The challenges of Multi-Clouds

Dana Petcu

West University of Timisoara & Institute e-Austria Timisoara

Agenda – more concrete

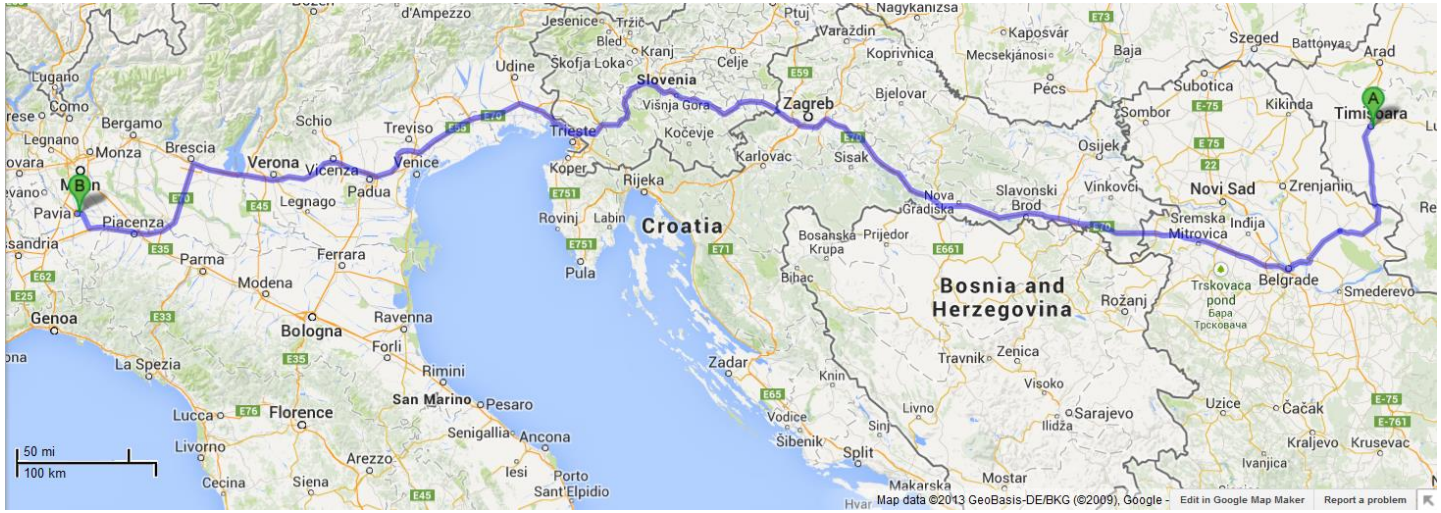
- **Generalities**
  - Backgound
  - Clouds and their future?
  - Why Multiple Clouds?
  - Taxonomy of Multiple Clouds
  - Interoperability & portability
- **Solutions**
  - mOSAIC: for portability
  - MODAClouds: model-driven engineering
  - SPECS: security SLA management

A Step Back

From Where? And Background

University and Faculty

- **West University of Timisoara** (www.uvt.ro/en)
  - More than 20 000 students
  - 11 faculties
- **Faculty of Mathematics and Computer Science** (www.math.uvt.ro)
  - More than 1000 students (undergraduate, master, PhD)
  - Two departments: Maths and CS



Department & Research Center

- **Computer Science Department** (web.info.uvt.ro)
  - Around 700 students (undergraduate, master, PhD)
  - Studies in Romanian and English
  - Foreign students coming in Erasmus programme
  - 35 teachers
  - Master (English): Artificial Intelligence & Distributed Computing (www.math.uvt.ro/invatamant/cicluri/masterat/informatica/aidc)

- **Research Center in Computer Science** (research.info.uvt.ro)
  - Parallel & Distributed Computing, AI & Nature Inspired Computing
  - Runs around 5 national & international R&D projects per year
  - Manage the biggest supercomputing center of Romania

HPC Center

http://hpc.uvt.ro

400 cores Cluster

4000 cores BlueGene/P

3000 cores GPU cluster

Research spin-off, IeAT

Institute e-Austria Timisoara (www.ieat.ro)

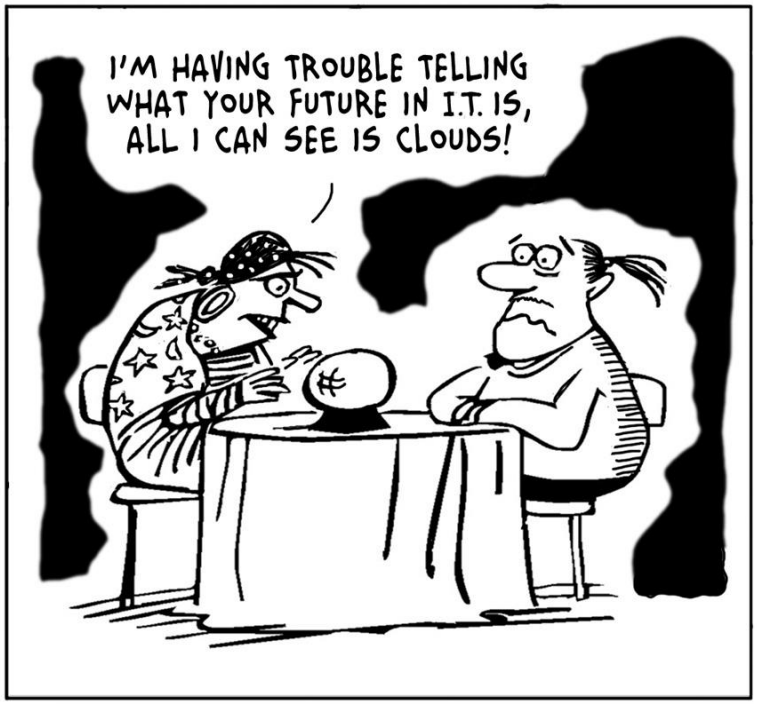
- 10 years old private research institute in Computer Science
- Non-profit association between 3 public institutions (2 universities from Romania and one from Austria)
- More than 40 employees
- Funded only on projects
- R&D project obtained by national/international competitions
- Technological transfer type of contracts with industry
- PhD and master students working in R&D projects to complete their theses
- Support the R&D activities of the universities involved

Parallel & Distributed computing Group

- …
- **2000-2009**
  - Grid computing – tools and applications in symbolic computing, Earth Observation
  - Services – orchestrations, semantics
  - Parallel computing in image processing, evolutionary computing, formal verification, symbolic computing
- **2010-2013**
  - Cloud computing
  - Scalability in parallel computing, scheduling

Projects/2013 @ UVT & IeAT

- **Cloud**
  - EC-FP7 MODAClouds — www.modaclouds.eu — 2012-2015
  - EC-FP7 mOSAIC — www.mosaic-cloud.eu — 2010-2013 Sci. lead
  - EC-FP7 SPECS — www.specs-project.eu — 2013-2016
  - EC-CIP SEED — www.seed-project.eu — 2012-2014
  - RO-PNII AMICAS — amicas.hpc.uvt.ro — 2012-2014
- **Grid**
  - EC-FP7 EGI Inspire — www.egi.eu — 2010-2014
- **Parallel**
  - EC-FP7 HOST — host.hpc.uvt.ro — 2012-2014 Lead
  - EC-FP7 HP-SEE — www.hp-see.eu — 2010-2013
- **Others: security, digital**
  - EC-FP7 SPaCioS — www.spacios.eu — 2010-2013
  - EC-FP7 SCAPE — www.scape-project.eu — 2011-2014

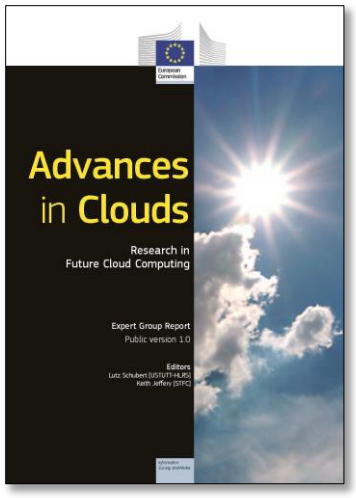


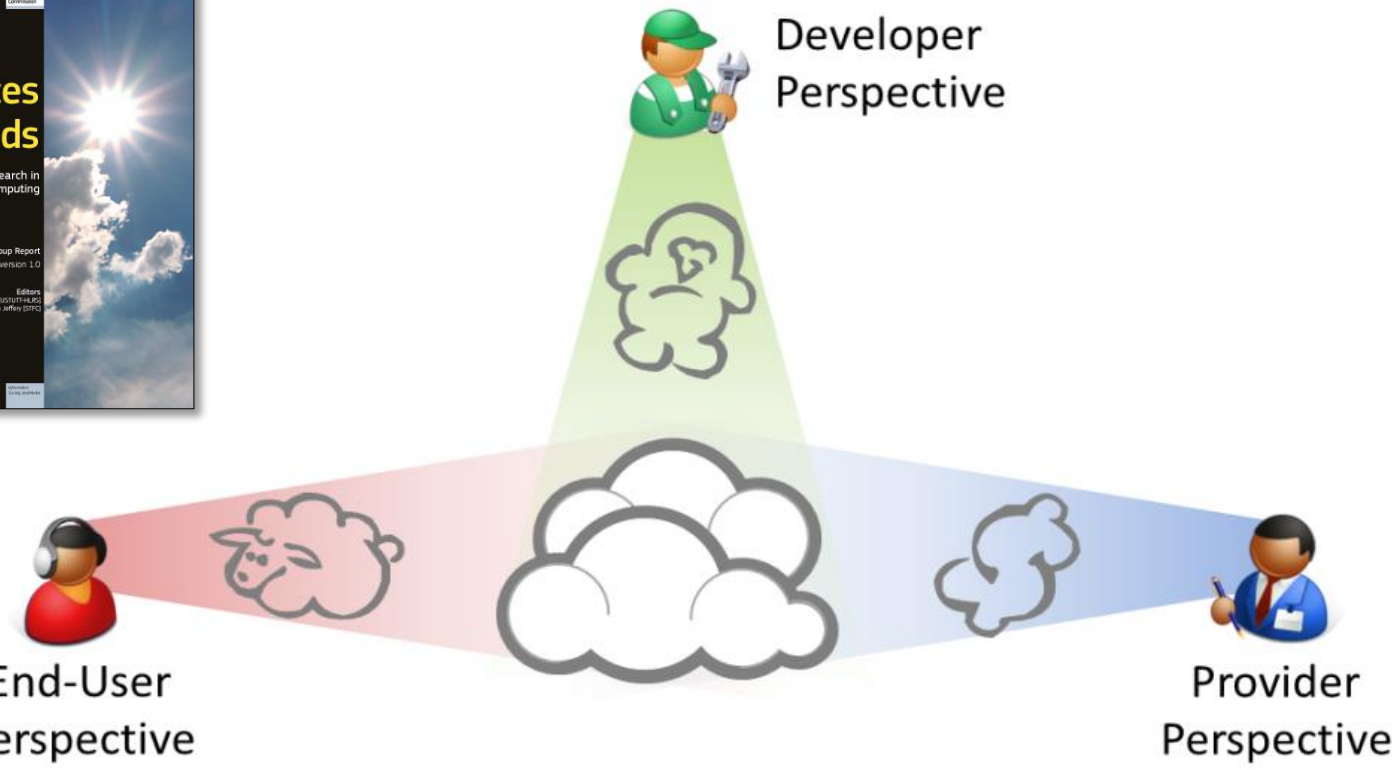
Clouds and their future

Generalities

Cloud Computing – Definition?

