

# Next Challenges of Cloud Computing

InfoH508 – ULB



**Skhiri Sabri, R&D director**  
**Brussels, 26/04/2011**

# Agenda

1. Cloud Ecosystem overview
2. SLA
3. Multi-tenancy
4. Governance
5. Standardization and interoperability

# Agenda

1. Cloud Ecosystem overview
2. SLA
3. Multi-tenancy
4. Governance
5. Standardization and interoperability

# Is it really cloudy?

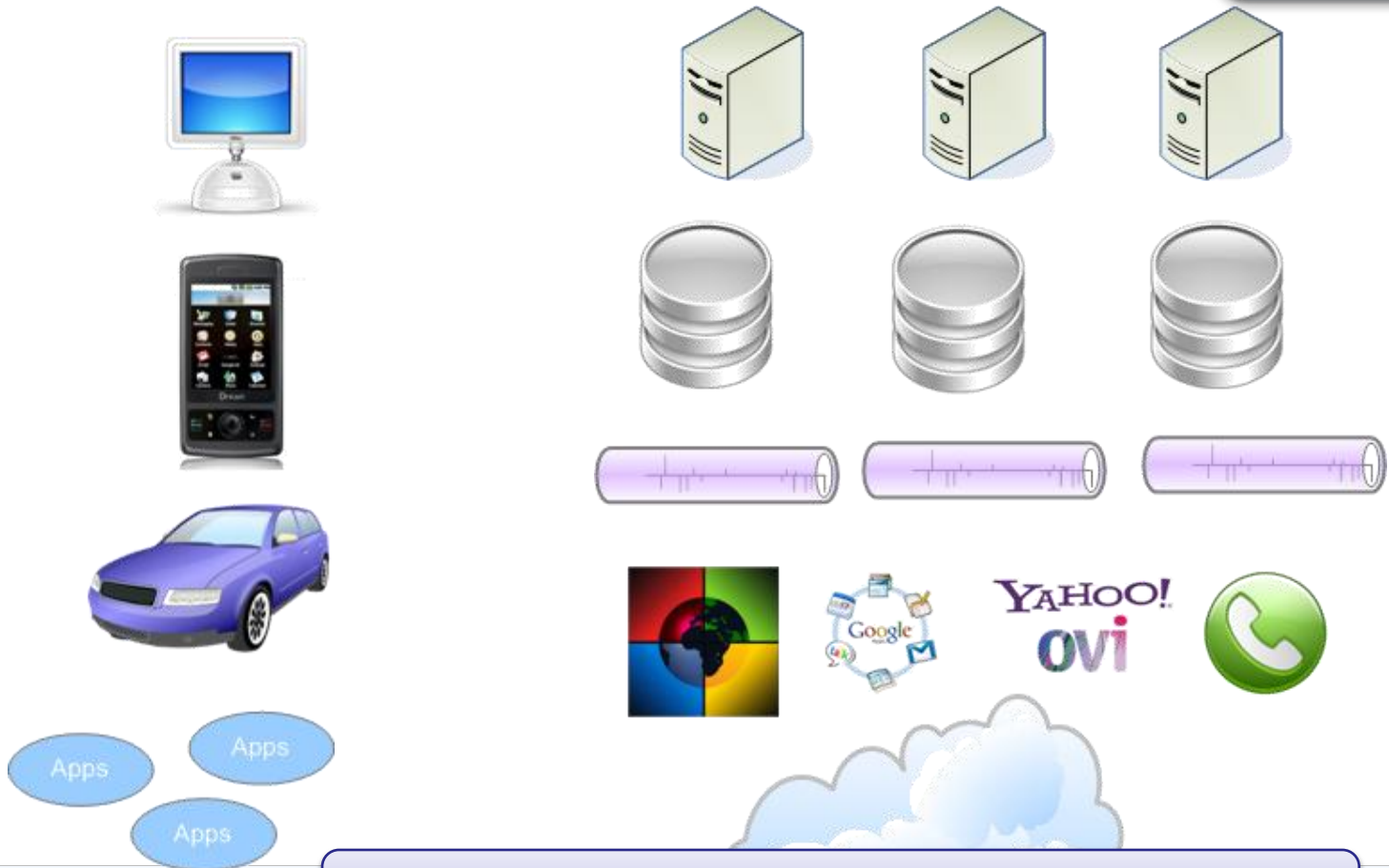
## Cloud definition



*Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*

# Is it really cloudy? (2)

Cloud definition



*Elasticity of resources*

# Delivery model

## User's perspective layers



use provider's application

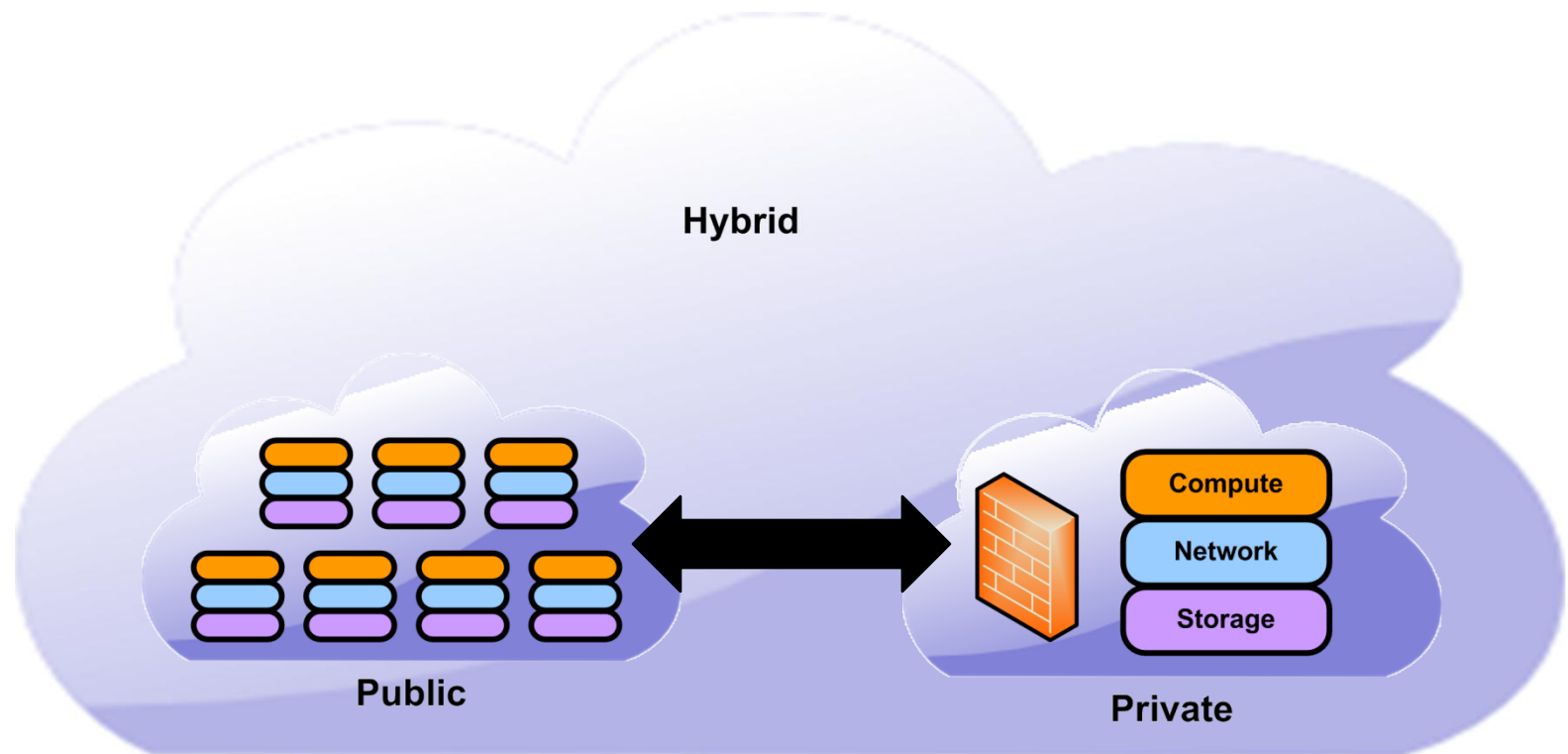


deploy our own application



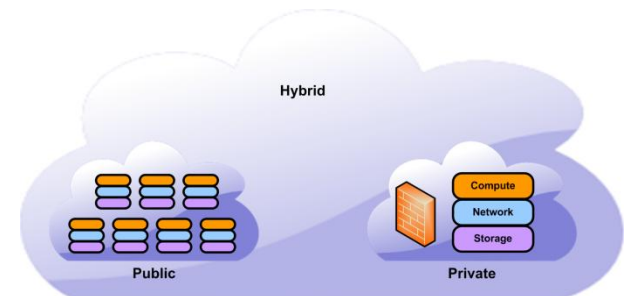
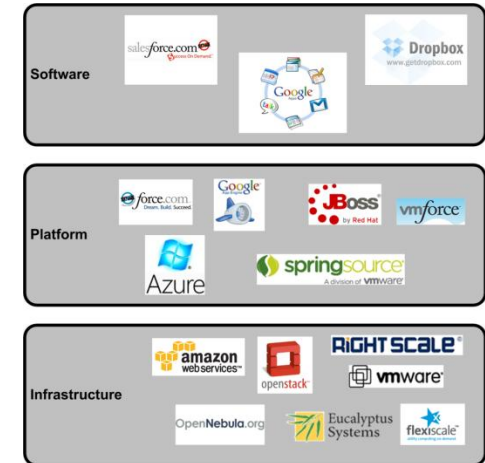
manage our own infrastructure

# Deployment models



# Key characteristics

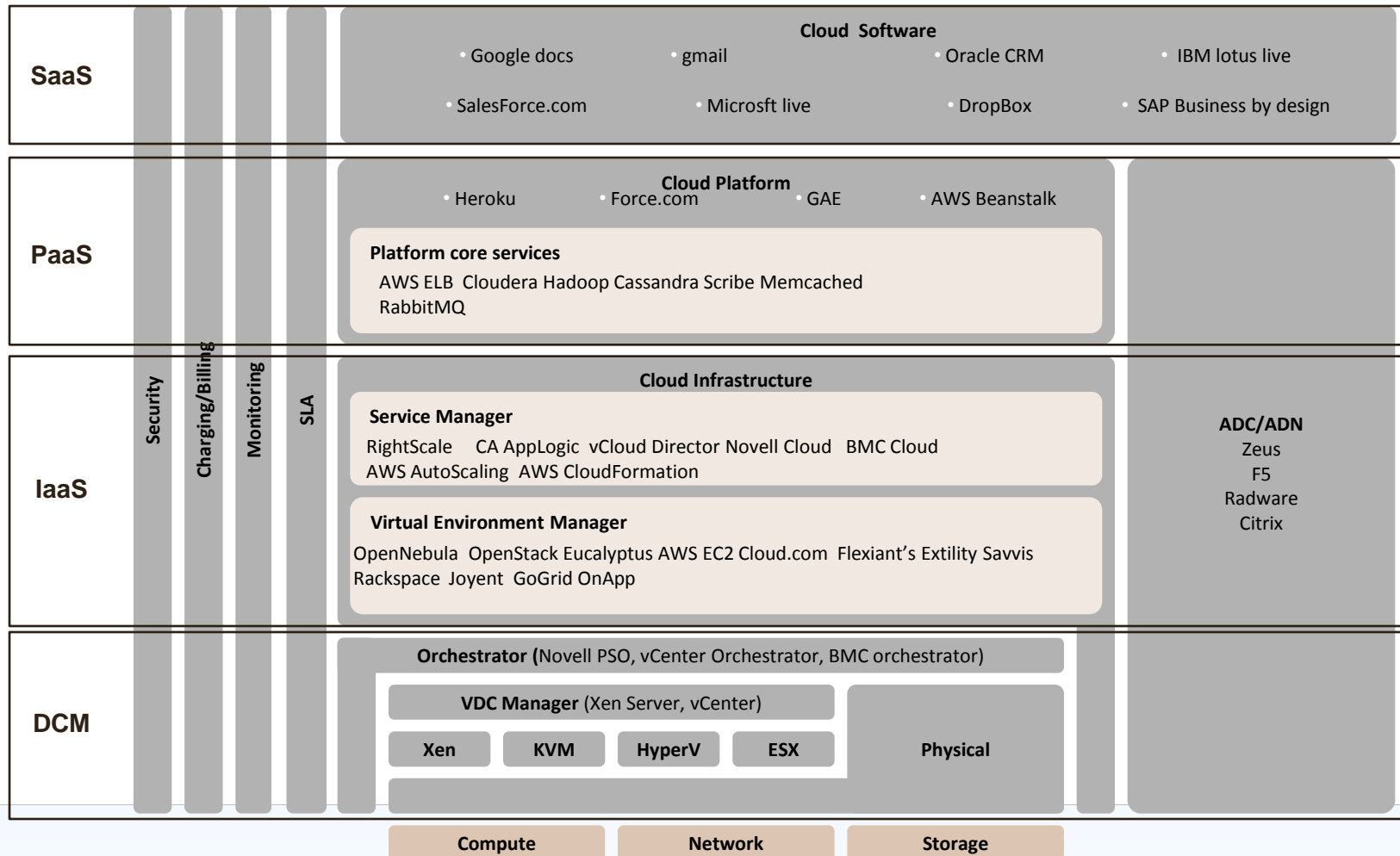
- Elasticity
  - Quickly provisionned and released
  - Unlimited resources access
- On-demand self-service
  - No human interaction on provider side
- Resource pooling
- Broadband network access
  - Always available from network
- Measured service
  - Monitoring transparency
  - Pay as you grow/use





# Cloud architecture layers

## Detailed view – major players



# The most important pieces

## Trends

- There is a clear convergence between DCM and IaaS
- Service management is emerging
- Platform is the current battle
- The global governance on all the stack is key element missing



