Overcoming the Cloud heterogeneity:

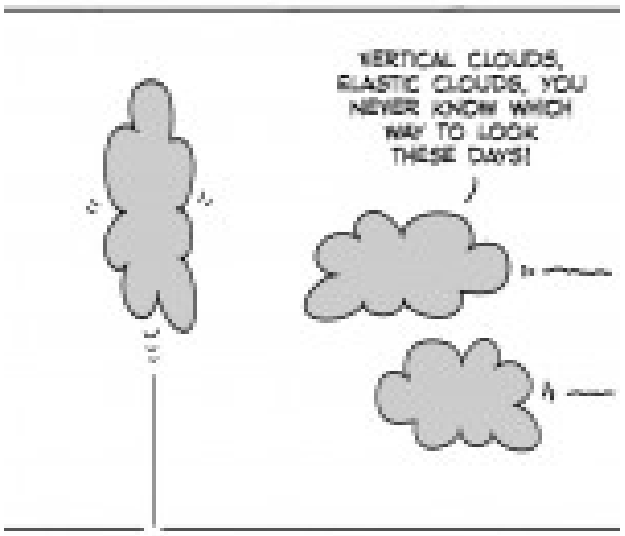
from uniform interfaces and abstract models to multi-cloud platforms

Dana Petcu
http://web.info.uvt.ro/~petcu,
petcu@info.uvt.ro

West University of Timisoara
& Institute e-Austria Timisoara, Romania

Overview

- Cloud heterogeneity
  - The problem
  - The current solutions
    - Examples
    - Advantages and disadvantages
  - Proposal for a research agenda
- Case study
  - Multi-Clouds
  - Model-driven engineering



From Cloud humour site

Solutions to overcome Cloud heterogeneity

New: proposal of a research agenda

Cloud heterogeneity

- Manifested in
  - the set of interfaces of the services from different Public Clouds
  - the set of services from the same provider
  - the software stacks
  - the hardware
  - the terms of performance or user quality of experience
- Favoring
  - the Cloud service providers allowing them to be competitive in a very dynamic market especially by exposing unique solutions
- Hindering
  - the interoperability between these services
  - the portability of the applications consuming the services
  - the seamless migration of legacy applications towards Cloud environments

Types of solutions

1. adoption of standards
   - existing standards
   - emerging standards
2. usage of intermediary layers
   - libraries for major programming languages
   - tools and services
3. adoption of high abstraction levels
   - semantics
   - model-driven engineering

Standards and protocols

Type	Examples
Standard	OCCI, OVF, CDMI, CIMI, TOSCA
Reference architecture	IETF, DMTF
Open groups	OCC, GICTF, CSCC, Open Group
Initiatives	ETSI, IEEE, CSA, NIST, OASIS, OW2, SNIA, TM-Forum

Advantages	Limitations
Result of a collective agreement	Not widely adopted
Extract the key actions and characteristics	From the point of view of the providers, hinders diversity
Should be implementable	No. emerging & overlapping standard makes the problem to grow

Open-source libraries

Type	Examples
Libraries	DeltaCloud, fog, jclouds, libcloud, SimpleCloud

Advantages	Disadvantages
Offers an abstraction layer that is simple to use	Usually refers to the common denominator of the Cloud services
Available for major languages	Language dependent
Similarity with major Cloud provider APIs	Adaptors needs to be build for emerging new services
Decouple the application code from the underlying Cloud service	The connected service programming style usually maintained
Adaptors available for major Cloud services	Require Cloud computing knowledge as deployment is usually not supported
	Introduces an overhead compared with the direct connection to Cloud service

Open-source services or tools

Type	Examples
Services/tools	Aoleus, CompatibleOne, Cloudify, ConPaaS, mOSAIC PaaS, Nimbus, OpenNebula, OpenStack, OPTIMIS

Advantages	Disadvantages
Offer unique entry point for application deployment and Cloud resource management	The diversity of deployment services raise also another dimension for the portability
Application deployment can be done by non-Cloud specialists	Manual intervention at deployment phase is usually still needed
Part of them are offering also APIs for programming applications	Life migration is still not possible
Usually it offers support for multiple programming languages	Re-deployments are not automated
Monitoring tools are generate alarms needed to trigger a redeployment	Rely upon adaptors that need to be build for new services or updated when a service version appears

Semantics

Type	Examples
Semantic solution	UCI, mOSAIC Onto &Sem.engine, Cloud4SOA, CPIM/MODAClouds

Advantages	Disadvantages
Offers an abstraction layer that can support various customers	Not widely adopted
Offers viable mechanisms for common understanding of service terminology and actions	The variety of taxonomies and ontologies makes the problem to grow
Allow the annotation of services with quality marks by externals from the provider team	The overhead of semantic processing is not negligible

Model-driven engineering

Type	Examples
Model-driven	MODAClouds, ARTIST, PaaSage

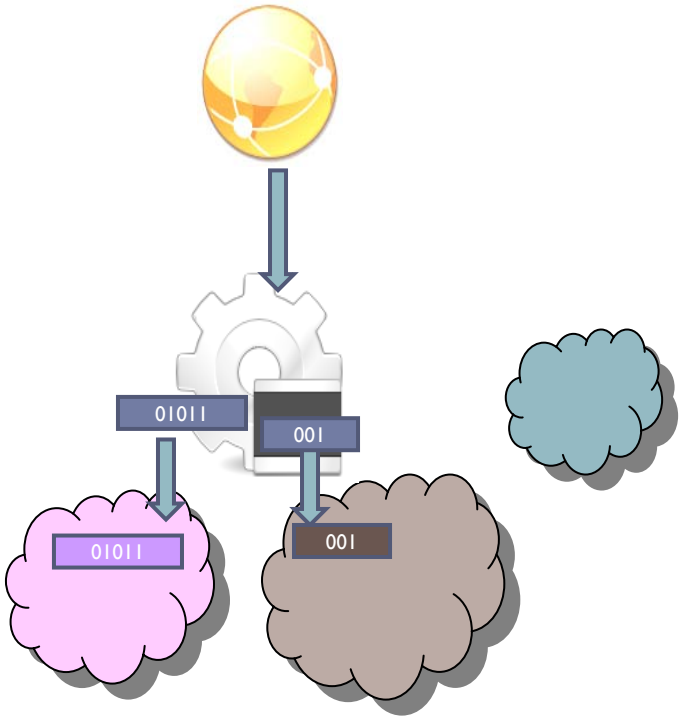
Advantages	Disadvantages
Enhance the abstraction layers with an automation process	Available tools are not yet generating code for various Clouds
Allow a feedback from operational modules to the design modules	The models that are used potentially omit special features of the services

Research agenda – short term

Approach	To Do
Standards	Enhance the number of standard implementations Establish standards for metrics, monitoring, accounting, security Establish standards for machine-readable representations of services, quality, negotiations, processes
Design	Support for decision making for Cloud migration Introduce Modelling-as-a-Service Mechanism for service compositions Build use cases and benchmarks for Cloud portability and interop Define the portability degree
Runtime	Adopt open-source platforms Increase the use of empirical evidence of portability and interop Automate re-deployments

Research agenda – long term

Approach	To Do
Standards	Unified policy of the service level agreements Establish standards for workload and data migration Reference architecture for basic components of software consuming Cloud services
Design	Define re-engineering process for Cloud Mechanisms for code inspections and rewriting Follow a structural approach in the design of the supporting tools Ensure the portability of elasticity rule engines Combine automation with customization
Runtime	Mechanisms for real-time migration Tools for the full service cycle, including Cloud governance Open-source platforms ensuring automated portability or encompassing various approaches