Source: http://cordis.europa.eu/fp7/ict/ssai/docs/future-cc-2may-finalreport-experts.pdf





Provider perspective

Clouds are dynamic (resource) environment
that guarantee availability, reliability & related quality aspects
through automated, elastic management of the hosted services

The automated management

- aims at optimising the overall resource utilisation
- whilst maintaining the quality constraints.

Source: http://cordis.europa.eu/fp7/ict/ssai/docs/future-cc-2may-finalreport-experts.pdf

User perspective

Clouds are environments
which provide resources and services to the user
in a highly available and quality-assured fashion,
thereby keeping the total cost for usage & administration
minimal and adjusted to the actual level of consumption.

The *resources and services* should be accessible

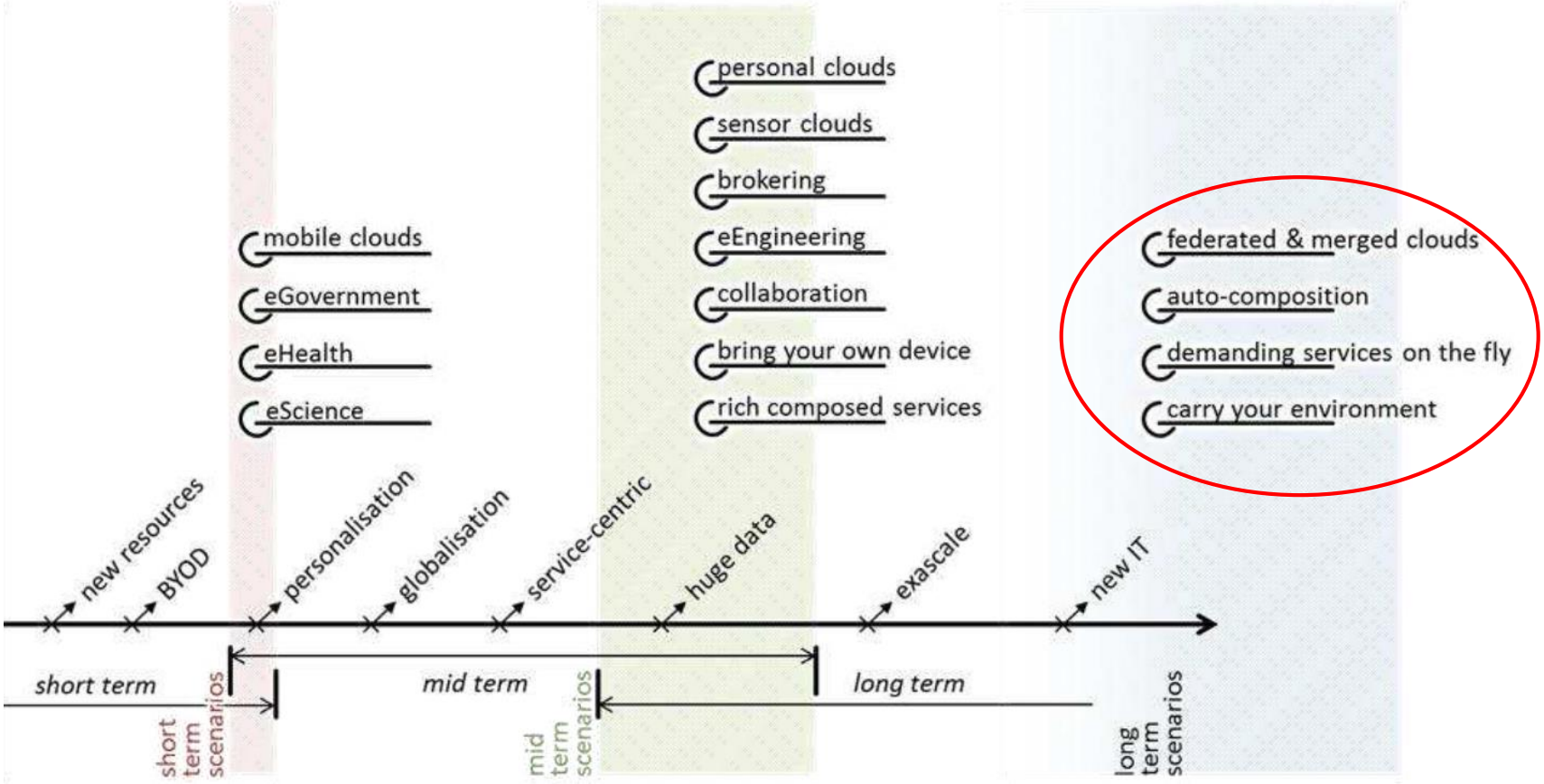
- for theoretically unlimited no. customers
- from different locations and
- with different devices
- with minimal effort and minimal impact on quality.

The *environment* should adhere to security and privacy regulations of the end-user, in so far as they can be met by the internet of services.

Source: http://cordis.europa.eu/fp7/ict/ssai/docs/future-cc-2may-finalreport-experts.pdf

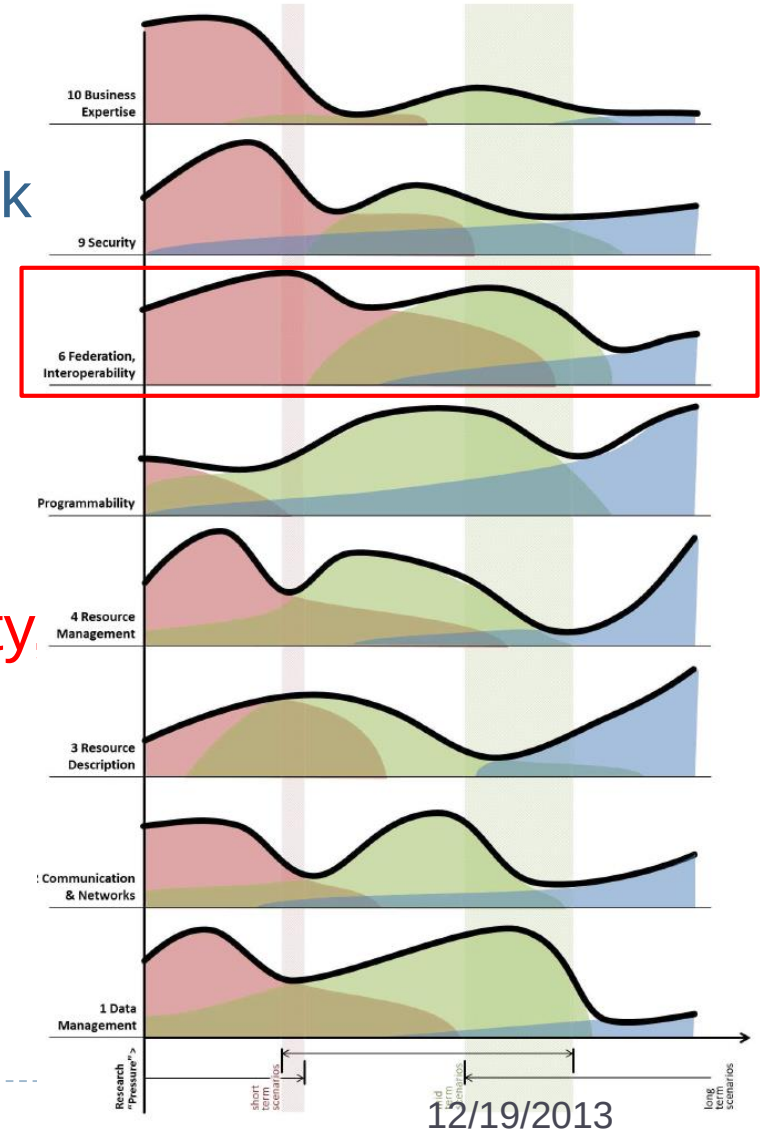
Expectations in terms of use cases

Source: http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-expert-group/roadmap-dec2012-vfinal.pdf



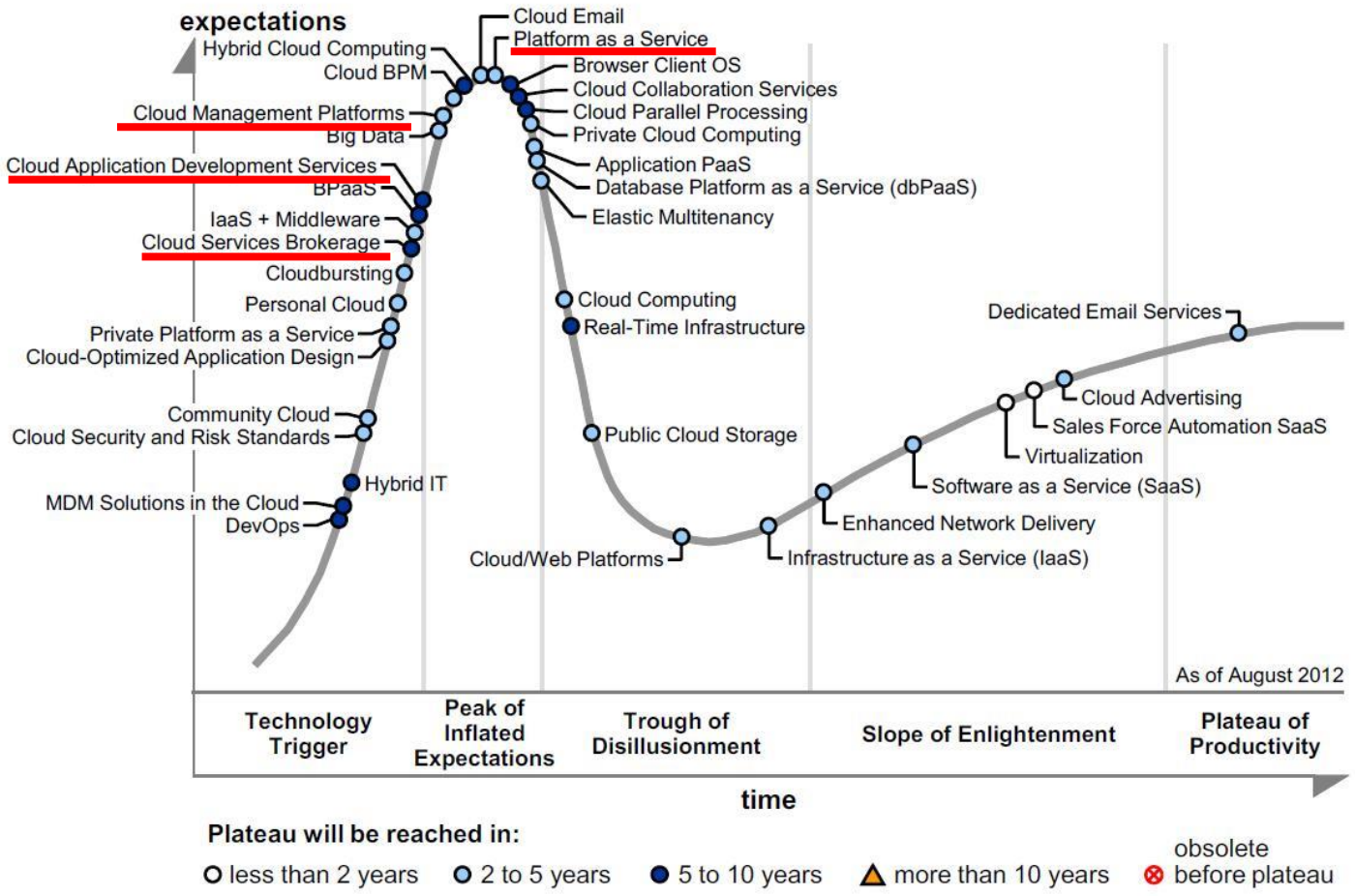
Main Topics to Address

1. Data Management
2. Communication & Network
3. Resource Description & Usage
4. Resource Management
5. Programmability and Usability
6. Federation, Interoperability Portability
7. Multiple Tenants
8. Political & Legislatory
9. Security
10. Business & Cost Models