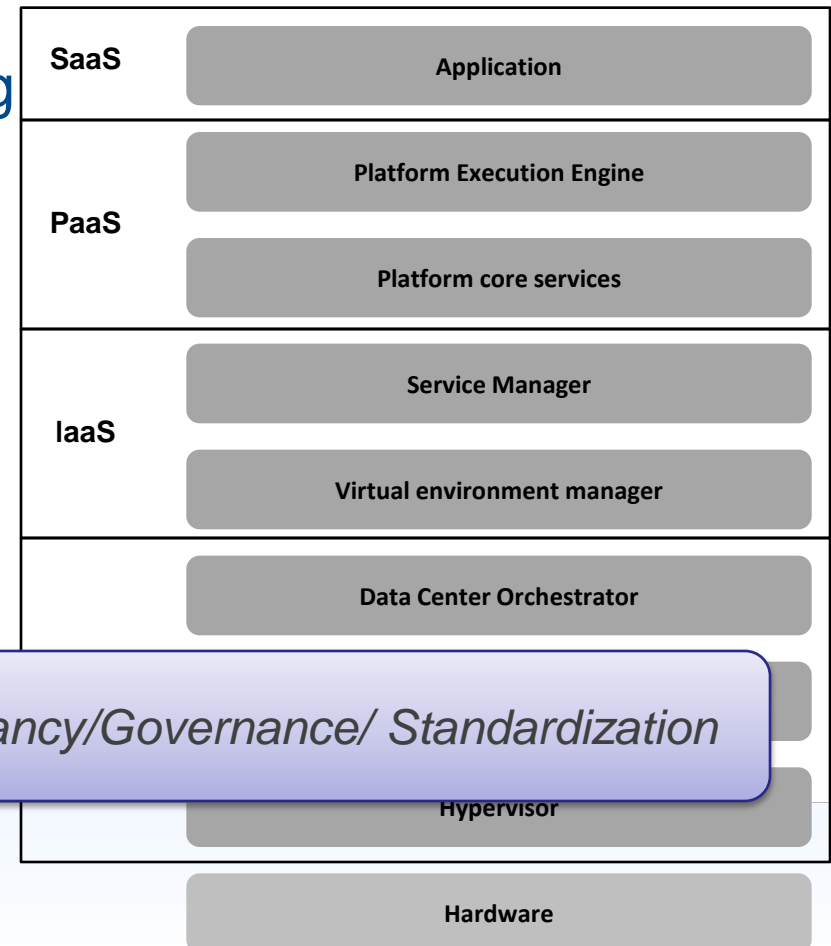
Admins



End users



Other roles



*This presentation focuses on SLA/Multi-tenancy/Governance/ Standardization*

# Agenda

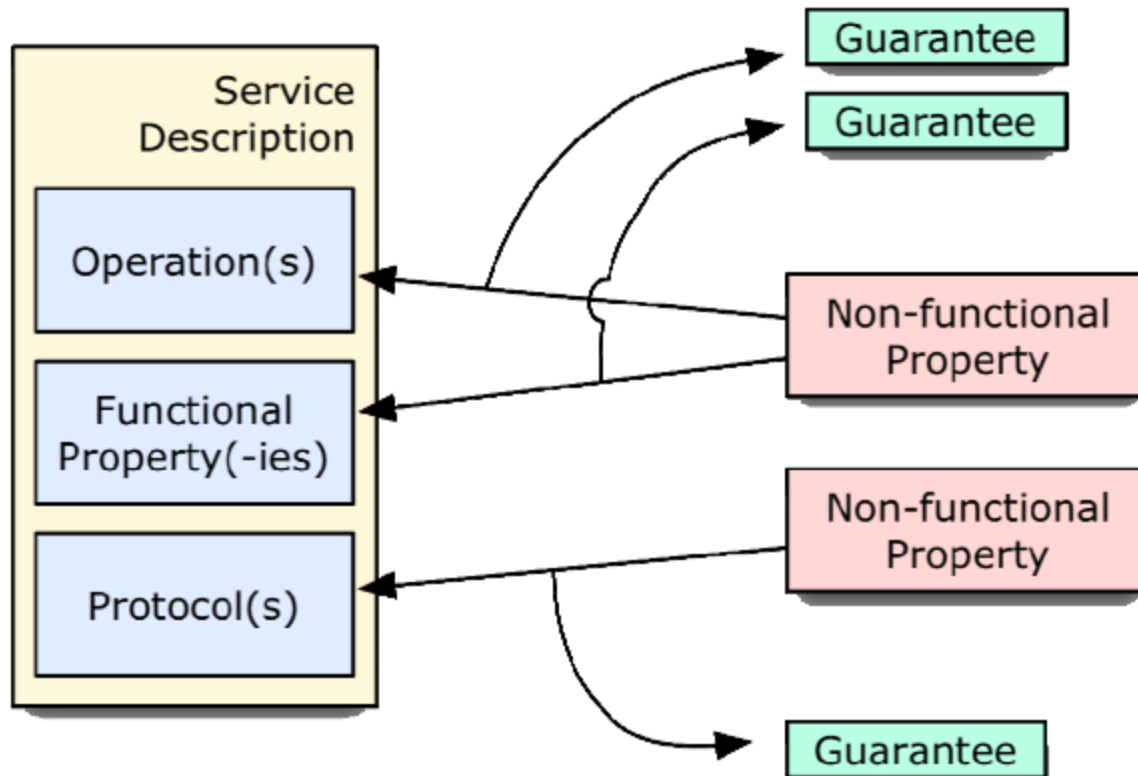
1. Cloud Ecosystem overview
2. SLA
3. Multi-tenancy
4. Governance
5. Standardization and interoperability

# What is a SLA?

Trying to define SLA

SLA is a generic term used at different levels of the IT

- 1.
  - 2.
  - 3.
- Many t
- 
- Difference
- S
  - a
  - F
  - C
  - k



Very  
t of things  
at

# Business level

What is the SLA?

- Under normal conditions
  - Service response time from the client viewpoint
    - X% of request must be treated in less than Y ms
  - Service availability
    - 99.xxx% availability Vs Gmail example
  - Service cost
- SLA may be defined for exceptional conditions:
  - Availability in case of natural disaster
  - Time to adapt to exceptional peak usage period
  - Using traffic shaping mechanisms to limit request rate to a defined acceptable rate

# Business level

What is the SLA?



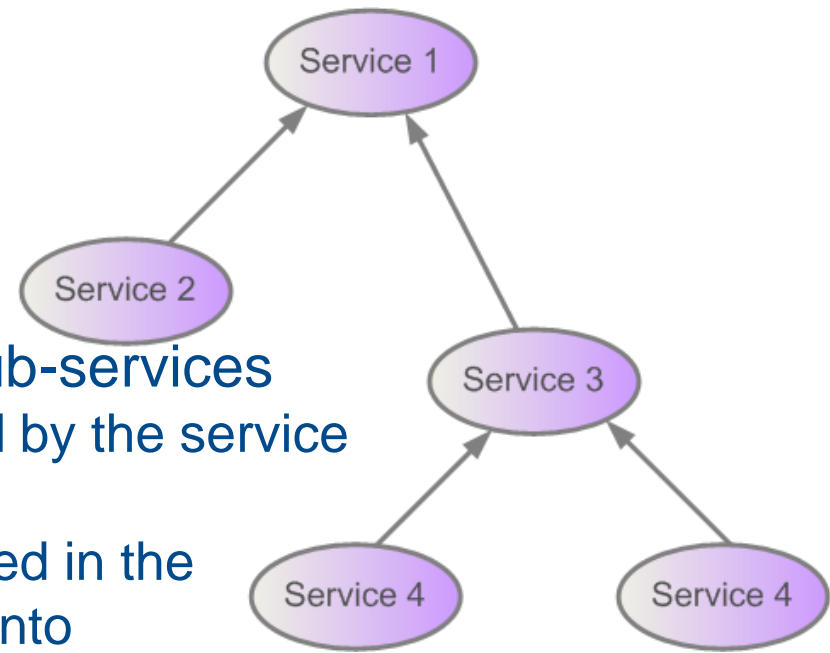
- Geographical constraints
  - Geographic zone where the service is allowed to execute
  - Geographic zone where the service is allowed to be stored and to store data (regulatory issues)
  - Geographic zone where the service should be accessible or not
- Outages
  - How to define an outage
  - How to prove an outage
  - How to get credit in case of outage
  - Time to Repair
    - Self-healing properties

# Service level

What is the SLA?

Each global service is composed of sub-services

- The global service level is impacted by the service level offered by each sub-service
- Each characteristic/constraint defined in the business SLA has to be translated into characteristics and SLO on each sub-service
  - Availability -> FT mechanism, DR, replication ...
  - Response time -> elasticity constraints, cache ...