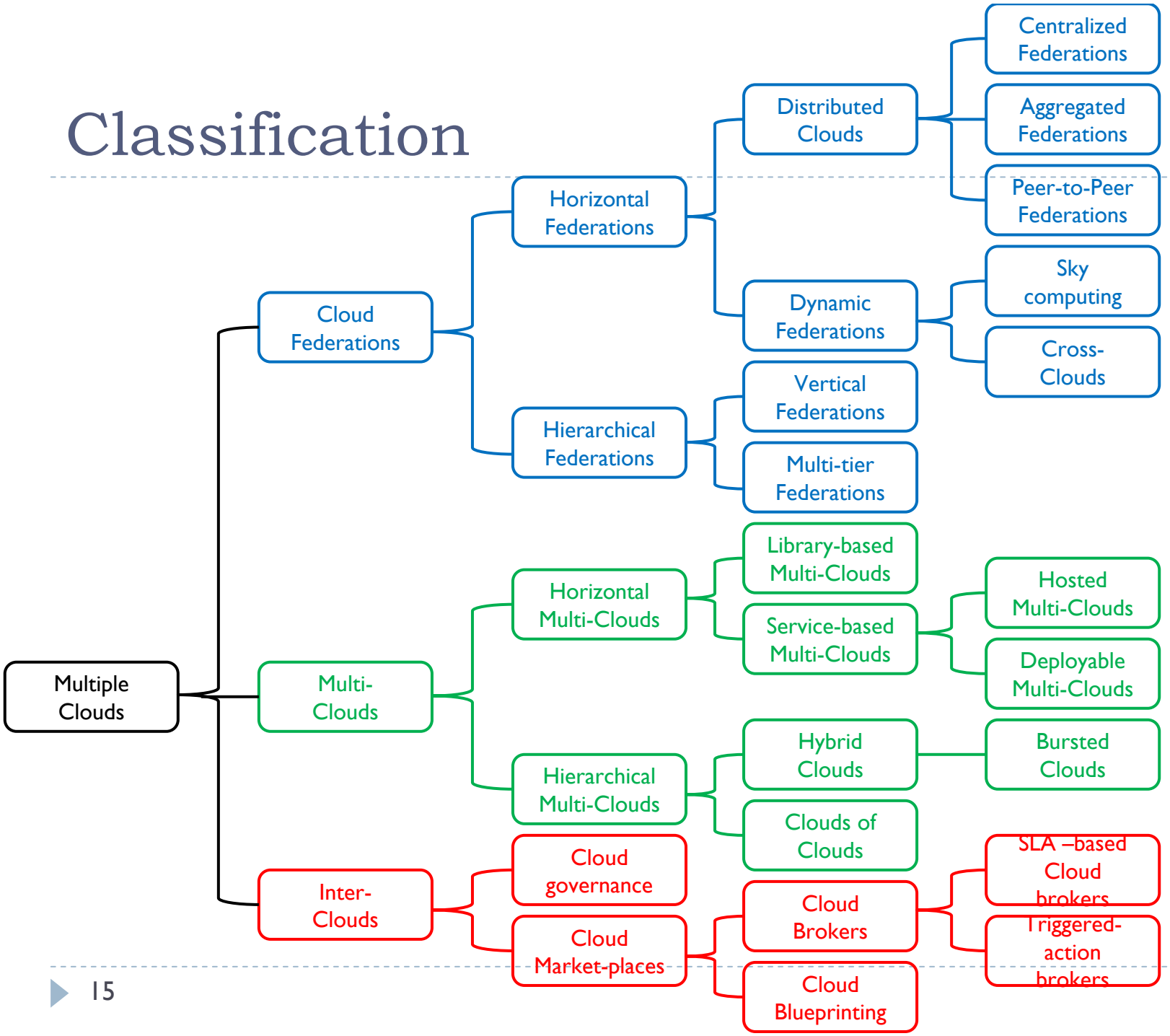
Case study: Multi-Clouds

New: classification

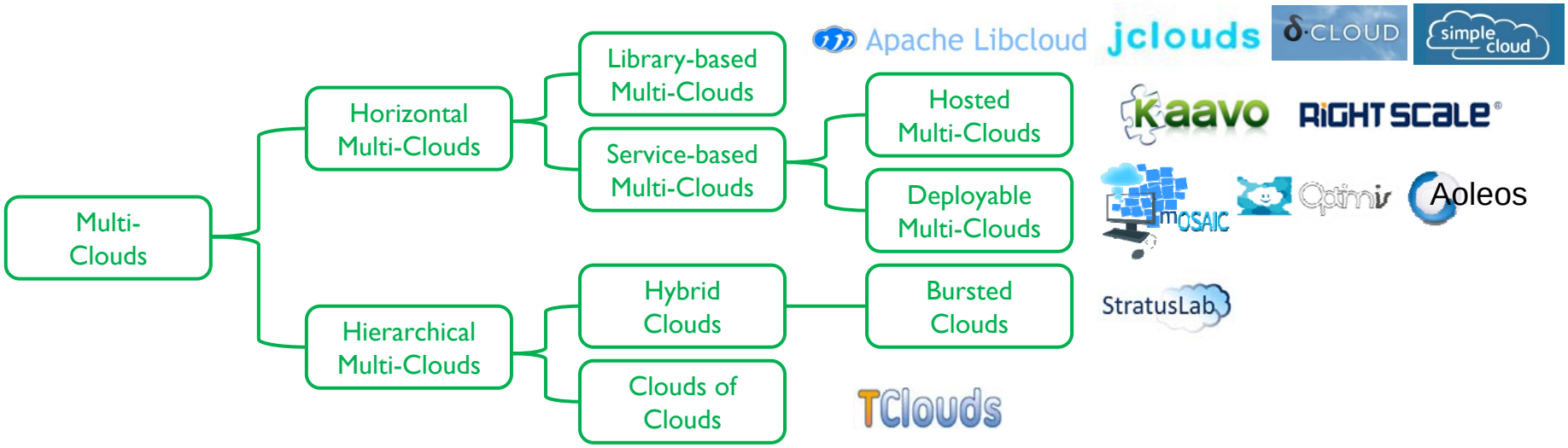
Delivery models for Multiple Clouds

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 third 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.

Classification



Examples of open libraries, services, tools



To solve in Multi-Clouds

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

On-going work/ team in Timisoara

- MODAClouds
  - FP7-ICT-8, 2012-2015
  - Model-driven engineering for Multi-Clouds
- SPECS
  - FP7-ICT-10, 2013-2016
  - Security SLA management
  - Part: Monitoring
- AMICAS
  - RO-PNII-PCE, 2012-2015
  - Automation in Clouds
- HOST
  - FP7-RegPot, 2012-2014
  - HPC services in Clouds

From Cloud humour site

Model-driven engineering in Clouds

MODAClouds

MODAClouds (www.modaclouds.eu)