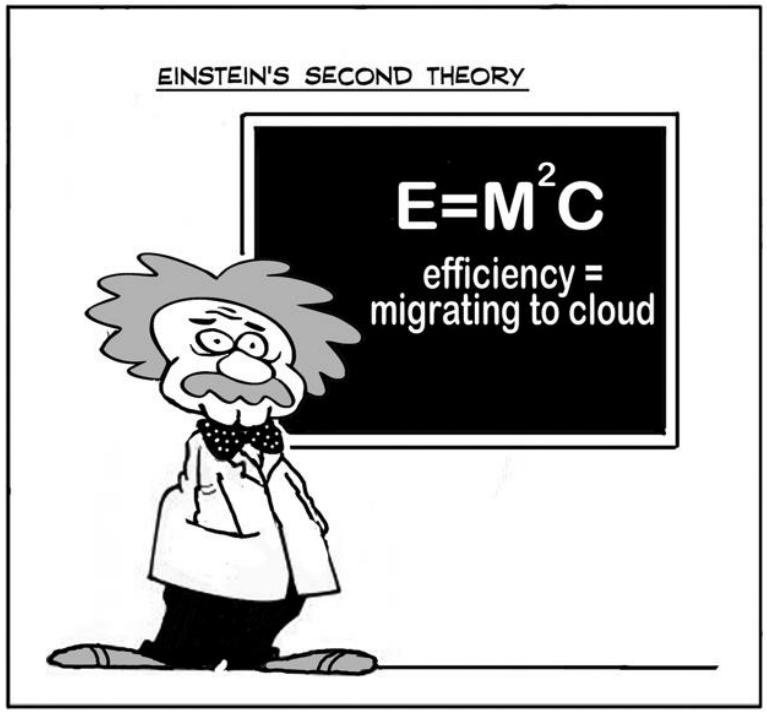


Topics of interest vs. Gartner Report

Figure 1. Hype Cycle for Cloud Computing, 2012



Source: Gartner (August 2012)



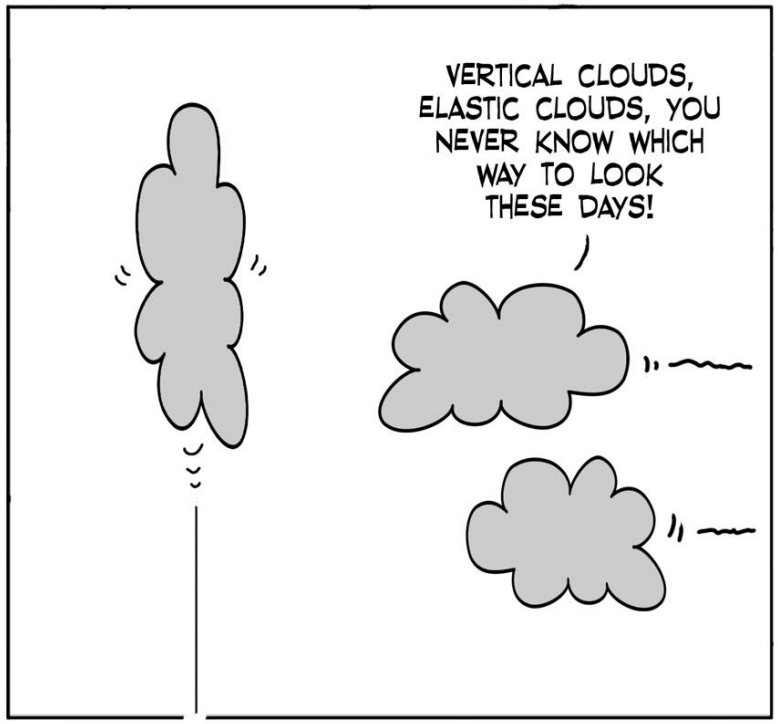
Why Multiple Clouds?

NIST scenarios: Multiple Clouds

- **Clouds can be used**
  1. serially, when moved from one Cloud to another, or
  2. simultaneous, when using services from different Clouds.
- **Simple scenarios:**
  1. [serial] migration from a Private Cloud to a Public Cloud
  2. [simultaneous] Hybrid Cloud, when some services are lying on the Private Cloud, while other services are lying on a Public Cloud

Top 10 Reasons for Multiple Clouds

1. deal with the peaks in service & resource requests using external ones, on demand basis;
2. optimize costs or improve quality of services;
3. react to changes of the offerts of the providers;
4. follow the constraints, like new locations or laws;
5. replicate the applications or services consuming resources or services from different Cloud providers to ensure their high availability;
6. avoid the dependence on only one external provider;
7. ensure backup-ups to deal with disasters or scheduled inactivity;
8. act as intermediary;
9. enhance own Cloud resource and service offers, based on agreements with other providers;
10. consume different services for their particularities not provided elsewhere.



Taxonomy of Multiple Clouds

Terminology

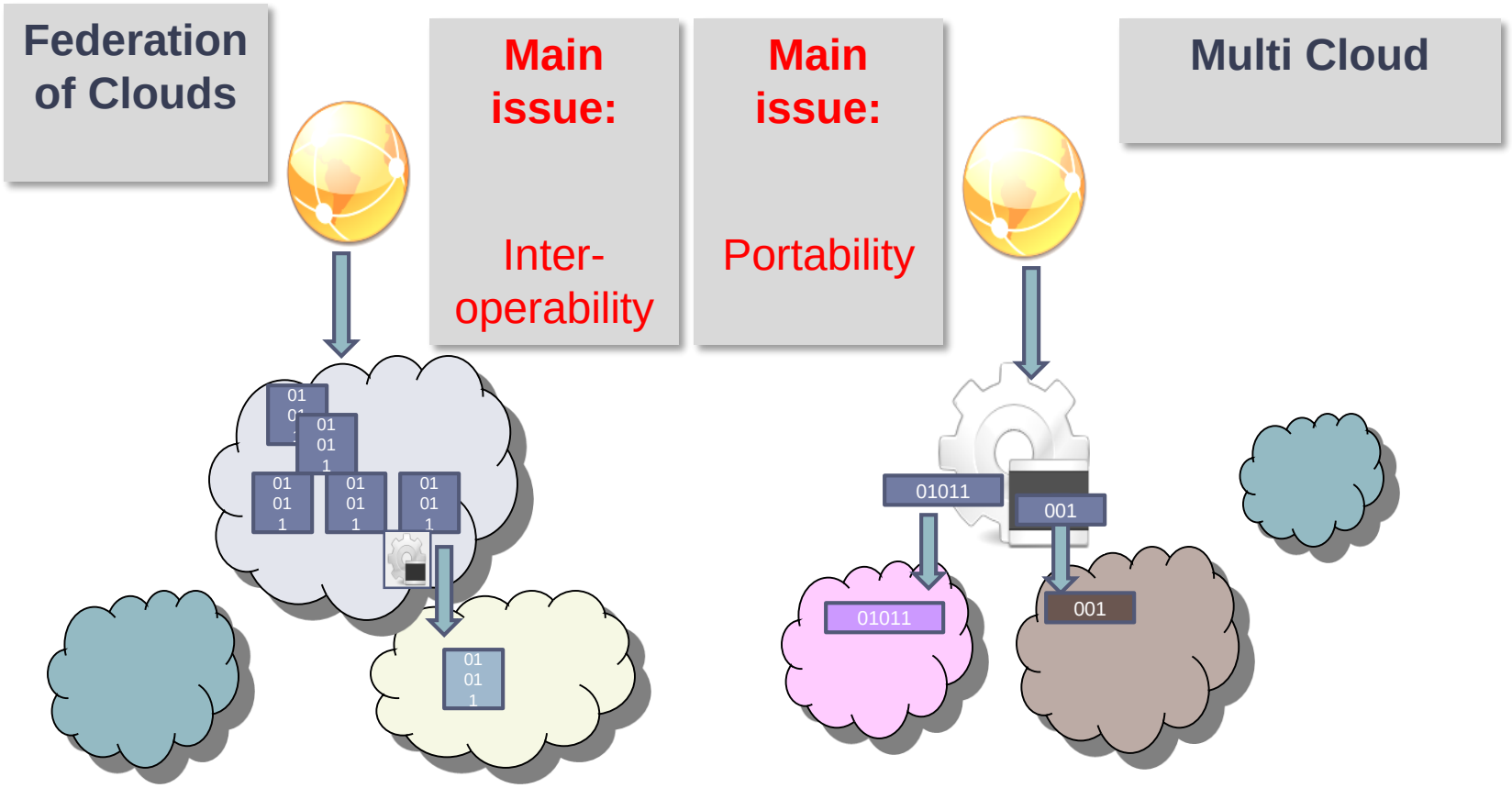
- Multi-Cloud,
- Cloud Federation,
- Inter-Cloud,
- Hybrid Cloud,
- Cloud-of-Clouds,
- Sky Computing,
- Aggregated Clouds,
- Multi-tier Clouds,
- Cross-Cloud,
- Cloud Blueprint,
- Cloud Merge,
- Fog Computing,
- Hierarchical Clouds,
- Distributed Clouds
- ...

Delivery models for Multiple Clouds

1. **Federated Clouds**
   - assumes
     - a formal agreement between the Cloud providers
   - service providers
     - are sub-contract capacity from other service providers
     - offer spare capacity to the federated group of providers.
   - the consumer of the service
     - is not aware of the fact that the Cloud provider he or she pays is using the services of another Cloud provider
2. **Multi-Cloud**
   - assumes that
     - there is no priori agreement between the Cloud providers
   - a 3rd party (even the consumer) is responsible for the services
     - contacts the service providers,
     - negotiates the terms of service consumption,
     - monitors the fulfillment of the service level agreements,
     - triggers the migration of codes, data and networking from one provider to another.

Source: http://www.buyya.com/papers/InterCloud-Brokering-Taxonomy.pdf

Scenarios for multiple Clouds



To solve in Cloud Federation

Federations

- Interoperability framework
- Integration as a service
- Match-making with available external services
- Live virtual machine migration
- Network overlay for connectivity problems
- Meta-schedulers
- Monitoring meta-system
- Intelligent management systems
- ...