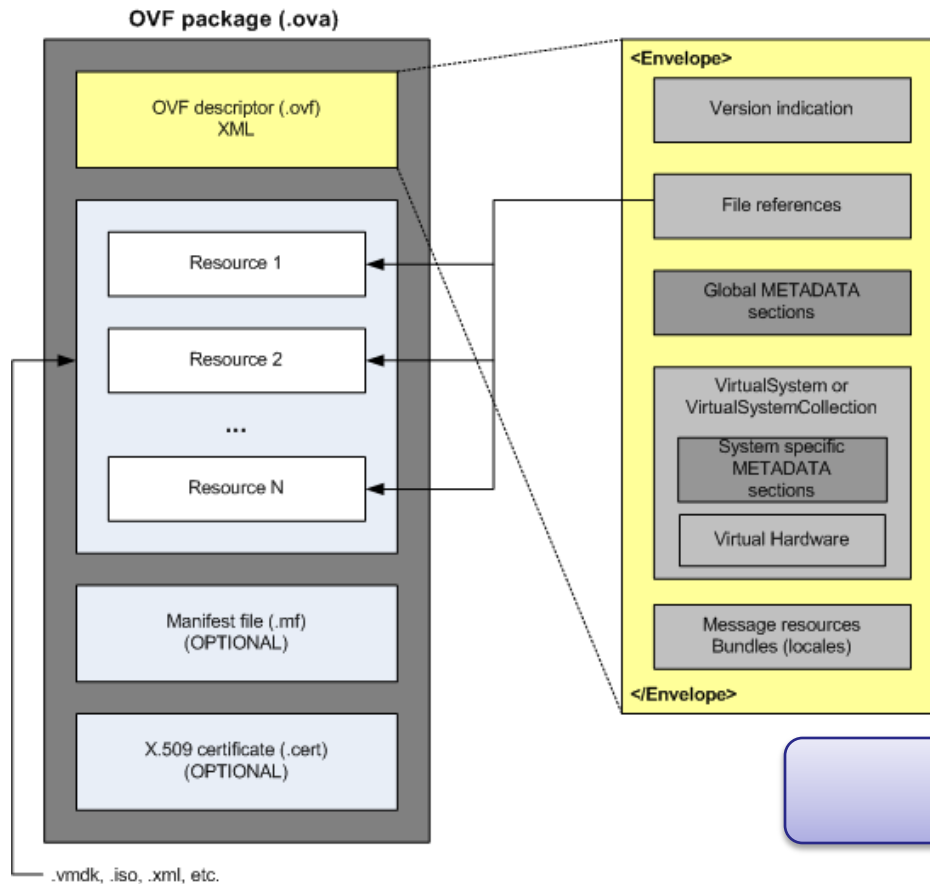


Different versions of a service may exist

- Different prices and different guarantees and performance levels => notion of **negotiation**