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

MODAClouds objective

To provide

Methods +decision support system +

+ IDE + runtime environment

to support

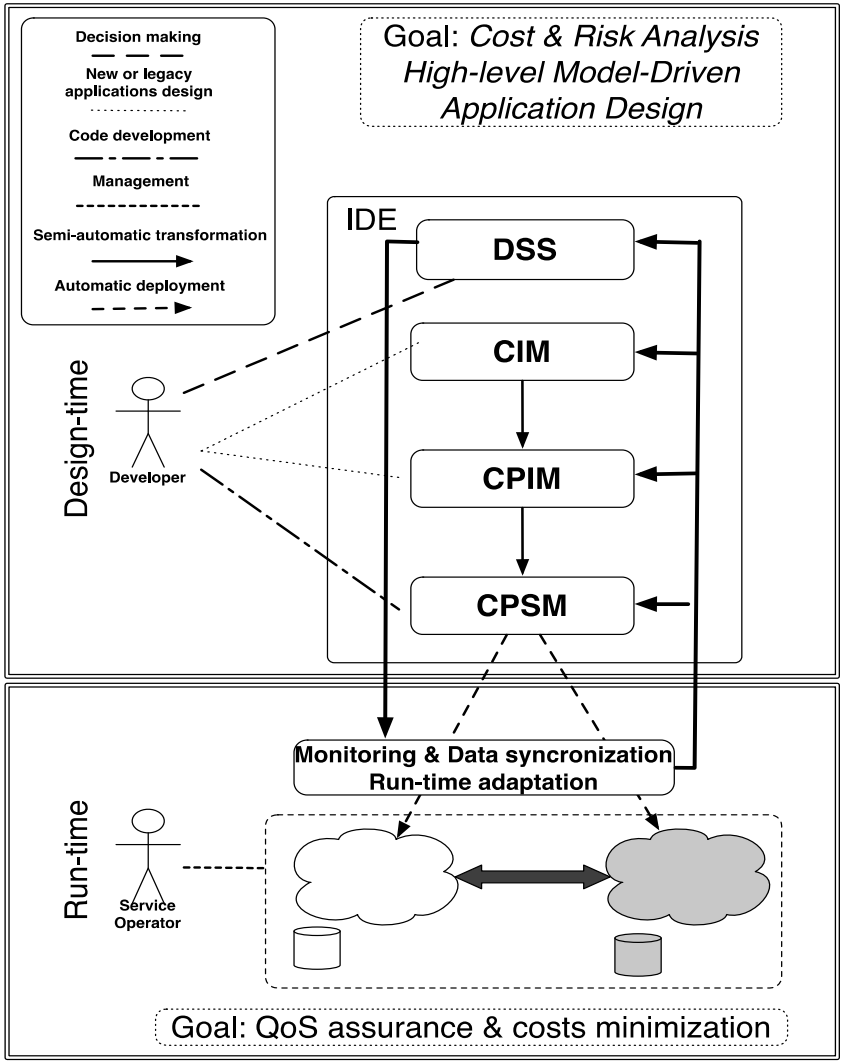
- High-level design
- Early prototyping
- Semi-automatic code generation
- Automatic (re)deployment
- Monitoring and self-adaptation

