Portability and Interoperability in Clouds: contributions from the mOSAIC Project

CLASS Conference—October 24th 2012 – Bled (SI)

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
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


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
(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 (especially at PaaS level)
  - Linked to application domains (especially 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
- Information extraction
  - Analysis of structures

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

Software platform support
- API implementations
  - Java cloudlets
  - Python cloudlets
  - Java connectors
  - Python connectors
  - Demo applications
- mOSAIC PaaS and IaaS
  - Application tools
- Application service components
  - SLA
  - Network
  - Benchmark
- Application support components
  - Deployable COTS
  - Drivers

Infrastructure support
- Semantic engine
  - Semantic query builder
  - Pattern builder
  - Reasoner
  - Maintainer
  - Search engine
  - Ontologies
- Cloud Agency
  - MTP
  - Mediator
  - Meter
  - Archiver
  - Tier agents
- Agents for Cloud Agency
  - Broker
  - Vendor agents
- Hosting services supported by mOSAIC
  - Amazon
  - IBM CloudBurst
  - Rackspace
  - Flexiscale
  - GoGrid
  - CloudSigma
  - Arctur
  - Hostko
- Others
- Deployable services support
  - Eucalyptus
  - OpenNebula
  - DeltaCloud
  - OpenStack
  - CHS
  - HDFS
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
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

**mOSAIC API Layers**

- **Native Protocol**
- **Native API**
- **Driver API**
- **Interoperability API**
- **Connector API**
- **Cloudlet API**
- **Core Components**
- **User Components**
- **Miscellaneous Components**
- **SLA Components**

**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

- **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 aligning 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
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
IEEE P2302 – “Intercloud” Standard for Intercloud Interoperability and Federation (SIIF)

Contribution to:

Section 6.9, Ontology Definition:
Section 6.10, Decentralized Ontology Deployment:
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.
Cloud Agency’s GUI

- Cloud IP: 127.0.0.1
- CA Port: 10001

**Provisioning**
- **startCFP**
- CFPs
- Proposals
- Authentication

**CFP**
- Open: `./aux_files/recentcfp/testCfp.xml`
- startCFP

**CFP List**
- `./aux_files/recentcfp/testCfp.xml`

**Notification**
- OK configuration file selection
- Successful connection press ok to continue
- Welcome to CloudAgency Client
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
mOSAIC SLA Architecture
SLA model

- The Agency will use a uniform SLA model
- SLA@SOI and **WS-Agreement** are our current reference
- **OCCI** is our current reference for description of Cloud resource

- Clients use this model to ask for Cloud Resources
- Vendor Agents translate SLA from Specific Provider Language into the internal model
<ws:Template ws:TemplateId="t1"
xmlns:ws="http://schemas.ggf.org/graap/2007/03/ws-agreement">
  <ws:Name>Test1</ws:Name>
  <ws:Context>
    <ws:ServiceProvider/>
    <ws:TemplateId>t1</ws:TemplateId>
    <ws:TemplateName>Test1</ws:TemplateName>
  </ws:Context>
  <ws:Terms>
    <ws:All>
      <ws:ServiceDescriptionTerm ws:Name="SAMPLE REQUEST0"
ws:ServiceName="SET VARIABLE">
        <mod:Storage xmlns:mod="http://occi-wg.org/model">
          <title>ANY</title>
          <size>ANY</size>
          <summary>ANY</summary>
        </mod:Storage>
      </ws:ServiceDescriptionTerm>
    </ws:All>
  </ws:Terms>
  <ws:CreationConstraints/>
</ws:Template>
Monitoring and Reconfiguration

- Two different Services
  - Monitoring of Cloud Resources (not applications)
  - Intelligent event-driven reconfiguration
- RESTful access
- Multi Agent implementation
- Multi model support for reasoning
- Policy based
Monitoring and Reconfiguration

- Benchmarkers and monitors
- Cloud
- Daemon
- Monitoring Results DB
- Observer
- Reasoner
- Executor
- Event Bus
- Archiver
- Store Data
- Communication with users's applications
- Read
- Subscribe
- Push event
- Read/Write
- External Application
Archiver

- Parses SLA
- Find available monitors for:
  - service levels
  - specific provider
- Starts monitoring
Data Collection

- Performance information are collected by the archiver.
- Data are collected, processed and stored.
- They are available to agents and to Cloud application.

[Diagram showing the process of data collection, archiving, and storage]
Linker and Benchmark

- Linker provides access to Resource functionalities
- Benchmark uses linker to collect necessary information
- Linker is technology dependent
- The suited couple of linker-benchmark is embedded into the user application.