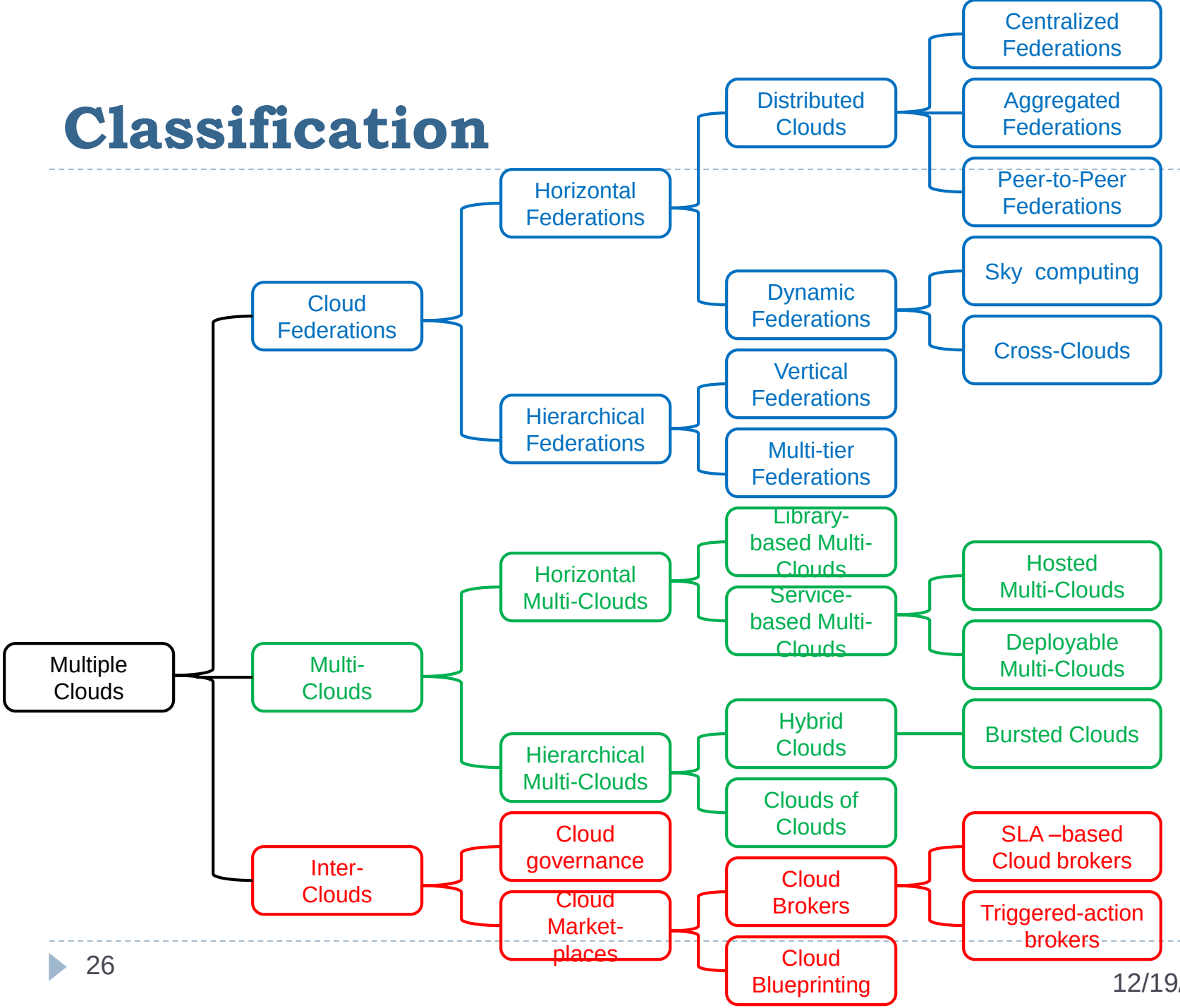
Multi-Cloud

- Portability
- Resource/service selection mechanism and methodology
- Uniform APIs
- Search engines
- Automated deployment
- Service aggregator
- Governance
- ...

InterCloud, Cloud Broker & Blueprint

- **InterCloud:**
  - A Cloud Federation or a Multi-Cloud that includes at least one Cloud Broker and offers dynamic service provisioning
- **Cloud Broker**
  - an entity that manages the use, performance and delivery of Cloud services and intermediates the relationships between Cloud providers and Cloud consumers
- **Cloud Blueprint**
  - an enhanced Cloud delivery model,
  - a reference architecture transforms Cloud stack into modular and easily combinable components that offer Integration-as-a-service functionality

Classification



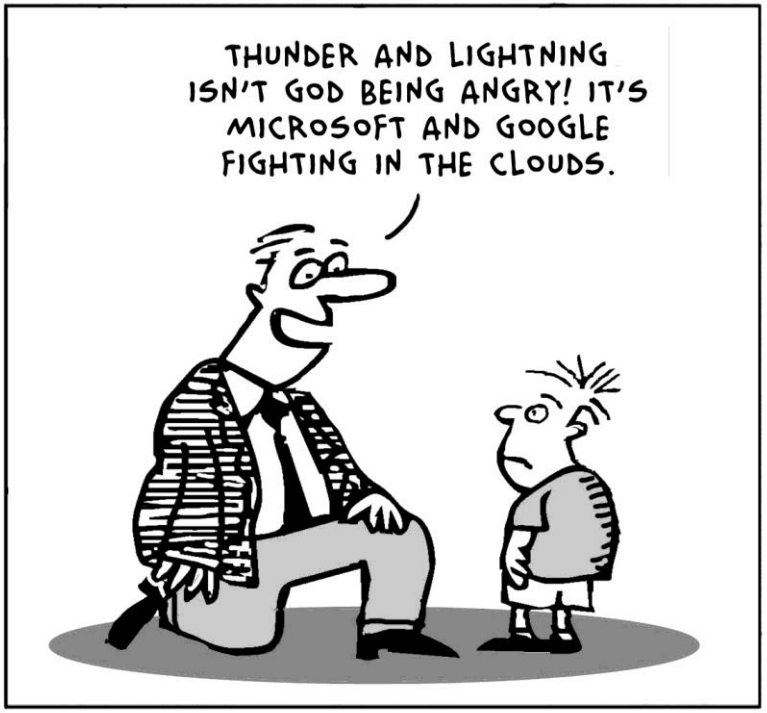
D. Petcu, Consuming Resources and Services from Multiple Clouds. From Terminology to Cloudware Support, J. Grid computing, to appear

Requirements/ Multi-Cloud

	Development	Deployment			Execution
Tools	Portal/service as entry point	Service/ resource meta-allocator	Generic deployer	Search engine	Meta-monitor for applications
	Cloud agnostic extra services	Meta-scheduler	Semi-automated deployer	Match-making service	Meta-monitor for services/ resources
	Interface for user's requirements	Meta- auto-scaler and load-balancer	Virtual network mechanisms	Selection service	Controller of application/service life-cycle
	Integration service	Debugger and tester	Credentials management	Recommen-dation system	QoS control and warning mechanisms
Principles	Portability support	Particularities preservation	Seamless join by new Clouds	No constraints on Clouds	Use standard protocols
	Abstract service con-trol interfaces	Use standard interfaces	Support for top Cloud providers	Allow dynamic allocation of resources	Small overhead

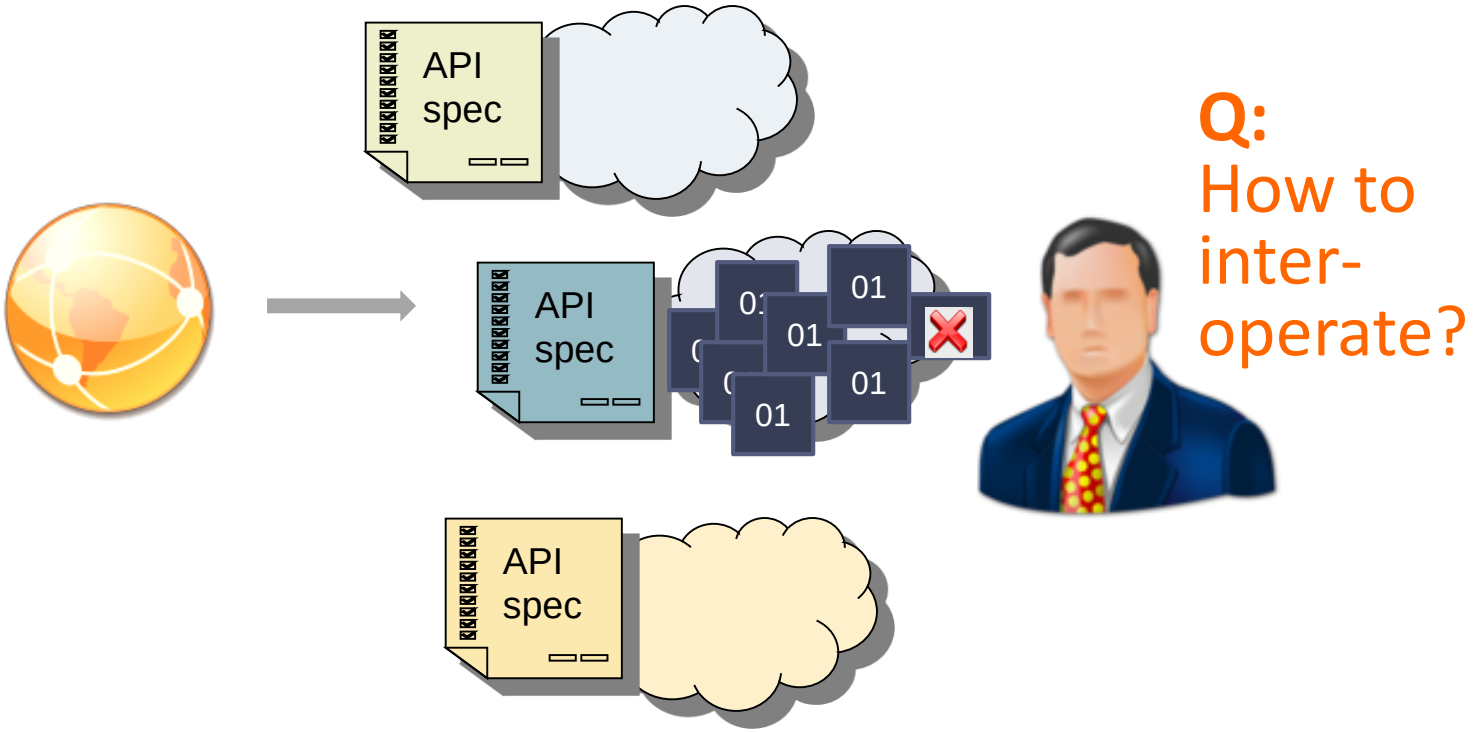
Middleware

Delivery model	Organization	Type	Architecture	Middleware examples
Federation	Horizontal	Distributed	Centralized	BonFIRE, ConPaaS
			Aggregated	OpenNebula
			Peer-to-Peer	OpenCirrus
		Dynamic	Sky computing	Nimbus
			Cross-Clouds	Xen-Blanket
	Hierarchical	Vertical		
		Multi-tier		
Multi-Cloud	Horizontal	Library-based		jclouds, Libcloud, δ-Cloud, SimpleCloud, SAGA
		Service-based	Hosted	RightScale, Kaavo
			Deployable	mOSAIC, Cloud4SOA, Optimis, Aoleus, MODAClouds
	Hierarchical	Hybrid Cloud	Bursted Cloud	StratusLab, Agility
		Clouds of Clouds		TClouds
Inter-Cloud	Governance			Enstratius
	Marketplace	Cloud brokers	SLA-based	SpotCloud, Stratos, CloudBroker
			Trigerred-action	Scalr
		Blueprint		4CaaST



Interoperability and portability

Interoperability in Clouds?