of applications on Multi-Clouds

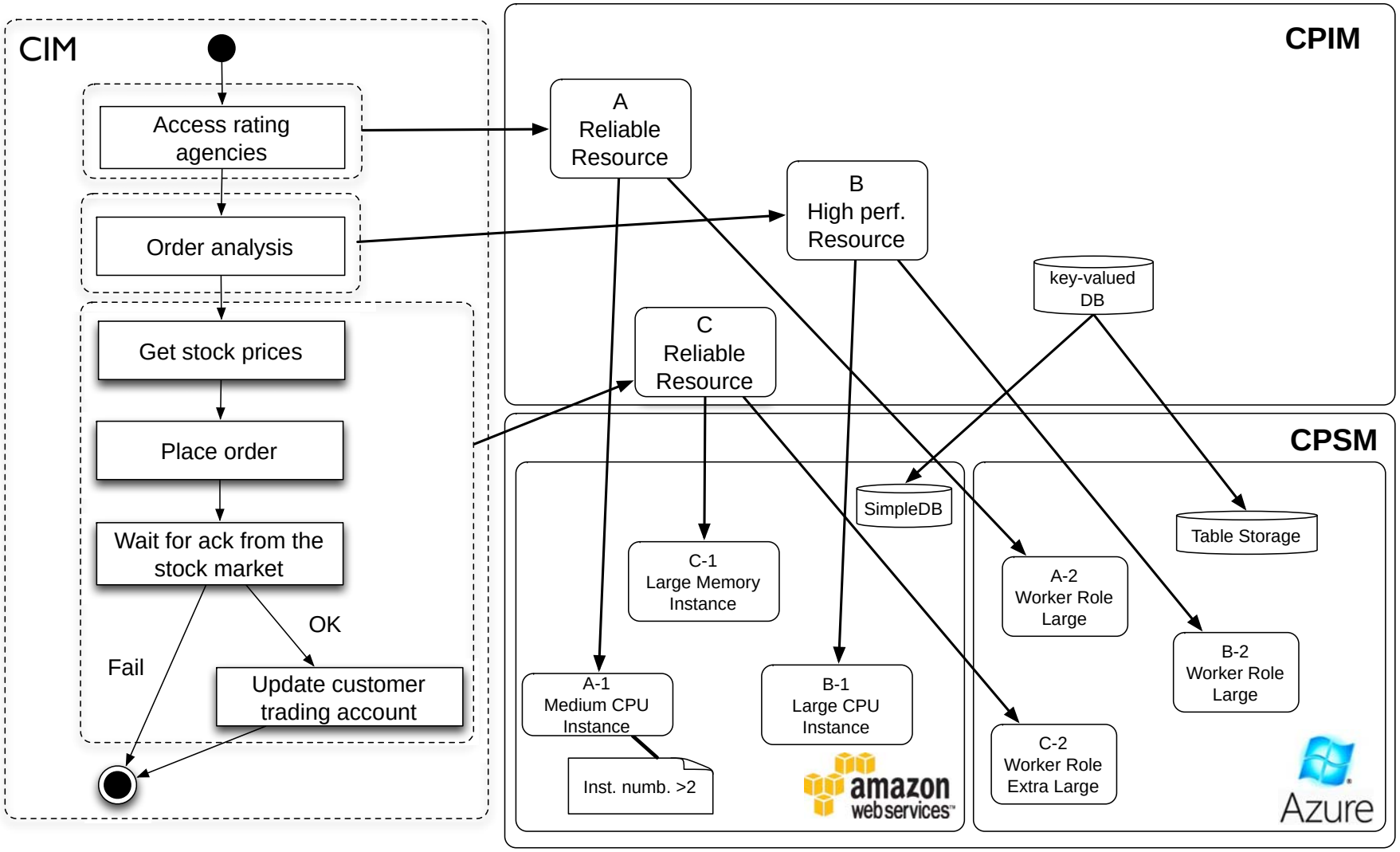
with guaranteed QoS

Multi-Cloud
DevOps Management

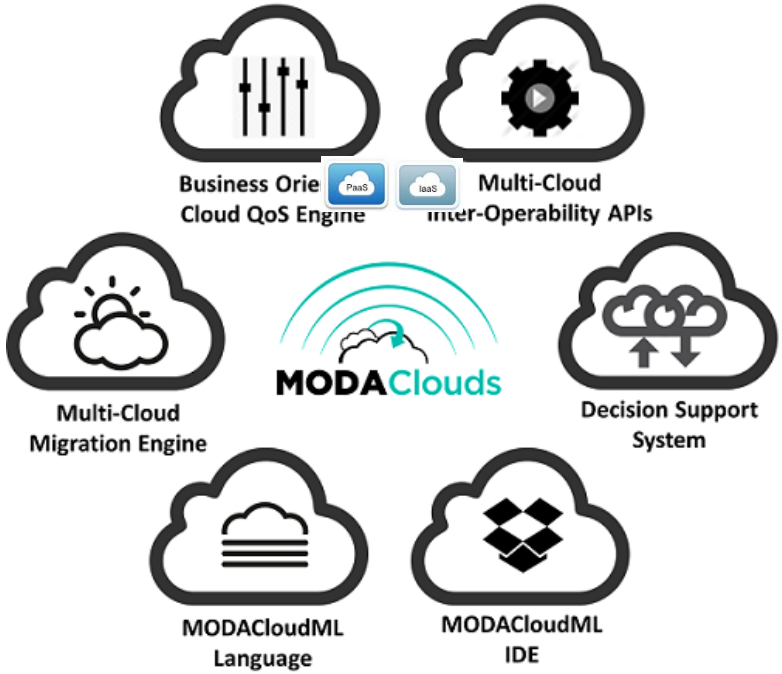
MODAClouds Vision



An example


CIM
Access rating agencies
Order analysis
Get stock prices
Place order
Wait for ack from the stock market
OK
Fail
Update customer trading account
CPIM
A Reliable Resource
B High perf. Resource
C Reliable Resource
key-valued DB
CPSM
SimpleDB
Table Storage
C-1 Large Memory Instance
A-2 Worker Role Large
B-2 Worker Role Large
A-1 Medium CPU Instance
B-1 Large CPU Instance