# Service management module

Service level SLA



OpenNebula.org

Automatic deployment



Auto scaling rules

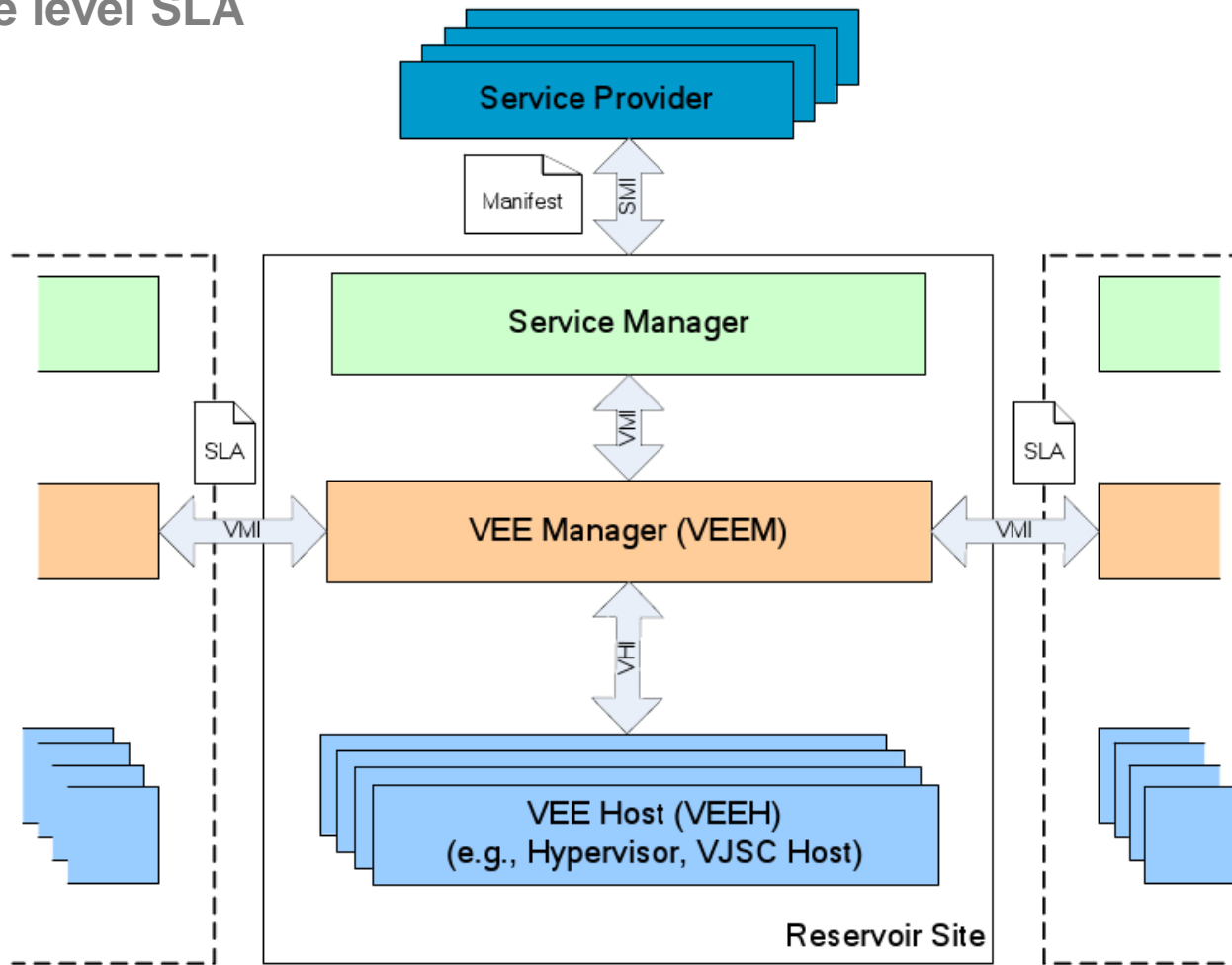
KPIs

<http://www.reservoir-fp7.eu/>



# Service management module

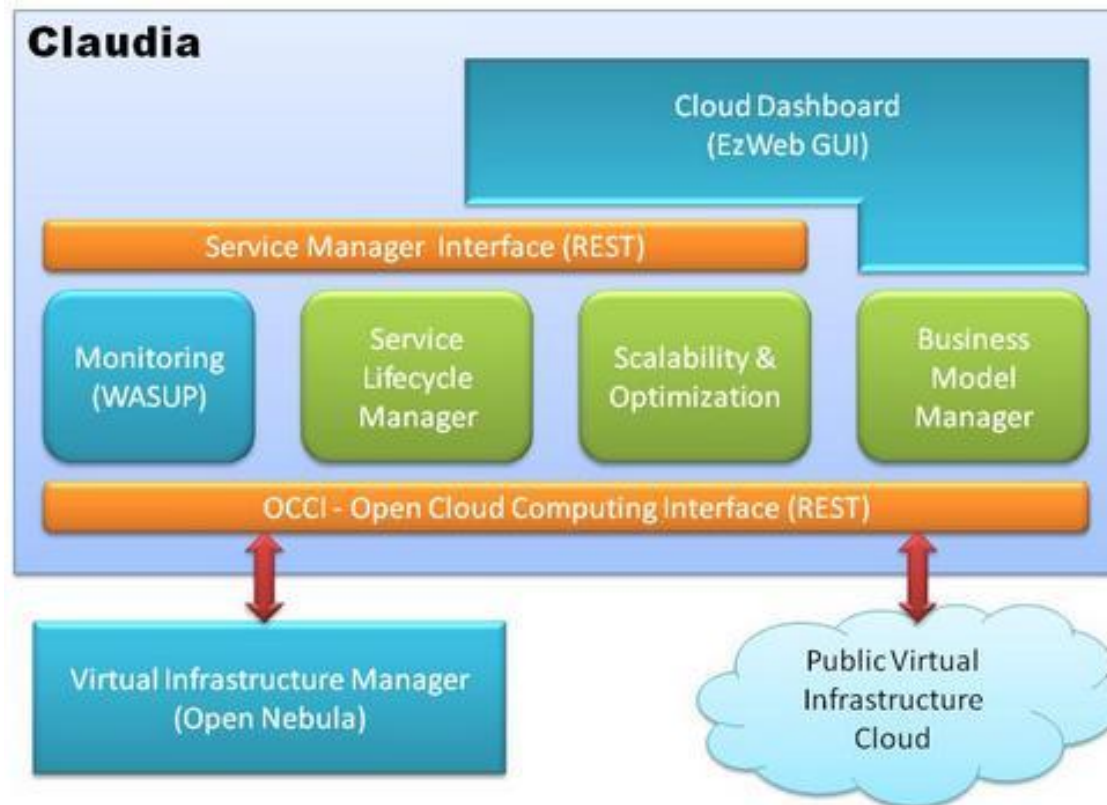
Service level SLA



# Service management module

Service level SLA

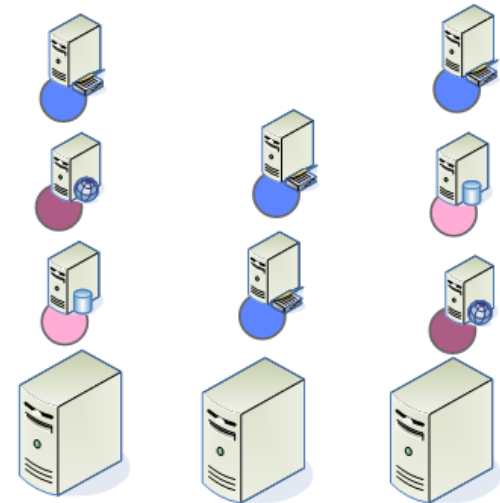
*Developed by Telefonica I+D*



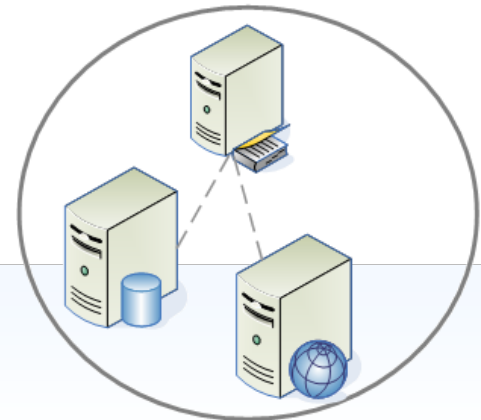
# What is a SLA?

## Infrastructure level

1. For each service, being able to define the characteristics of each VM
  - Cpu
  - Memory
  - vDisk
  - Network