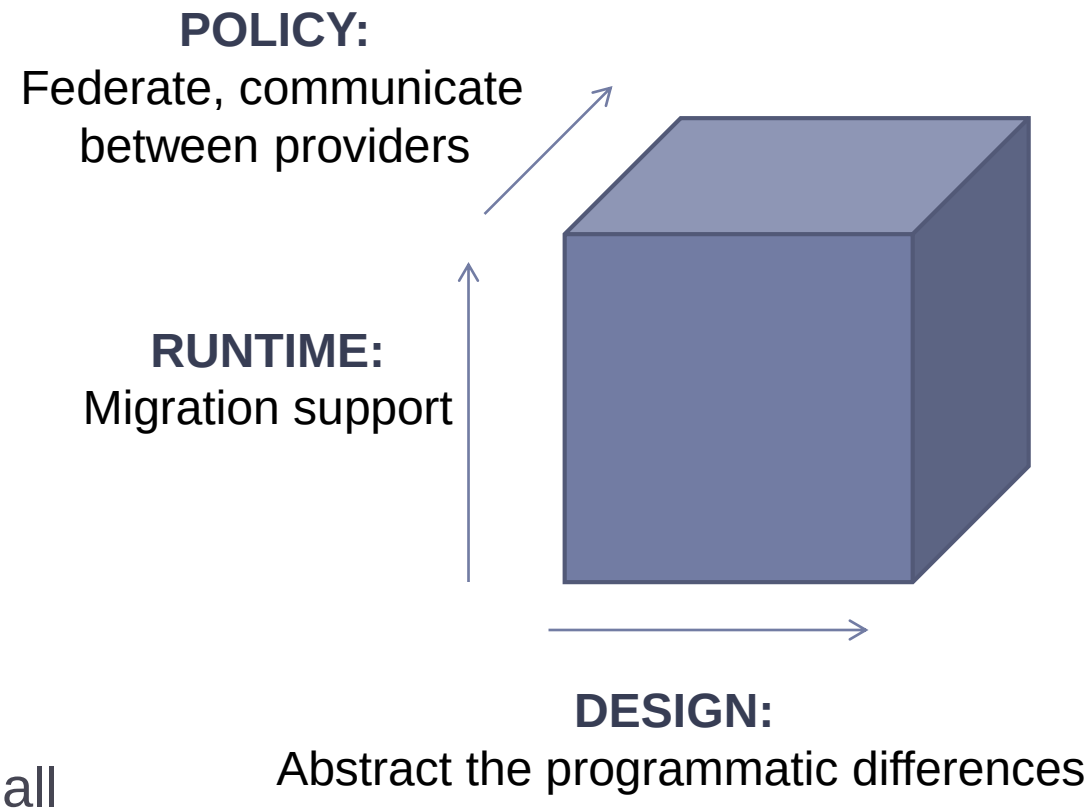


Interoperability/Clouds- history

1. Migration – targets VMs
   - Create, import, share VMs (e.g. use OVF)
2. Federation – targets networking
   - Portable VMs moved between clouds and hypervisors without reconfiguring anything
3. On-demand (burst) – targets APIs
   - Migration and federation on demand
   - Interoperability focused on storage and compute (e.g. CDMI, OCCI)

Interoperability definition & dimensions

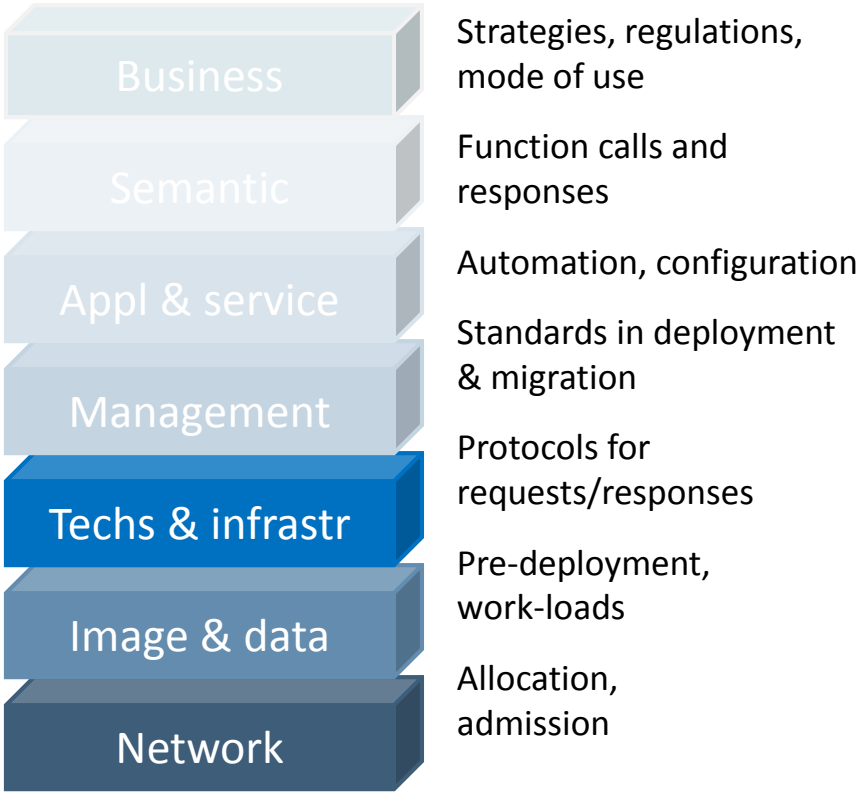
- **Dictionary:**
  - Property referring to the ability of diverse systems to work together
- **By mottos:**
  - avoid vendor lock-in
  - develop your application once, deploy anywhere
  - enable hybrid clouds
  - one API to rule them all



D. Petcu, Portability and Interoperability between Clouds: Challenges and Case Study, ServiceWave 2011

Current solutions

Levels



E.g.

Strategies, regulations, mode of use

Function calls and responses

Automation, configuration

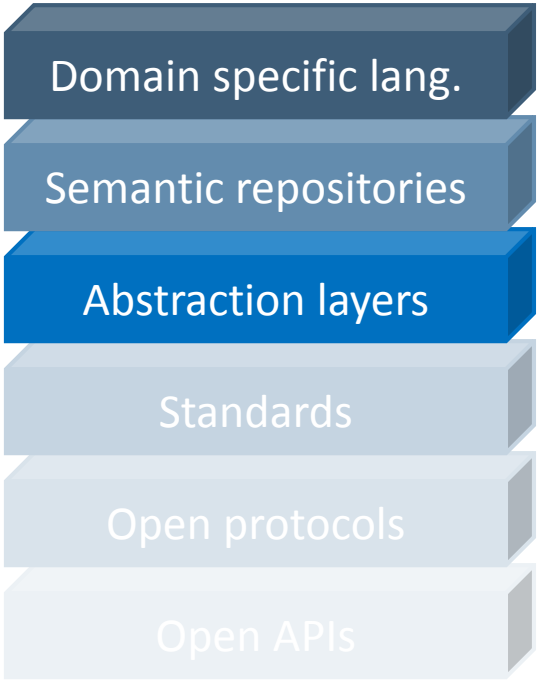
Standards in deployment & migration

Protocols for requests/responses

Pre-deployment, work-loads

Allocation, admission

Techs



E.g.

Automated translation in code

UCI

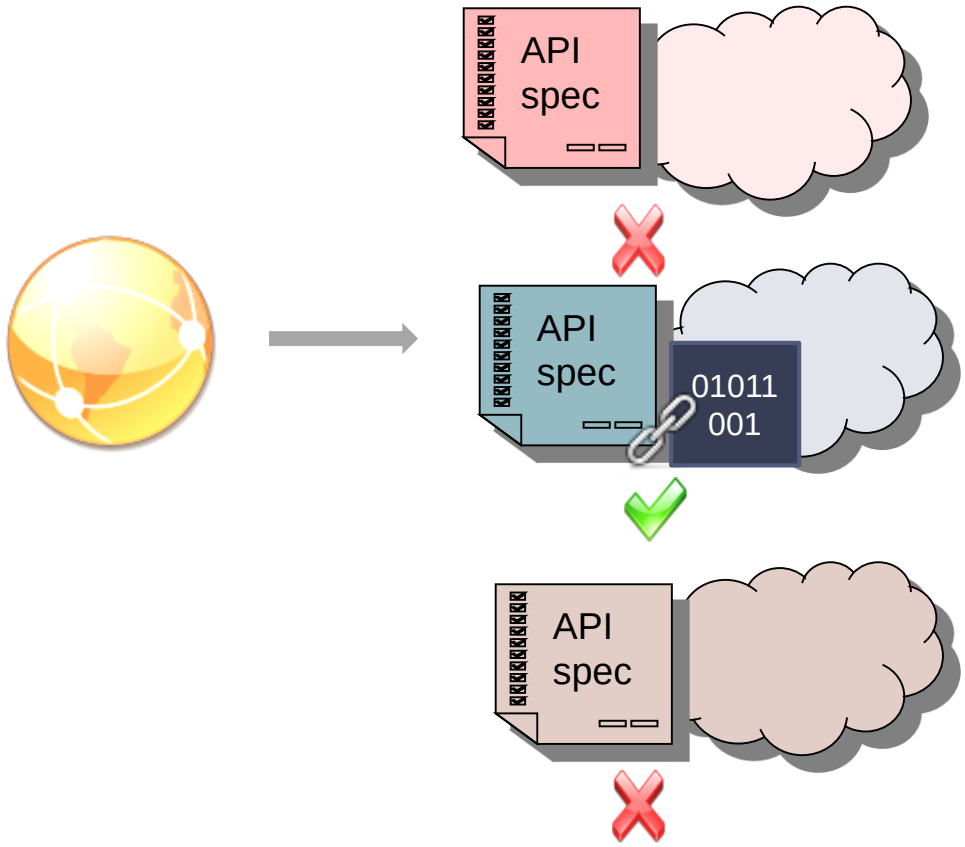
Mediators, frame-works (SLA@SOI)

OVF/DMTF, CDMI/SNIA

OCCI, Deltacloud

jClouds, libcloud, OpenStack

Portability in Clouds?

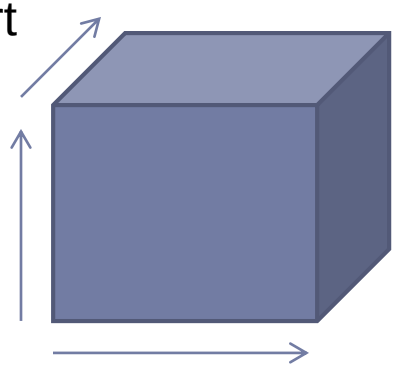


Q:
How to port the appl?

Portability between Clouds

- Ability to use components or systems lying on multiple hardware or software environments
- Dimensions:

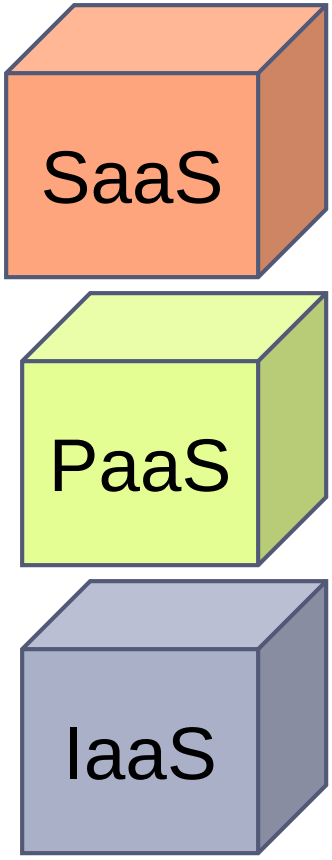
DATA:
Import & export functionality



SERVICE:
On the fly add, reconfig and remove resources

FUNCTION:
Define appl. functionality in platform-agnostic manner

Portability at XaaS level

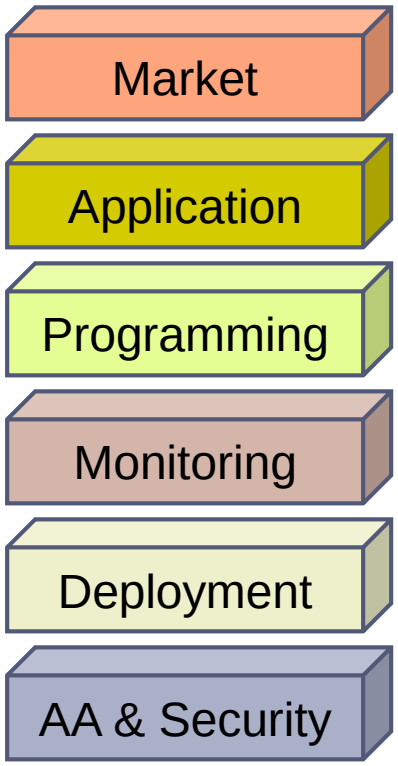


Preserve/enhance functionality when substitute softw
Measures:
- open source; proprietary/open formats;
- integration techs; appl server/OS