C-2 Worker Role Extra Large
Inst. numb. >2
amazon web services
Azure

Outputs



Cloud Development Tools

IDE +
MODACloudML (agnostic and QoS ready) modelling language

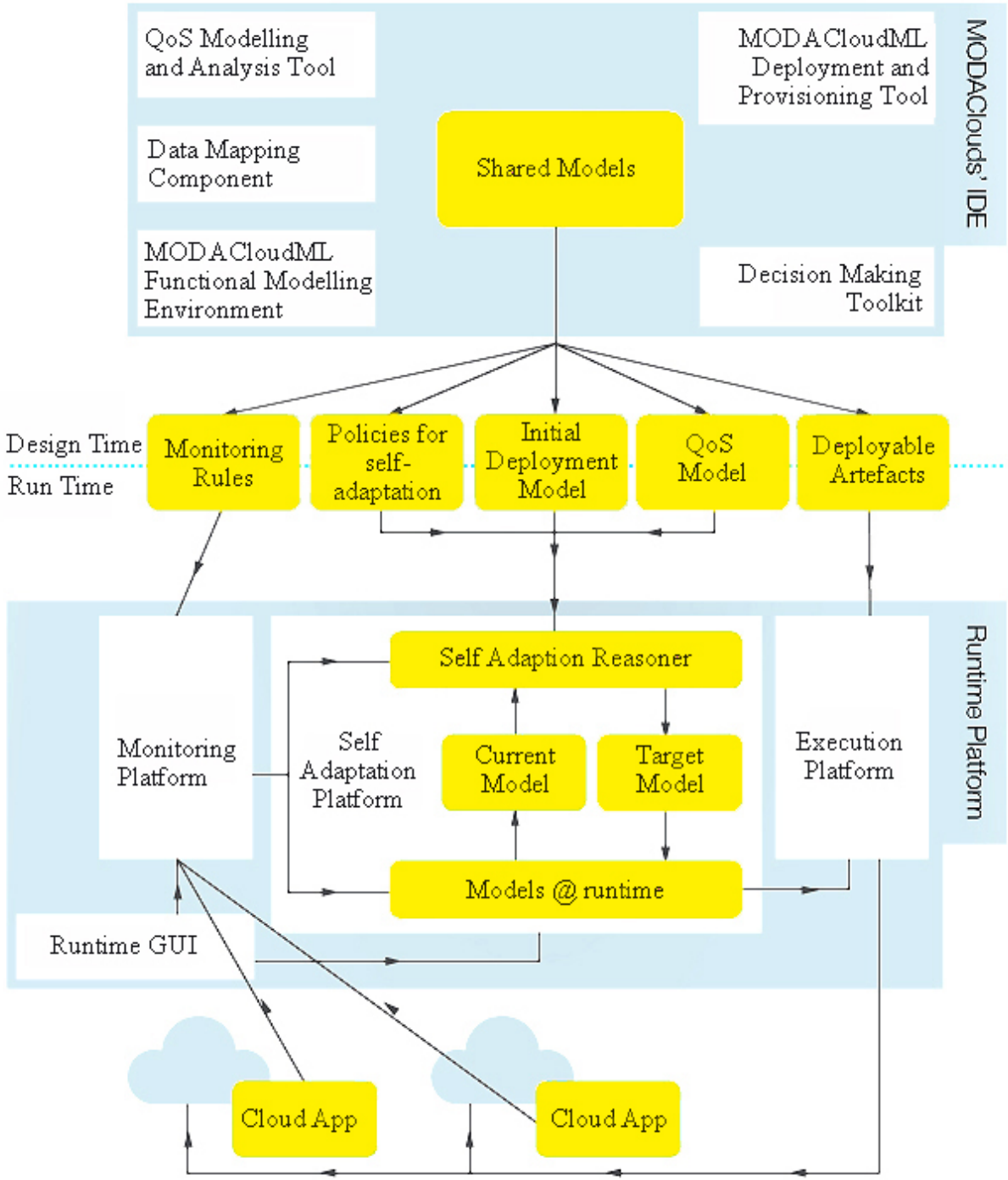
Flexible Multi-Cloud Apps Management, Monitoring & Operation Environment

maximizes automation with QoS Engine, Monitoring, Portability of underlying infrastructure providers (IaaS /PaaS)

Decision Support System

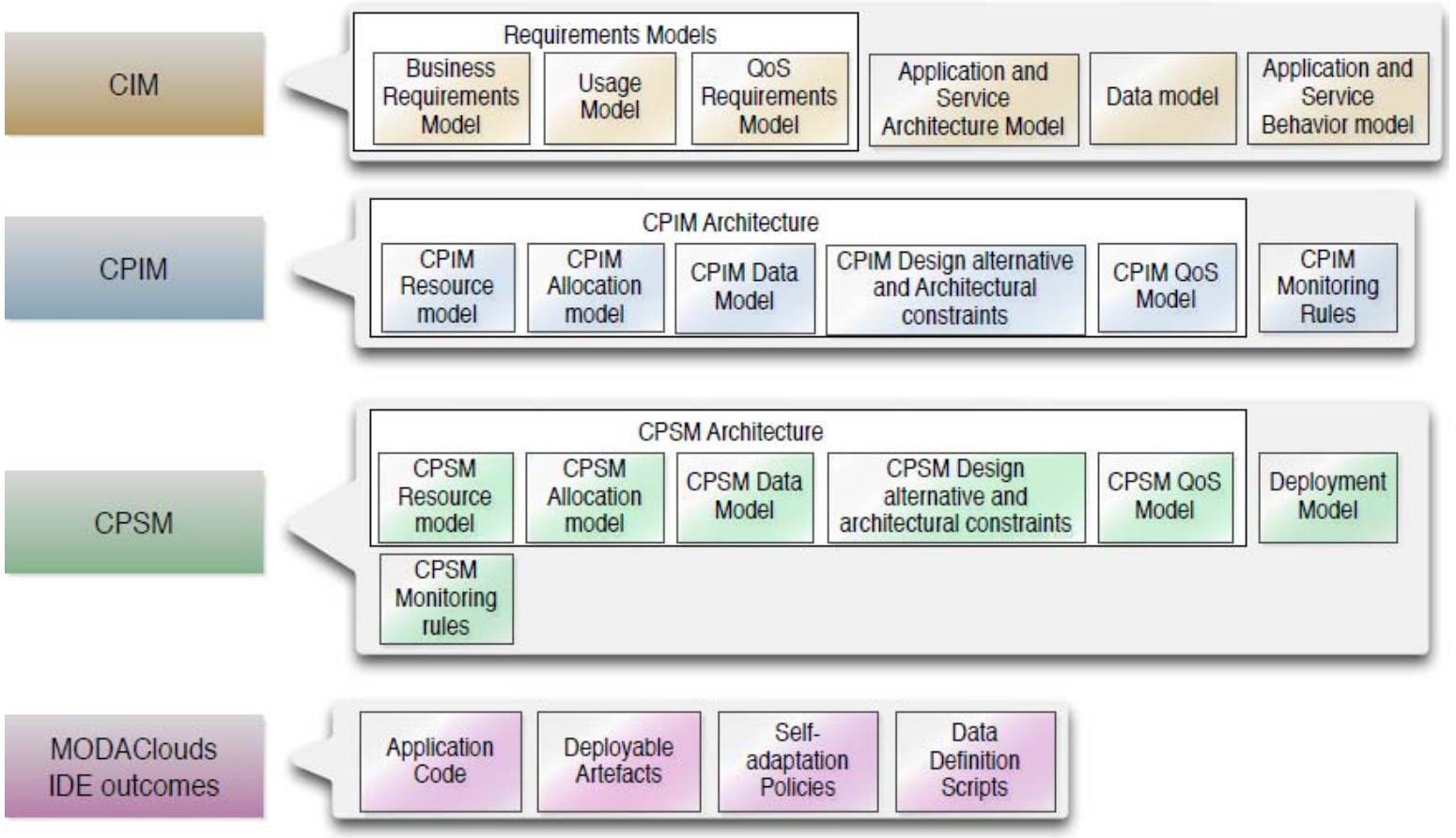
is a system on its own enables selection of provider at development & testing phase; and adds automation of runtime adaptation