2. Priority level of each VM
  - Scheduling

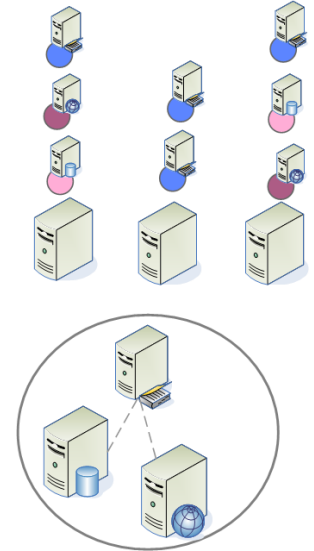
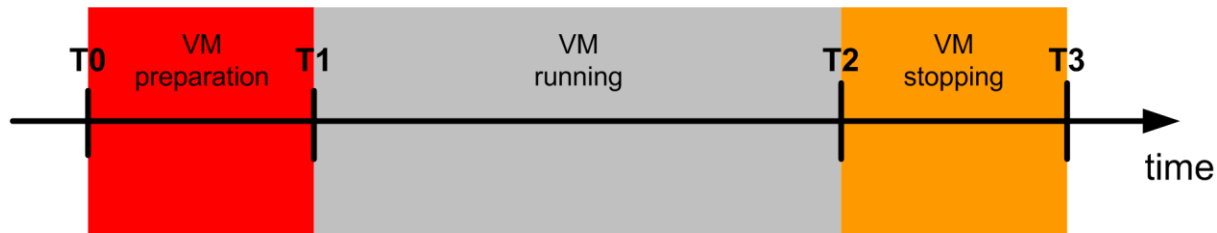


<http://haizea.cs.uchicago.edu>

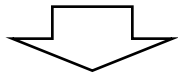


# VM Scheduling

## Problem statement



### Predictable workload



Advanced Reservation

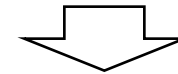
Predict VM stopping time

Predict VM preparing time

OpenNebula.org

HAIZEA

### Unpredictable workload



Reduce VM preparation & stopping

Give priority to VMs according to SLAs

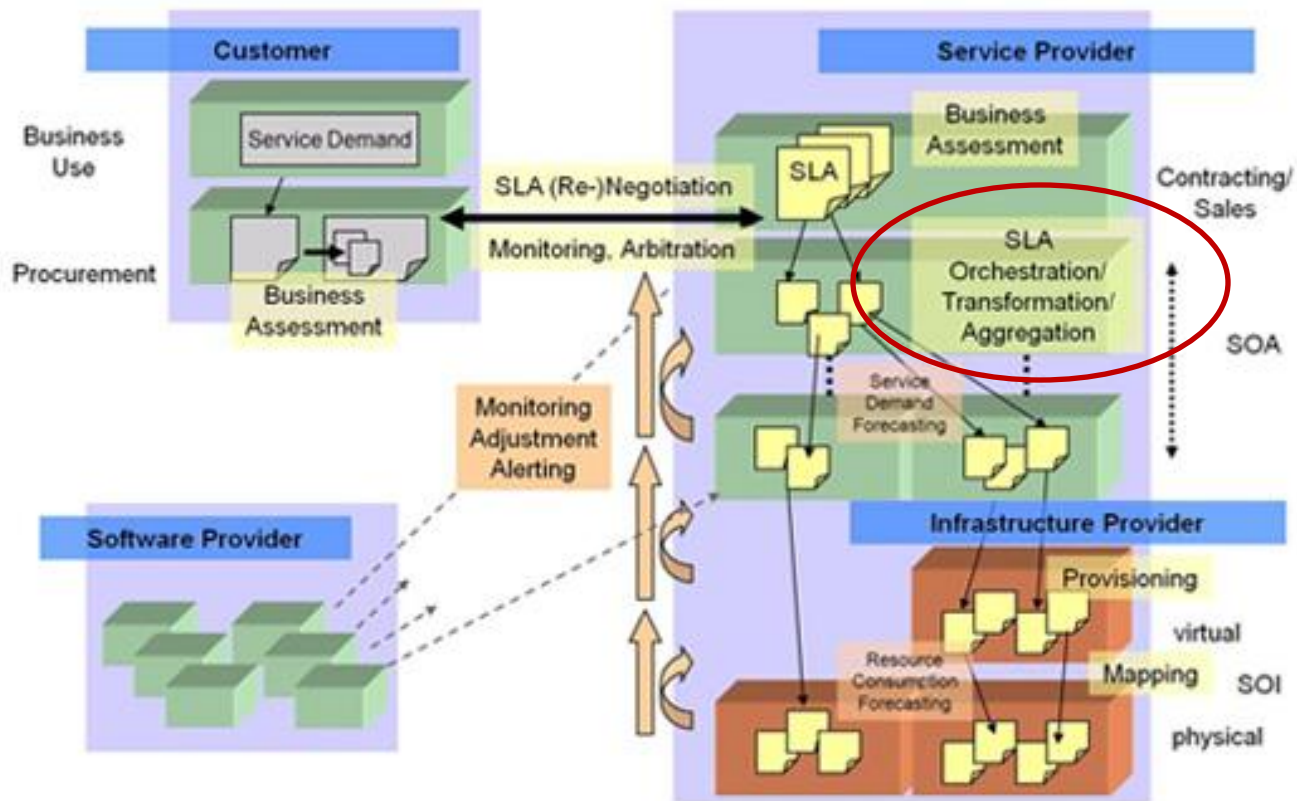
Pool of spare running VMs for critical Services