Minim.appl.rewriting while preserve/ enhance control
Measures:
- proprietary vs.open APIs, progr.languages,data formats
- tight vs. loose coupled services
- abstract layers for queuing & messaging

Appls and data migrate and run at a new provider
Measures:
- ability to port VMs and data
- underlying configurations across providers

Requirements for portability



Economic models, cost-effectiveness, license flexibility, negotiated SLAs, leasing mechanisms

Data portability and exchange, scale-out, location-free, workflow management

Minimal reimplementation when move, standard APIs, same tools for cloud-based and entreprise-based appls

SLA and performance monitoring, QoS aware services, service audit, sets of benchmarks

Deploy in multiple clouds with single management tool, navigation between services, automated provisioning, resource discovery and reservation, behavior prediction

Single sign-on, digital identities, security Standards, trust mechanisms, authentication

mOSAIC

Open source ApI & Platform for Multiple Clouds

mOSAIC

marketing motto: “Flying through the Clouds”

1. a tool for developing portable Cloud-applications which can consume hardware and software resources offered by multiple Cloud providers;
2. a brokerage system to support the decision of Cloud service provider selection at the deployment stage.
3. an open-source PaaS that can be easily deployable by service providers and which can be customized and enhanced by service providers;

mOSAIC as R&D collaboration effort

www.mosaic-cloud.eu

Consortium:

1. Second University of Naples, Italy
2. Institute e-Austria Timisoara, Romania
3. European Space Agency, France
4. Terradue SRL, Italy
5. AITIA International Informatics, Hungary
6. Tecnalia, Spain
7. Xlab, Slovenia
8. University of Ljubljana, Slovenia
9. Brno University of Technology, Czech Republic

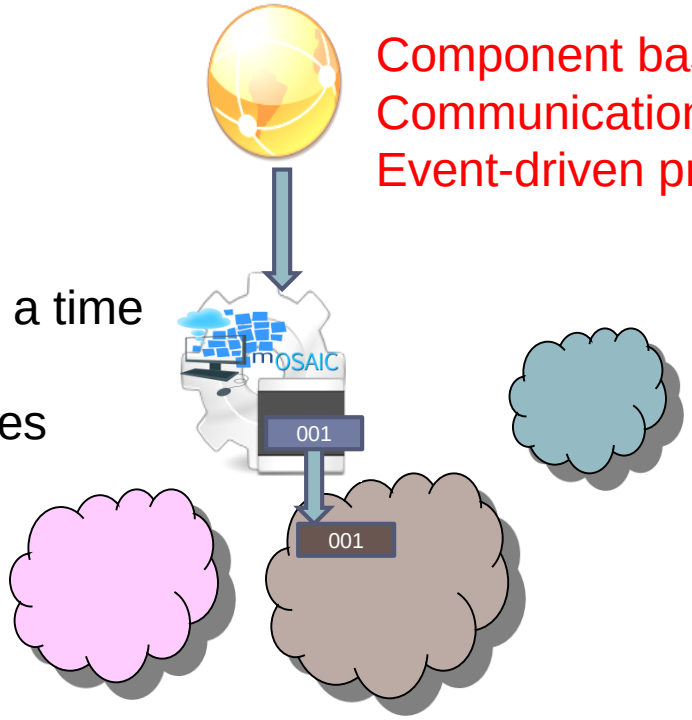
European Commission

September 2011: 1st API implementat. (Java)
September 2012: 1st stable PaaS,
2nd API impl. (Python)
March 2013: Full software package

Scenario for multiple Clouds

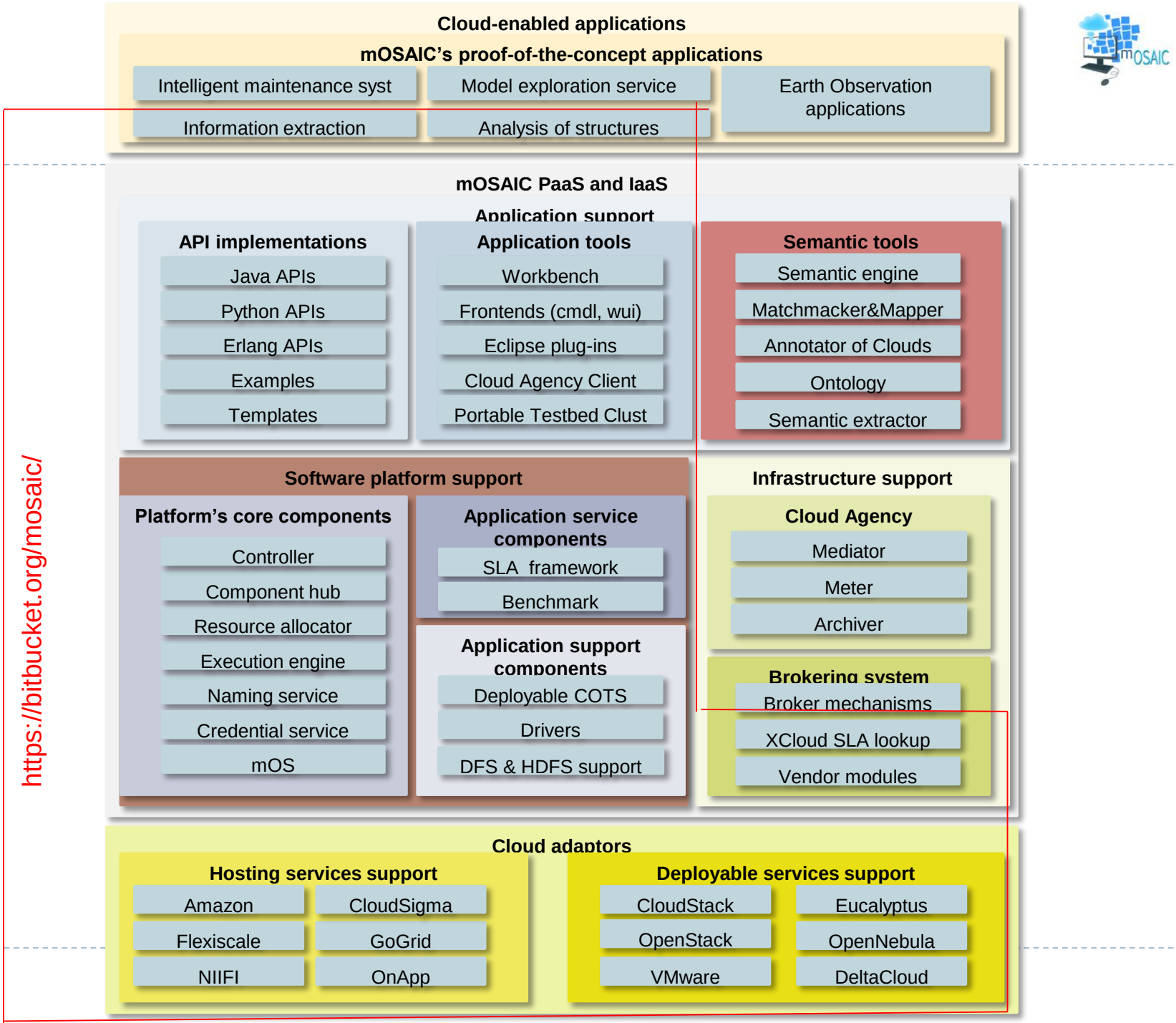
Component based application
Communications via messages
Event-driven programming style

Select 1 IaaS-Cloud at a time
Deploy
Control appl & resources



Overview paper:

Petcu et al, *Experiences in Building a mOSAIC of Clouds*
Journal of Cloud Computing: Advances, Systems and Applications 2013, 2:12
doi:10.1186/2192-113X-2-12, May 24, 2013
http://www.journalofcloudcomputing.com/content/2/1/12/abstract


Cloud-enabled applications
mOSAIC's proof-of-the-concept applications
Intelligent maintenance syst
Model exploration service
Earth Observation applications
Information extraction
Analysis of structures
mOSAIC PaaS and IaaS
Application support
API implementations
Java APIs
Python APIs
Erlang APIs
Examples
Templates
Application tools
Workbench
Frontends (cmdl, wui)
Eclipse plug-ins
Cloud Agency Client
Portable Testbed Clust
Semantic tools
Semantic engine
Matchmacker&Mapper
Annotator of Clouds
Ontology
Semantic extractor
Software platform support
Platform's core components
Controller
Component hub
Resource allocator
Execution engine
Naming service
Credential service
mOS
Application service components
SLA framework
Benchmark
Application support components
Deployable COTS
Drivers
DFS & HDFS support
Infrastructure support
Cloud Agency
Mediator
Meter
Archiver
Brokering system
Broker mechanisms
XCloud SLA lookup
Vendor modules
Cloud adaptors
Hosting services support
Amazon
CloudSigma
Flexiscale
GoGrid
NIIFI
OnApp
Deployable services support
CloudStack
Eucalyptus
OpenStack
OpenNebula
VMware
DeltaCloud
https://bitbucket.org/mosaic/

How to use it?

- Write component-based application
  - Languages: Java, Python, [Erlang, Node.js]
  - Communications through message passing
  - Respect the event-driven style of programming
- Debug application on the desktop or on-premise server(s)
  - Within Eclipse
  - Use Personal Testbed Cluster using VirtualBox for the VMs
- Deploy application in a Cloud
  - Assisted by Cloud Agency and Broker (with SLAs) OR
  - Use Resource Allocator
- Control the application
  - Control the life-cycle of the components (start/stop/replace)

Tutorial & Documentation

- Tutorial for the installations and first example:

  http://wiki.volution.ro/Mosaic/Notes/Platform/Tutorial

- Documentation:

  http://developers.mosaic-cloud.eu

- Application videos & links
  - Civil engineering (Matlab @ Cloud): http://youtu.be/EztdyThs39w
  - Earth Observation (ESA&Terradue): https://vimeo.com/64316032
  - Model exploration: http://youtu.be/fU8VONfg6Z0
  - Information extraction on the open-source repository
  - Sensor data in the Intelligence Maintenance use case
  - Olaii (www.olaii.com) RightScale + Amazon

Simple example: Videos

- **Hello example (one Cloudlet):**
  - Hello run and debug on PTC: http://youtu.be/pDrktFOMZWA
  - Hello run on AWS: http://youtu.be/GW1WjZhJXH8

- **Real time feeds example**
- **(multiple Components & Cloudlets):**
  - Deploy manually component by component in AWS: http://youtu.be/uYD8sxMStz8
  - Package send to S3 and start of the appl: http://youtu.be/AK1LqAMjvfU

Tools Videos

- **Use PTC:**
  - How to start the [desktop] platform: http://youtu.be/TPHHXg1ggvU
  - How to start application on AWS: http://youtu.be/oGf2wDce-sk
- **Vendor selection:**
  - Vendor offers: http://youtu.be/T54qh0cWroY
  - XCloud: http://youtu.be/r3kXeBHSVF8
- **SLA**
  - negotiation: http://youtu.be/3X5Kih-Oi6E
  - SLAgw & security: http://youtu.be/ZKcWhl1WG14
  - Usage of Cloud Agency & OpenNebula: http://youtu.be/6SBGYc7fCWA
- **Benchmarks**
  - http://developers.mosaic-cloud.eu/confluence/display/MOSAIC/Benchmarks
  - Shell scripts: CPU, Network, Message queues, Data stores