Concepts & components maps

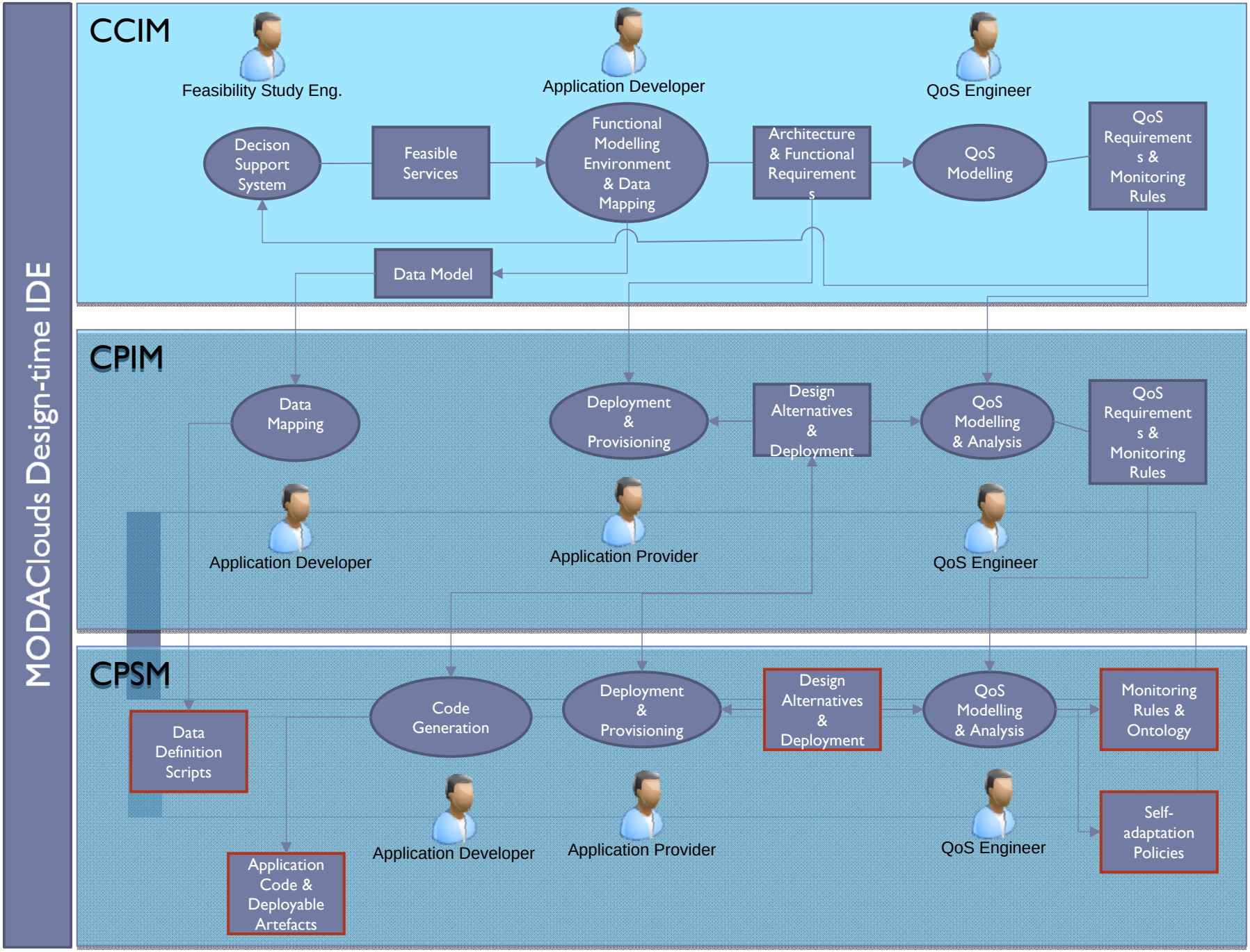


Details in the public deliverable D3.2.1

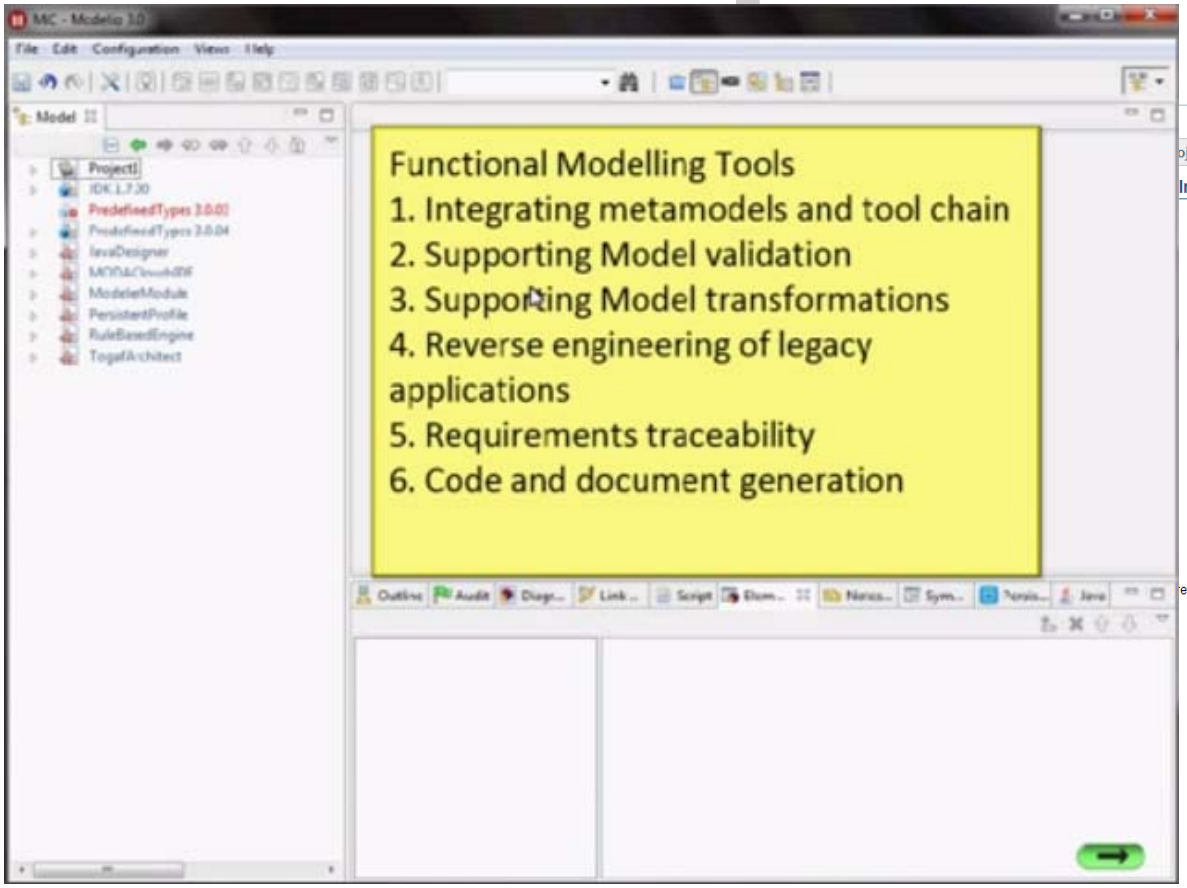
Shared models

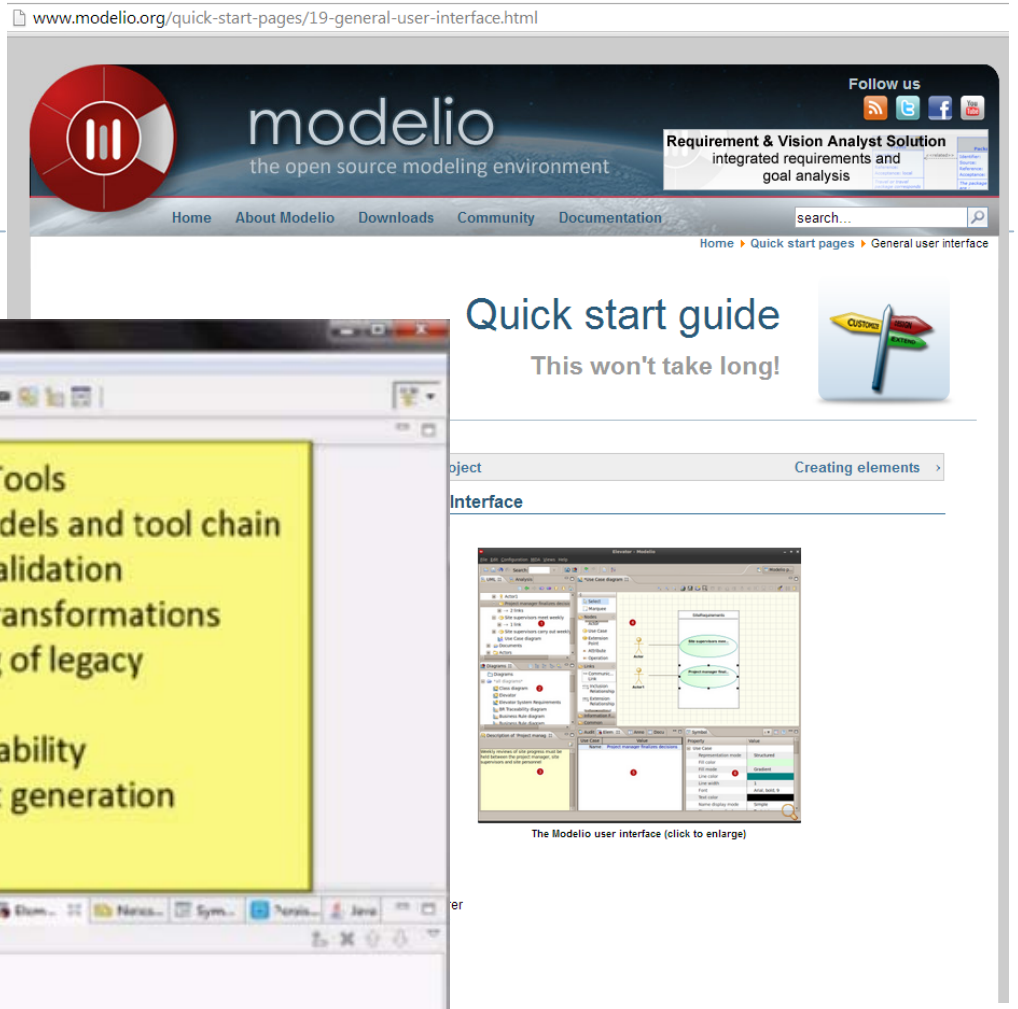


Details in the public deliverable D3.2.1

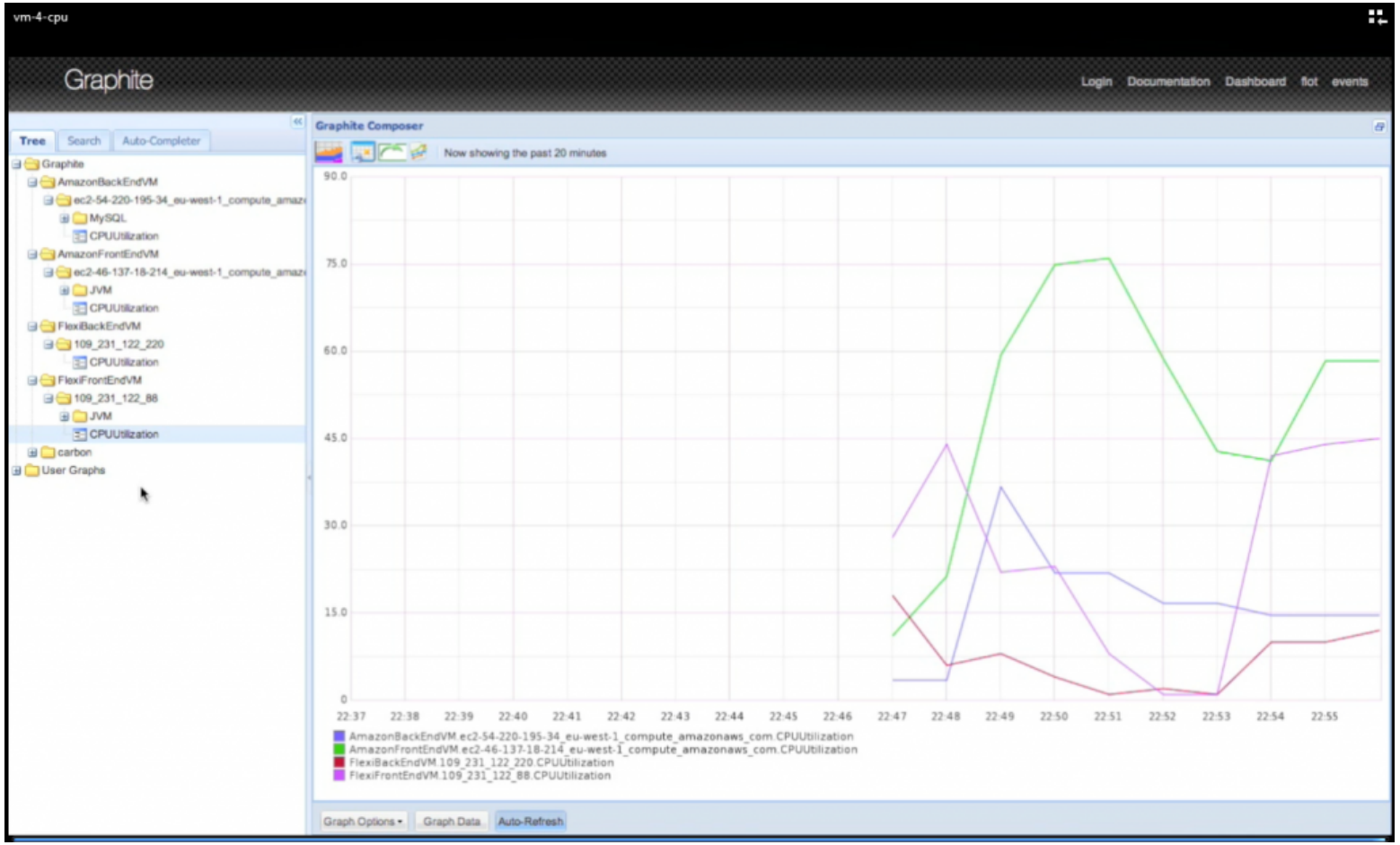

MODACIouds Design-time IDE
CCIM
Feasibility Study Eng.
Application Developer
QoS Engineer
Decison Support System
Feasible Services
Functional Modelling Environment & Data Mapping
Architecture & Functional Requirements
QoS Modelling
QoS Requirements & Monitoring Rules
Data Model
CPIM
Data Mapping
Deployment & Provisioning
Design Alternatives & Deployment
QoS Modelling & Analysis
QoS Requirements & Monitoring Rules
Application Developer
Application Provider
QoS Engineer
CPSM
Data Definition Scripts
Code Generation
Deployment & Provisioning
Design Alternatives & Deployment
QoS Modelling & Analysis
Monitoring Rules & Ontology
Application Code & Deployable Artefacts
Application Developer
Application Provider
QoS Engineer
Self-adaptation Policies