# SLA@SOI

EU FP7 project

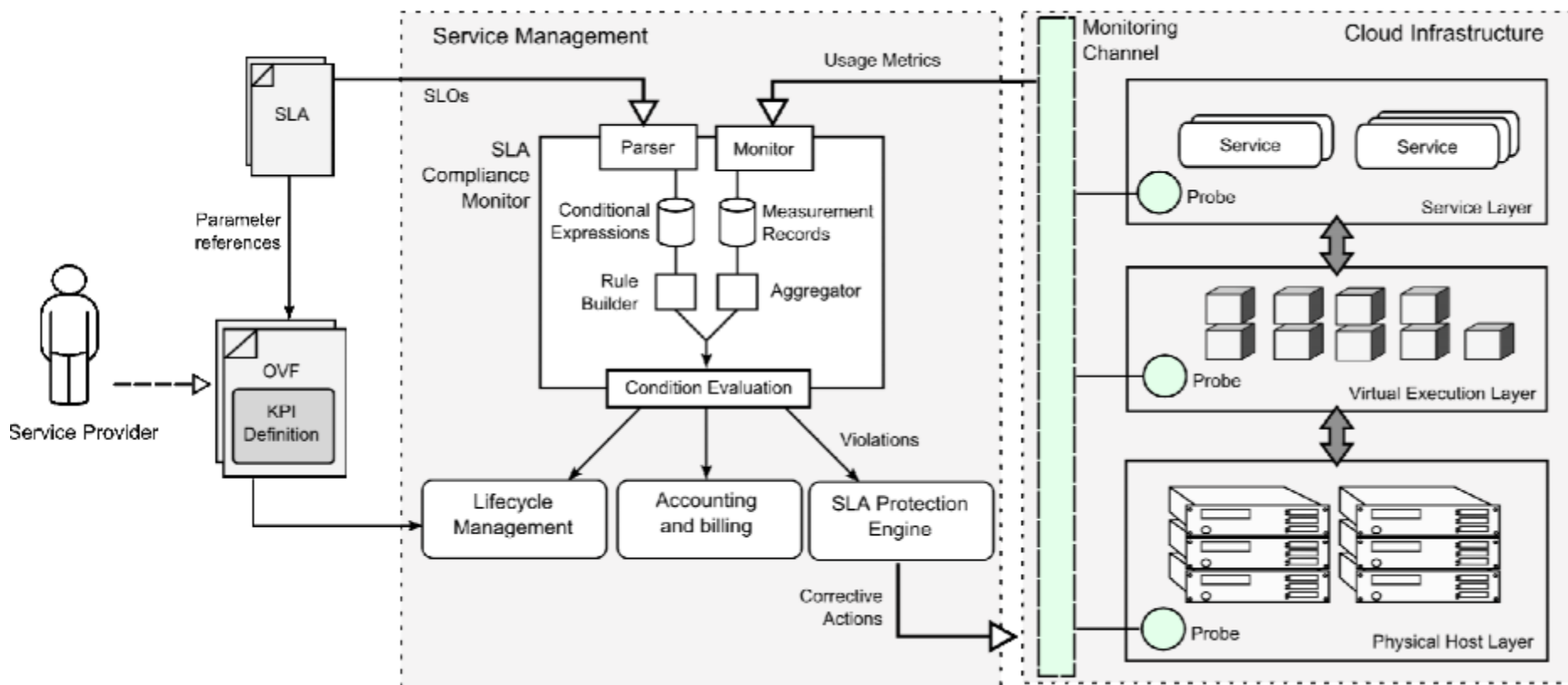
SLA is defined as a “formal contract that specify the characteristics, quality parameters and non-functional properties of a service – like price, performance and availability”

<http://sla-at-soi.eu/research/>



# High level architecture

## The SLA big picture



# Hot topics around SLA

## Challenges



- KPI management / Self-\* properties
- SLA -> SLOs transformation
- Cloud brokering for SLA management
  - Guaranteeing SLAs on different cloud providers
- Metering
  - Measuring usage accurately
- Auditing
  - Infrastructure validation – SOX compliancy
- Regulatory issue for data location

# Agenda

1. Cloud Ecosystem overview
2. SLA
- 3. Multi-tenancy**
4. Governance
5. Standardization and interoperability



# What is the multi-tenancy?

Offering shared resources

- Offering shared resources to multiple users
  - While guaranteeing isolation & SLA
- Which use cases are you thinking about ?



# Challenges

## Multi-tenants architectures

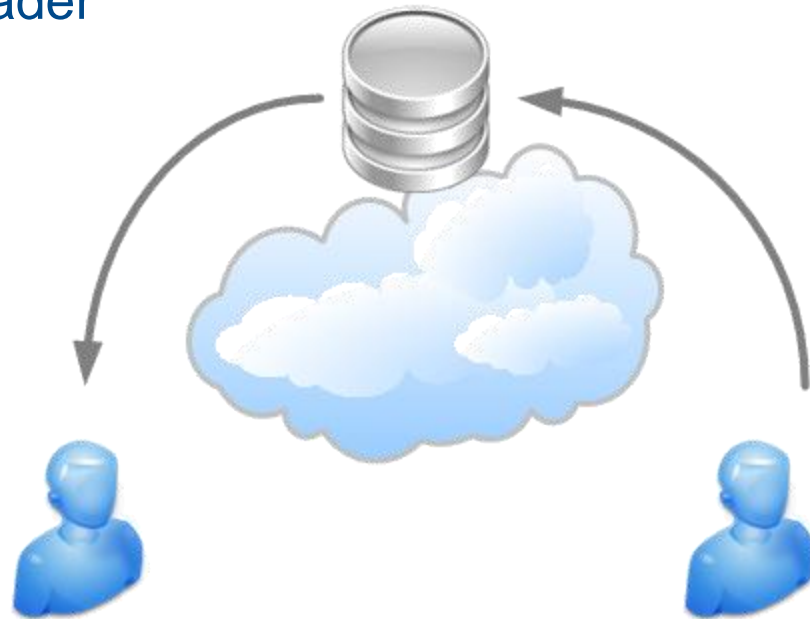
- Sharing storage
  - Access & performance/space/isolation/Security
- Sharing CPU and memory
  - Guarantees on isolation
- Sharing bandwidths
  - Guarantee on access & performance
- Charging to each tenant according to SLA



# Multi-Actors billing

Tenants can have different roles

- The App Store example
  - The app provider does not pay neither for the storage nor for the uploading bandwidth
  - Paid by the app downloader



# Agenda

1. Cloud Ecosystem overview
2. SLA
3. Multi-tenancy
- 4. Governance**
5. Standardization and interoperability

# What is the governance?

## Working definition

*Global governance must help the developers and designers during each phase and must ensure the global coherence of the infrastructure by validating services with its policy rule against the system. Moreover, it is as well all other mechanisms about getting People (developer, architect, and designer) to do the right thing at the right time. In other words, it is about encouraging the behaviour that will achieve Enterprise business goals.*

- [1] Oracle, *SOA Governance: Framework and Best Practices*, Oracle Whitepapers 2007
- [2] L. F. Kenney, D.I C. Plummer, *Magic Quadrant for Integrated SOA Governance Technology Sets*. Gartner Research, 2009.
- [3] M Afshar and al., *SOA Governance: Framework and Best Practices*. Oracle Whitepaper, 2007.
- [4] P. Malinverno, *Simple Governance Mechanisms for a Service Oriented Architecture*. Gartner Group Research Note G00139465, 27/ April 2006.
- [5] W.A. Brown and al., *SOA Governance: Achieving and Sustaining Business and IT Agility*. IBM press 2008.