1: Read SLA
2: Get Google linker
3: Get benchmarks
4: Pack all components
5: Deploy on cloud
Reasoner anatomy

- Reasoner is a MAS itself.
- Observers subscribe to the Event bus to detect events.
- Observers make decision about what actions to be taken.
- Executors execute action according to observer requests.
- Executors can raise new events.
Configuring interaction

- Reasoner configuration
- Observers and Executors are launched according to it
- Messages among agents are routed by an agent again
Information flow

1: Read Monitoring Results
2: Process Results
3: Generate Event
4: Send Event
5: Forward Event
5: Push Event

Monitoring Results DB
Observer
Routing Agent
Executor
Event Bus
# Relational Observer

- From the SLA it is possible to select provider
- Resource
- Parameter to be monitored
Defining a new rule
Defining a new rule

Values aggregation: average

Rule relation:
- % of SLA var
- absolute var
- verification
- periodical, with period [s]:
- on event from: Logger

Action:
- send event

Active rules: Add Rules

Disable rules: Add Rules

Executors:
- ALL
- Select Executors

OK Cancel
Defining a new rule
Examples of rules

Rule list:

IF the average value is lower than the 98% of sla value THEN send event. Verification on event from 'ReconfigurationExecutor'. State: ACTIVE

IF the average value is not equal to the 98% of sla value THEN disable rules. Verification on event from 'Logger'. State: ACTIVE

IF the last value is equal to 1000.0 THEN send event. Rule period [s]: 60. State: IDLE

IF the minimum value is greater than the 98% of sla value THEN active rules. Rule period [s]: 1000. State: ACTIVE

Add rule  Delete rule
Prolog Based Observer

:- dynamic(addEvent/6).

saturation(ID) :- event(storage,ID,free_space,average,lower,X), X < 5 .
saturation(ID) :- event(storage,ID,free_space,average,lower_equal,X), X <= 5 .
saturation(ID) :- event(storage,ID,free_space,last,lower,X), X < 5 .
saturation(ID) :- event(storage,ID,free_space,last,lower_equal,X), X <= 5 .
slow(ID) :- event(compute,ID,free_memory,average,lower,X), X < 5 , event(compute,ID,cpu_usage,average,greater,Y), Y > 95 .
slow(ID) :- event(compute,ID,free_memory,average,lower,X), X < 5 , event(compute,ID,cpu_usage,average,greater_equal,Y), Y >= 95 .
slow(ID) :- event(compute,ID,free_memory,average,lower_equal,X), X <= 5 , event(compute,ID,cpu_usage,average,greater,Y), Y > 95 .
slow(ID) :- event(compute,ID,free_memory,average,lower_equal,X), X <= 5 , event(compute,ID,cpu_usage,average,greater_equal,Y), Y >= 95 .

action(saturation,update,storage).
action(slow,add_resource,compute).

% program interface
performAction(ACTION,RESOURCE,ID) :- action(saturation,ACTION,RESOURCE), saturation(ID) .
performAction(ACTION,RESOURCE,ID) :- action(slow,ACTION,RESOURCE), slow(ID) .

addEvent(RESOURCE,ID,PARAMETER,AGGREGATION,RELATION,PERCENTAGE_VALUE) :- asserta(event(RESOURCE,ID,PARAMETER,AGGREGATION,RELATION,PERCENTAGE_VALUE)) .

%
LOG from Executor for Re-Negotiation

Tue May 10 13:53:18 CEST 2011
Event type: observer_event
parameter_result: 953.06666666667
parameter_type: writetime
relation: lower
aggregation: average
parameter_sla_value: 1000.0
resource: Storage
parameter_limit: 980.0
resource_id: google_storage01
provider: google

Tue May 10 13:53:18 CEST 2011
Event type: reconfigurator_event
resource: storage
action: change_provider
resourceId: google_storage01
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/insist for the future and when we are hoping that future will occur.

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.

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

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, Cebit IT & KM, Cloud Perspectives, CloudOne Corporation, Ekantia, Fort Technologies, Hoboken Consulting Group LLC, IBM, Kroger, Lockheed Martin, Powersoft Computer Solutions Ltd, Second University of Naples, and Wohl 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 Morgan (IBM) – Lead Technical Editor; Introduction and Keys to Success Section Leader
Garpeet Singh (Ekarthia) – Current SLA Landscape and Disaster Recovery Section Leader
Steven Woodward (Cloud Perspectives) – Roles & Responsibilities; Performance Objectives Leader
Salvatore Ventimiglia (Second University of Naples) – Service & Deployment Model Section Leader
Massimiliano Rak (Second University of Naples) – Service & Deployment Model Section Leader
David Harris (Boeing) – Business Policies Section Leader
Gerry Murray (Fort Technology) – Business Policies Section Leader

Gianluca Di Martino (Second University of Naples) – Business Policies Section Leader

Frederic Rousseau (CA Technologies) – Security & Privacy Section Leader

John McDonald (CloudOne Corporation) – Service Management Section Leader
Ryan Kean (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 Malekakis (Powarsoft 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), Asher Bond (Elastic Provisioner, Inc), Claude Baudoin (cobe IT & EMI), Christopher Ferris (IBM), Mehvin 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 Carraway (The MITRE Corporation), Kenneth Dilbeck (TmForum), Roopali Thapar (IBM), Tobias Kunz (Red Hat).

April 10, 2012

http://www.cloud-council.org/04102012.htm
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

- André 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 Wolfsthal, IBM, Israel

Thank you for your contribution!
Thanks for your attention!

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