MODAClouds

Model-Driven Engineering for Clouds

MODAClouds objective

- ***provide***
  - methods,
  - a decision support system,
  - an IDE and
  - a runtime environment
- ***to support***
  - High-level design
  - Early prototyping
  - Semi-automatic code generation
  - Automatic (re)deployment
  - Monitoring and self-adaptation

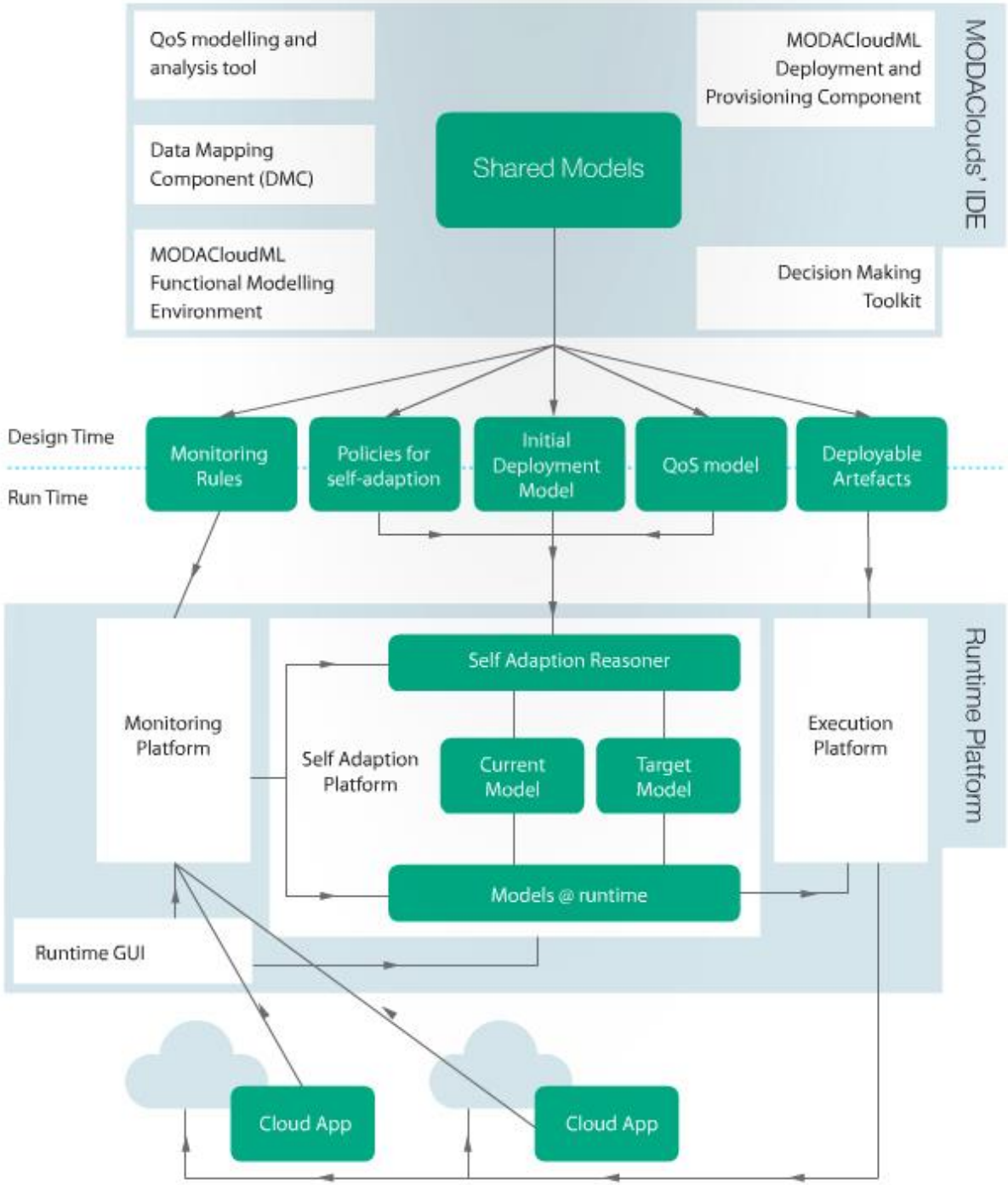
of applications on multi-Clouds with guaranteed QoS

MODAClouds (www.modaclouds.eu)

- Integrated Project n. 318484
- October 1st 2012 – September 30th 2015

Architecture

http://www.modaclouds.eu/publications/public-deliverables/



MODAClouds

Software

www.modaclouds.eu/software

for:

- Docs
- Source codes
- Videos

Design-time components

- MODAClouds IDE
  - Functional Modeling Tools
  - QoS Modelling and Analysis Tools
    1. LINE
    2. Space4Cloud
- MODACloudML

Run time components

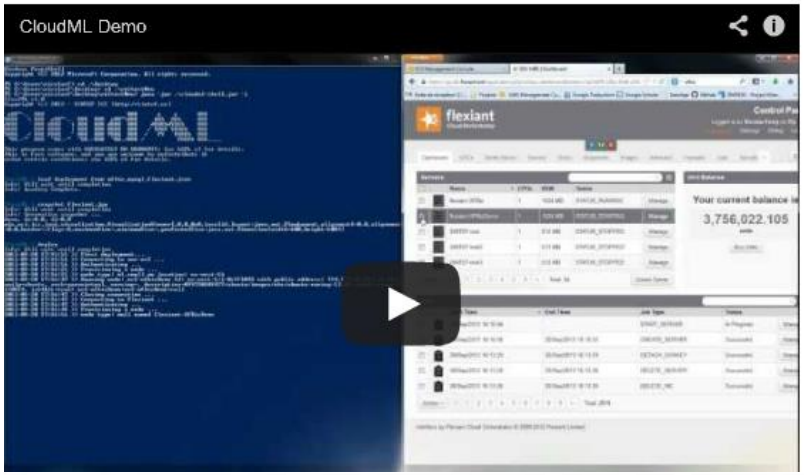
- Monitoring Platform
- Execution Platform
  1. Updates of mOSAIC open-source PaaS
  2. Cloud4SOA

