acknowledgements

The Practical Guide to Cloud Service Level Agreements is a collaborative effort that brings together diverse customer-focused experiences and perspectives into a single guide for IT and business leaders who are considering cloud adoption. The following participants have provided their expertise and time to this effort.

Workgroup Leaders
John Meegan (IBM) – Lead Technical Editor; Introduction and Keys to Success Section Leader
Gurpreet Singh (Ekartha) – Current SLA Landscape and Disaster Recovery Section Leader
Steven Woodward (Cloud Perspectives) – Roles & Responsibilities; Performance Objectives Leader
Salvatore Venticinque (Second University of Naples) – Service & Deployment Model Section Leader
Massimiliano Rul (Second University of Naples) – Service & Deployment Model Section Leader
David Harris (Boeing) – Business Policies Section Leader
Gerry Murray (East Technologies) – Business Policies Section Leader
Salvatore Di Martino (Second University of Naples) – Business Policies Section Leader
Eric Le Roux (CA Technologies) – Security & Privacy Section Leader
John McDonald (CloudOne Corporation) – Service Management Section Leader
Ryan Kran (The Kroger Co.) – Service Failure Management Section Leader
Marlon Edwards (Hoboken Consulting Group, LLC) – Disaster Recovery Section Leader
Dave Russell (IBM) – Management Process Section Leader
George Mallekis (Powersoft Computer Solutions Ltd) – Exit Process Section Leader

Extended Workgroup Members
The workgroup leaders wish to recognize the following individuals for their outstanding efforts to provide content, share their expertise and ensure completeness of the Practical Guide to Service Level Agreements: Amy Wohl (Wohl Associates), Acker Bond (Elastic Provisioner, Inc.), Claude Baudoin (Obe IT & KMI), Christopher Ferris (IBM), Mehrin Greer (Lockheed Martin), Richard Miga (Synergistic Solutions), Thomas Somers (IBM).

Additional Reviewers
The following reviewers provided feedback on the Practical Guide to Cloud Service Level Agreements:
Jenny Huang (AT&T), Karen Coraway (The MITRE Corporation), Kenneth Dilbeck (ThuForum), Roopali Thapar (IBM), Tobias Kunze (Red Hat).

April 10, 2012

http://www.cloud-council.org/04102012.htm
IEEE P2302 – “Intercloud” Standard for Intercloud Interoperability and Federation (SIIF)

Contribution to:

Section 6.9, Ontology Definition:
Section 6.10, Decentralized Ontology Deployment:
Thanks for your attention!

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