Modelio as IDE

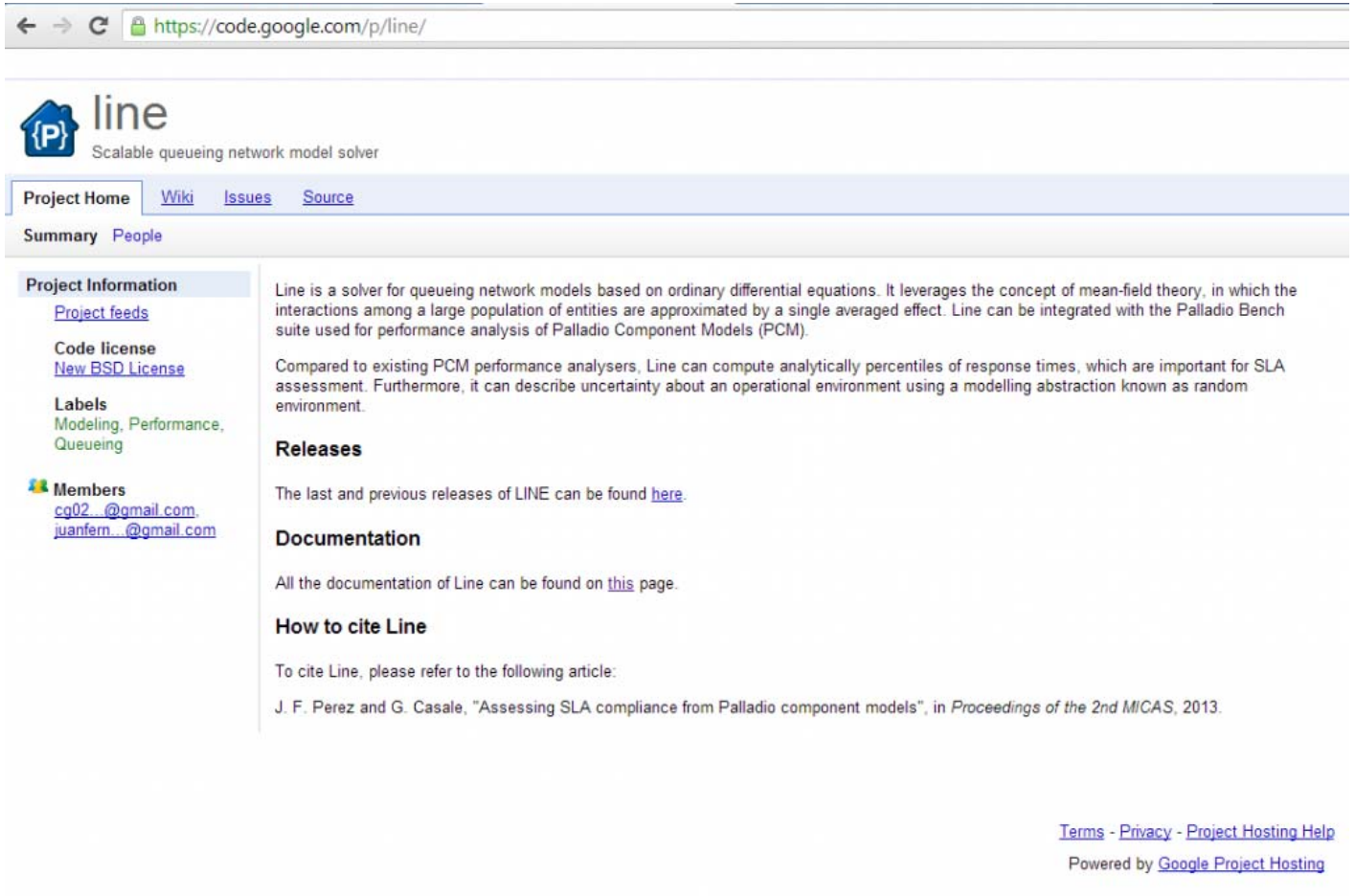




Monitoring & statistical data analyzers



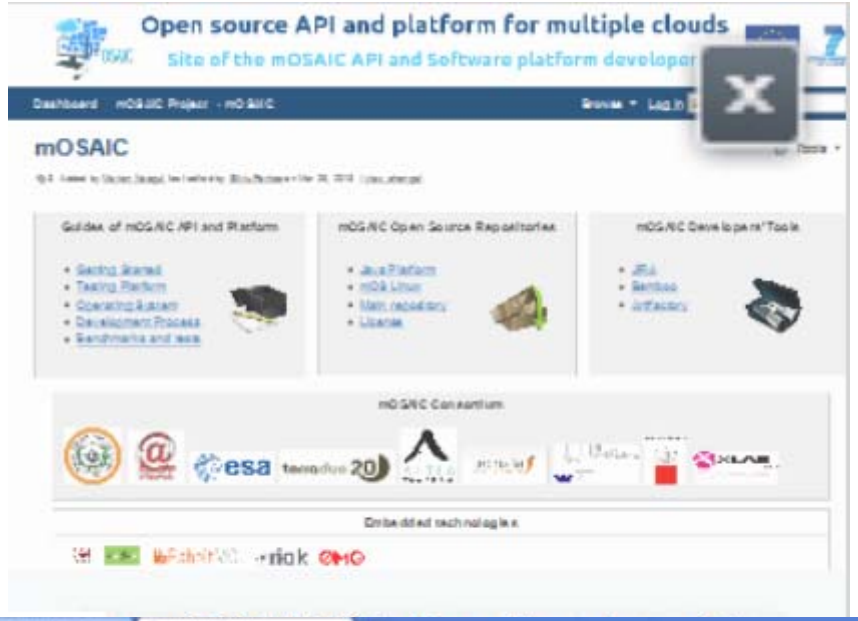
LINE:
Performance analyzer

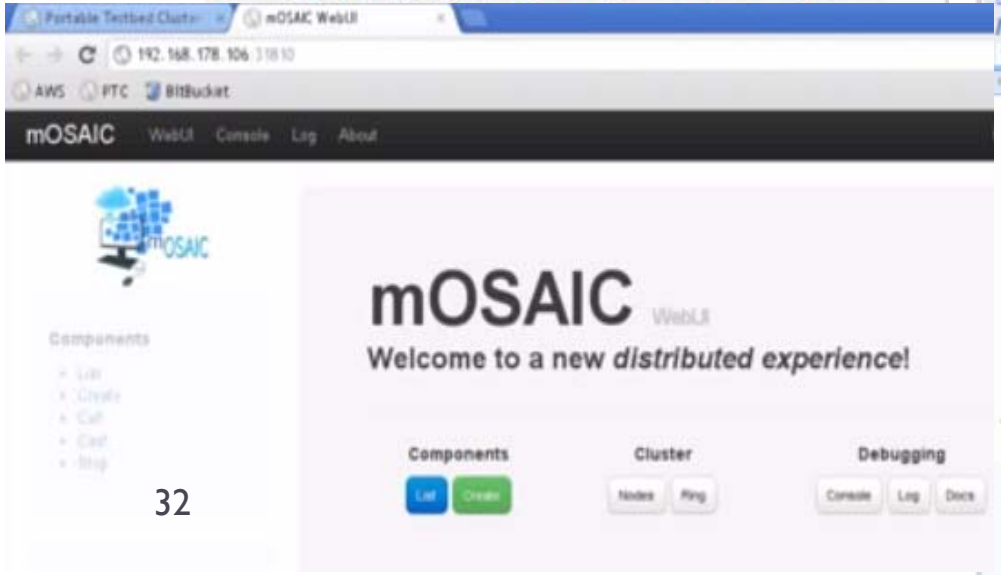


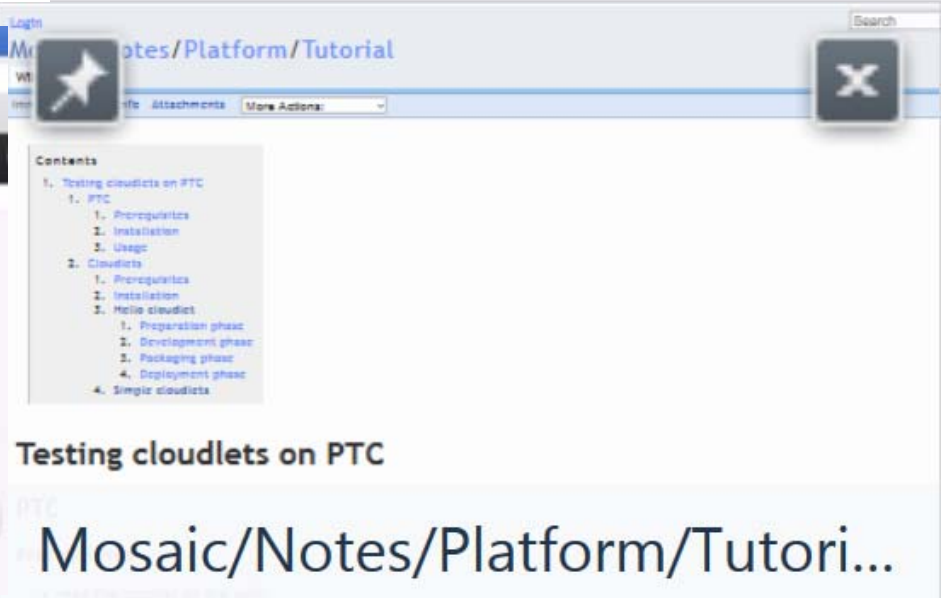
CloudML

domain-specific modelling language + run-time environment that facilitate the specification of provisioning & deployment

mOSAIC:
Run-time platform for Multi-Clouds







Where to find details

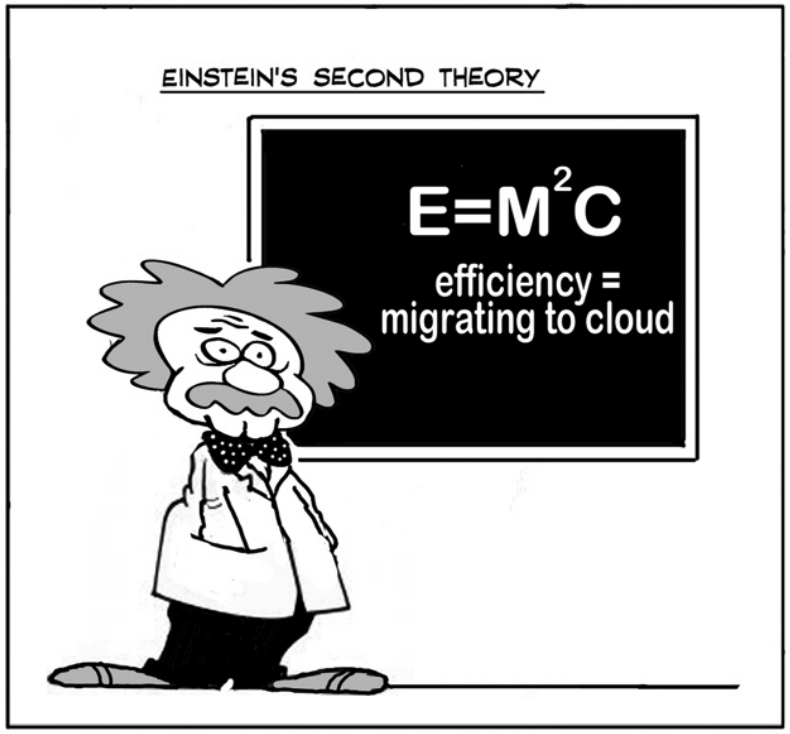
1. **Concepts:**
   - Public deliverables
   - Scientific Publications
   - White paper
2. **Software:**
   - Open-source components
   - Demos videos

... all on www.modaclouds.eu

Papers behind this presentation

- **Research agenda:**
  - Journal paper under evaluation
- **Multi-Clouds:**
  - D. Petcu, Consuming Resources and Services from Multiple Clouds. From Terminology to Cloudware Support, Journal of Grid Computing, January 2014, doi: 10.1007/s10723-013-9290-3
- **MODAClouds:**
  - E. Di Nitto et al, Supporting the Development and Operation of Multi-Cloud Applications: The MODAClouds Approach. 15th SYNASC, 23-26 Sept. 2013, 417-423, doi: 10.1109/SYNASC.2013.61
  - D. Ardagna et al, MODACLOUDS: A Model-Driven Approach for the Design and Execution of Applications on Multiple Clouds, Procs. MISE 2012, 50-56, doi: 10.1109/MISE.2012.6226014

Q & A ?



From Cloud humour site