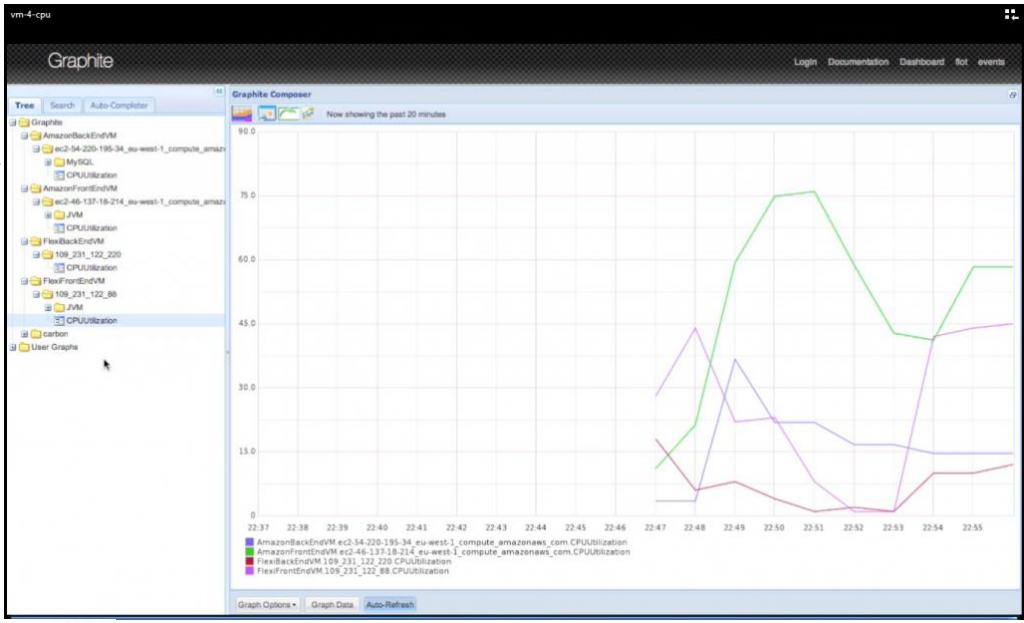
Re-used components

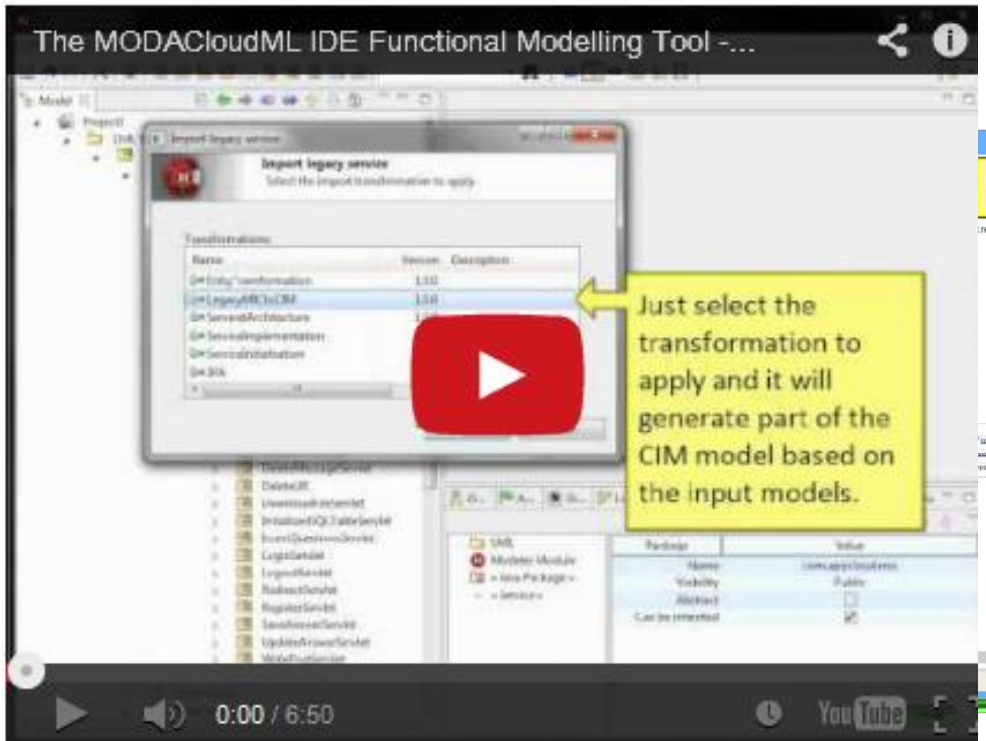
- Modelio
- Paladio

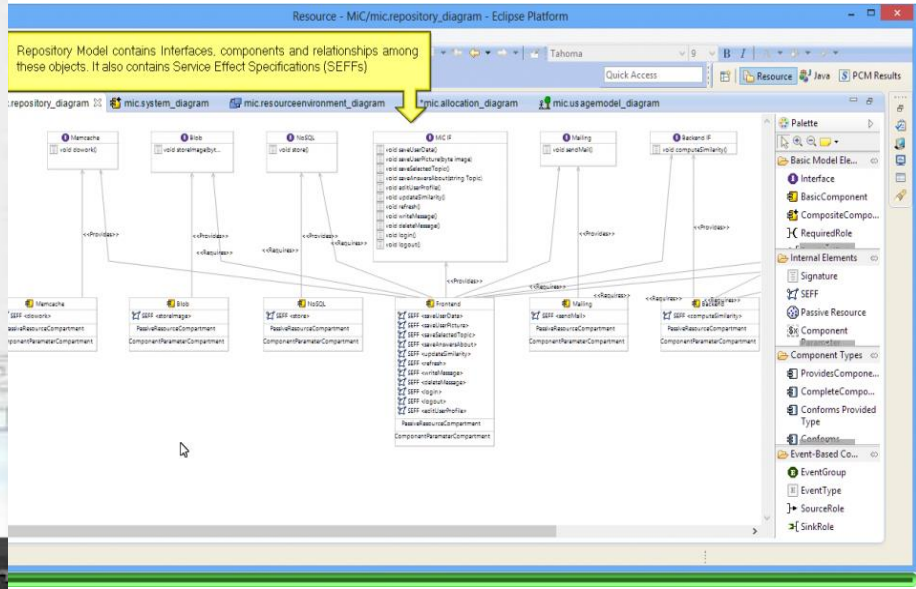
Snapshots

MODAClouds





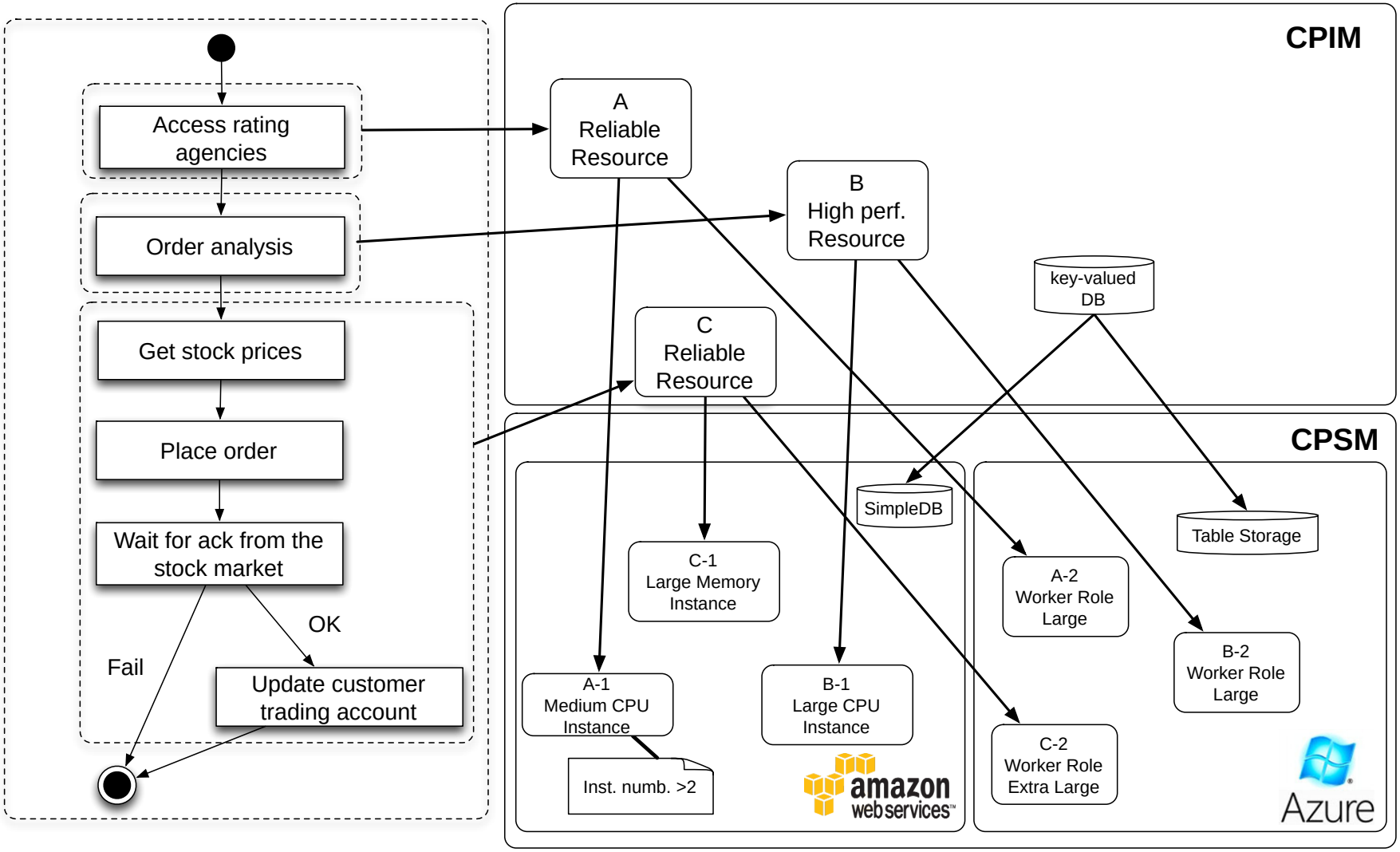


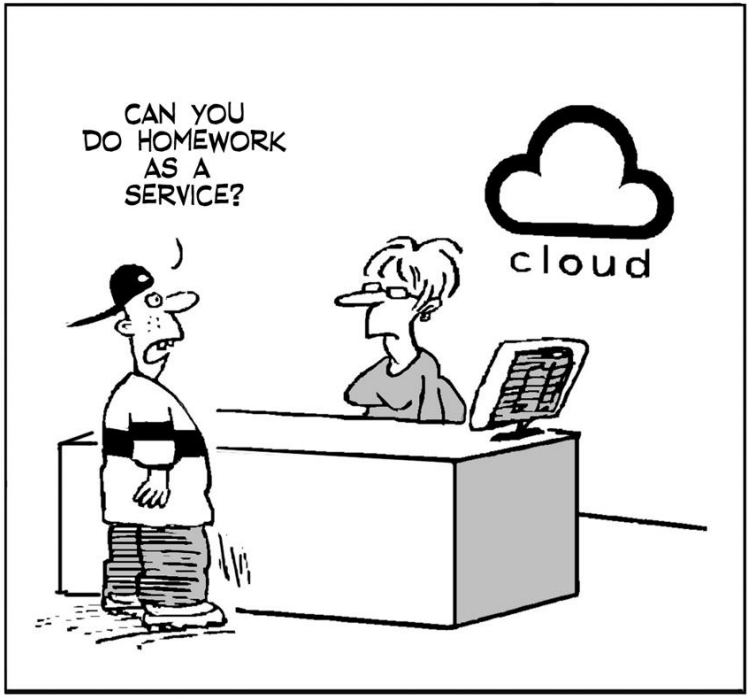


An example

D. Ardagna et al. MODACLOUDS: A Model-Driven Approach for the Design and Execution of Applications on Multiple Clouds. Procs. MiSE 2012

MODAClouds





SPECS

Secure Provisioning of Cloud Services based on SLA management

SPECS Partners

CeRICT, Italy (coordinator)

TUD, Germany

IeAT, Romania

CSA, United Kingdom

XLAB, Slovenia

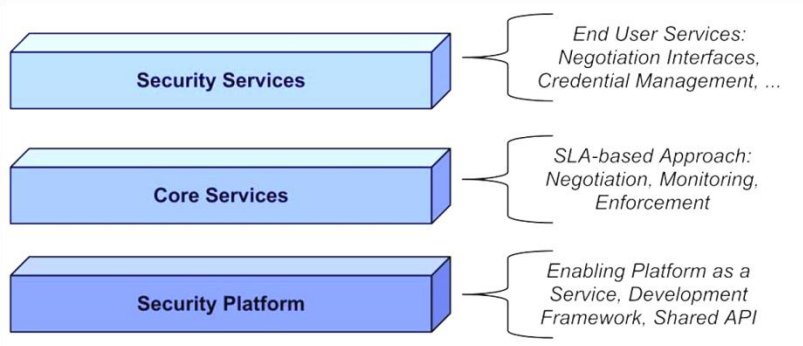
EMC, Ireland

FP7-ICT-10-610795

Project Start: 1/11/2013
Project Type: STREP
Duration: 30M
Total Funding: 3.5 M
EU Contribution: 2.4 M

SPECS aim

- **developing and implementing an open source framework**
  - to offer Security-as-a-Service,
- **by**
  - relying on the notion of security parameters specified in Service Level Agreements (SLA),
  - providing the techniques to systematically manage their life-cycle

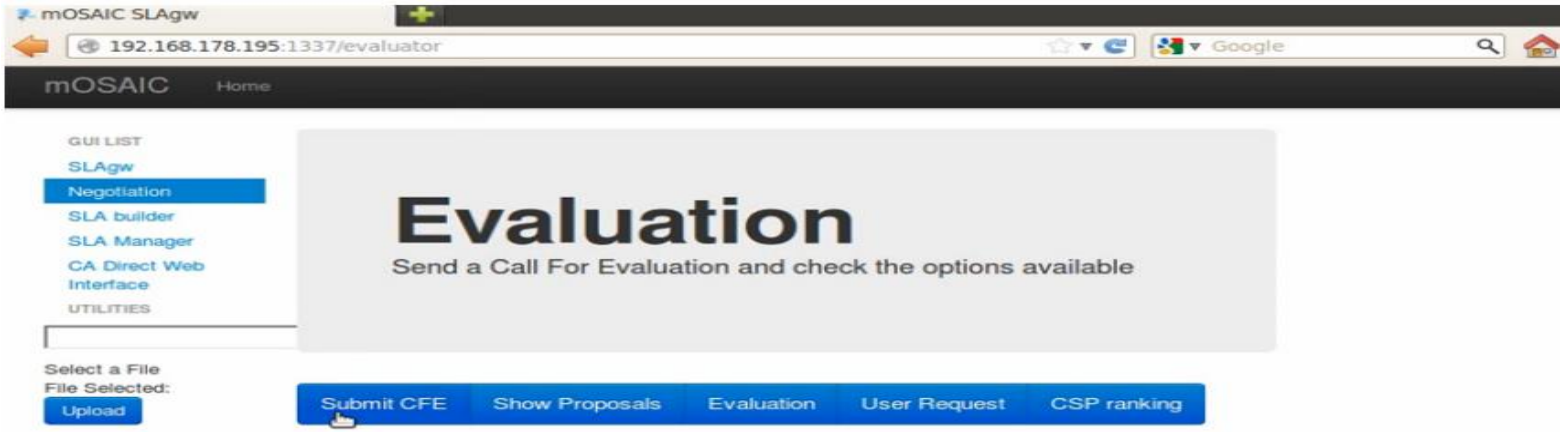


Framework: techniques & tools

1. **Negotiation** of security parameters in Cloud SLA,
   - user-centric
   - along with a trade-off evaluation process among users & CSPs,
   - in order to compose and use Cloud services fulfilling a minimum required security level
     - termed QoSec or Quality of Security in SPECS
2. **Monitoring** in real-time the fulfillment of SLAs
   - SLA agreed with one or more Cloud Service Provider (CSP)
   - enable notifying users &CSPs, when a SLAs not being fulfilled
     - e.g., due to a cyber-attack
3. **Enforcing** agreed Cloud SLA
   - in order to keep a sustained QoSec that fulfills the specified security parameters
   - framework reacts/adapts in real-time to fluctuations in QoSec
     - by advising/applying the correct countermeasures
     - e.g., triggering a two-factor authentication mechanism

A preview

- SLAgw & security: http://youtu.be/ZKcWhl1WG14



Conclusions

- Communities' high interest in tools/middleware
  - to support the easy consumption of Clouds resources
  - will continue in the next half decade
- Multi-Clouds
  - Challenges related to the heterogeneity of the services
  - Multiple emerging solutions from research & industry
- Open problems
  - Standards, protocols
  - Reliability, trust, security, verification
  - Automated management, self-adaptivity

Q&A