# Governance framework

## Implementing the governance

Governing the complete life-cycle of infrastructure components and services

- Designing service
- Building and deploying service
- Linking Service
- Operating service
- Modifying, make evolve service
- Retiring service

**Tooling  
layer**

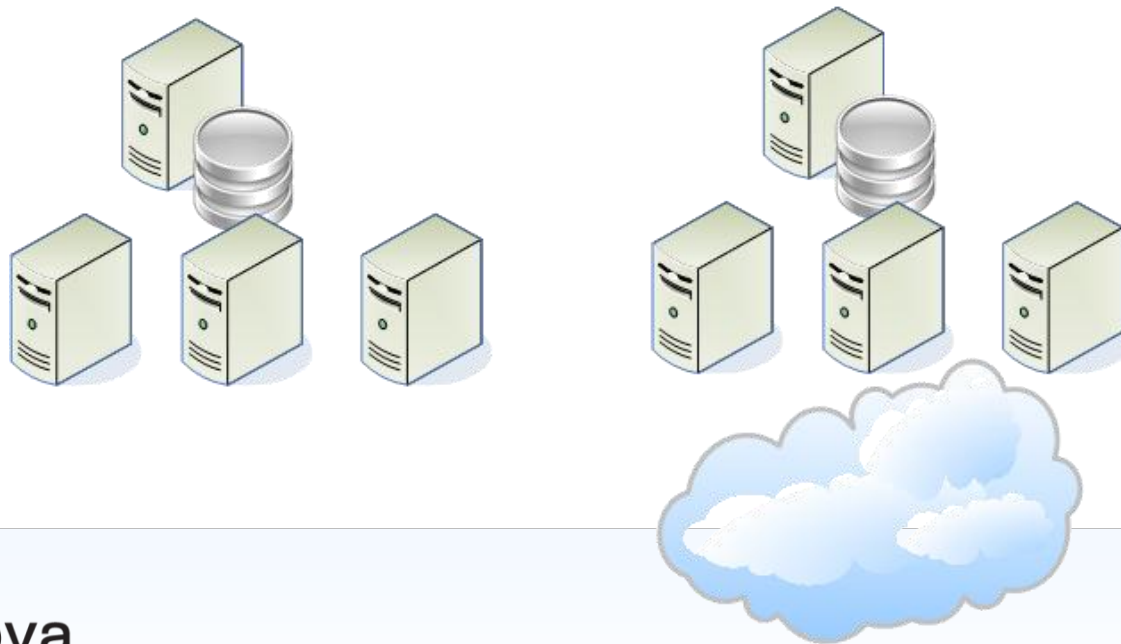


**Runtime  
layer**



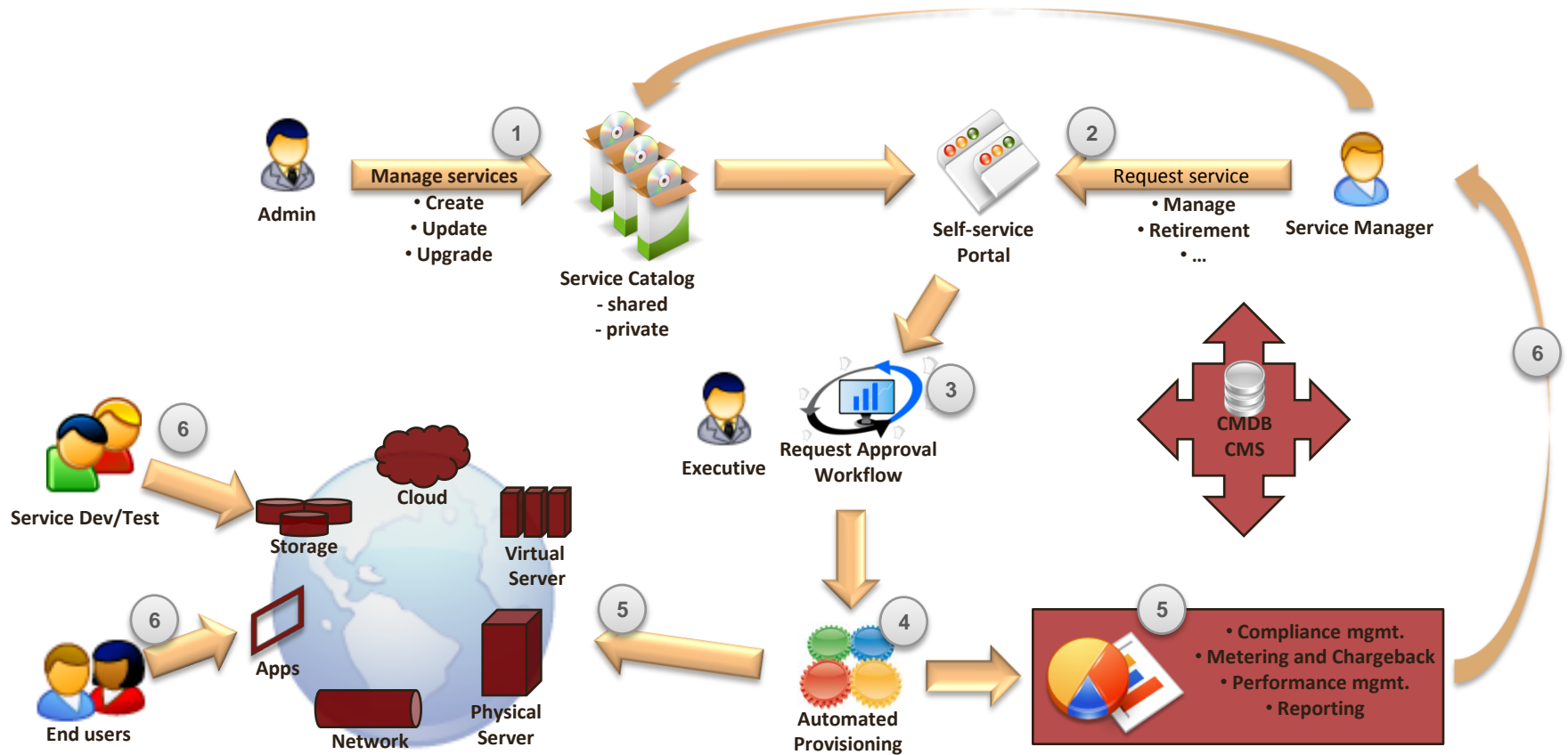
# Why do we need governance in the cloud?

- What is the difference between an IT infrastructure and a *cloudified* IT infrastructure?
- Two viewpoints to consider
  - The cloud customer who deploy services on its *cloudified* IT
  - The cloud provider who must manage the entire infrastructure



# Governance in the cloud

## A cloud service life-cycle





# Governance in the cloud infra

## A cloud service life-cycle

- Service catalogs
  - CMDB – Dependencies management – UDDI servers
- Deploying on  $x10^3$  machines requires other kind of tools for
  - Configuring VMs
  - Updating the template on live VMs
- Cloud configuration stack
  - Elastic group
  - Elastic IP management
  - KPI management – alerting system – corrective actions
  - Auditing – monitoring
  - Charging and billing configuration

# Governance in the cloud infra

## A cloud service life-cycle

- Workflow management
  - Approval – orchestration and complex provisioning
- Multi-site load balancers
  - CDN and service-aware cache optimizations

# Agenda

1. Cloud Ecosystem overview
2. SLA
3. Multi-tenancy
4. Governance
- 5. Standardization and interoperability**

# Why do we need standards?

## Example of Hypothetical Customer Scenarios: Microsoft Use Cases

1. Move three-tiered application from on-premises to cloud
2. Move three-tiered cloud application to another cloud
3. Move part of an on-premise application to cloud to create a “hybrid” application
4. Hybrid application with shared user ID and access services
5. Move hybrid application to another cloud with common infrastructures
6. Hybrid cloud application that uses platform services
7. Port cloud application that uses platform services to another cloud
8. Create cloud application with components that run on multiple clouds
9. Cloud application workload requires use of multiple clouds (Cloudburst)
10. Users and developers shop across hosted or public cloud offerings for best price/performance ratio, while optimizing against other considerations

Microsoft



euranova

# What kind of standards?

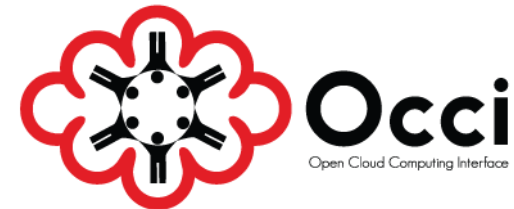
- Infrastructure management
  - API to create elastic group - Starting Removing VM – Resources management
- Appliance format
  - The package portability from one provider to another
- Storage & Core services
  - Standard interfaces

# Infrastructure management

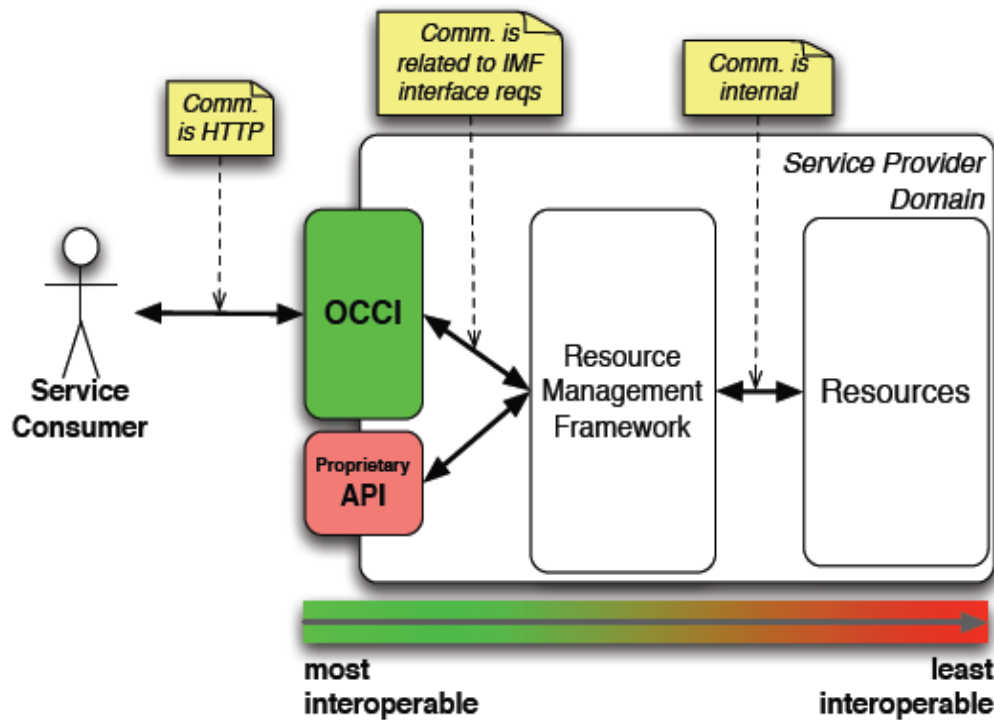
Emerging standards OCCI

Open Cloud Control Interface (OCCI)

- Open Grid Forum
- Supported by the EU via SIENA



<http://occi-wg.org/>

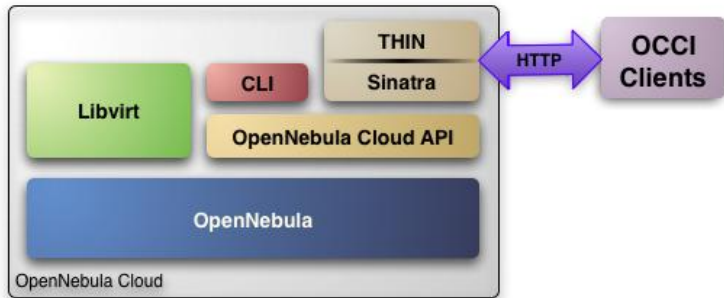


- Focus on Interoperability
- Easily extensible
  - Infrastructure
  - Platform
  - Service
- Different possible renderings
  - Only RESTful at the moment
- Reference implementation from OpenNebula

# Infrastructure management

## OCCI Reference implementation & tools

### OpenNebula using Sinatra



### OCCI compliance Test

The screenshot shows the 'OCCI compliance test' application window. It features a 'Go' button and a 'Reset' button. The 'OCCI service URL' is set to 'http://localhost:8888'. The 'Session information' section includes a checked 'Login required?' checkbox, a 'Username' field with 'foo', and a 'Password' field with '\*\*\*'. The 'Service information' section displays 'Server version: pyocci OCCI/1.1' and 'Number of registered categories: 18'. The 'Tests' section lists 11 tests, all of which are marked as 'OK'. A 'Quit' button is located at the bottom right.

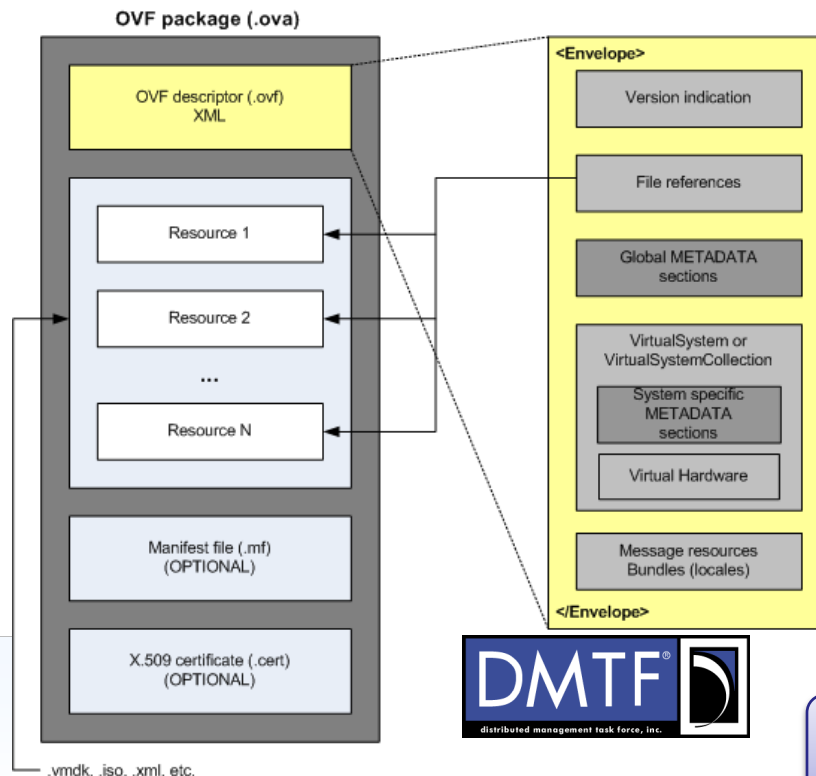
Tests	Result
Checking for correct version information:	OK
Checking completeness of infrastructure model:	OK
Checking correct handling of Content-type/Accept headers:	OK
Testing instantiation of compute/storage/network kinds:	OK
Testing correct handling of user-defined mixins (tagging/grouping):	OK
Testing links between compute/storage compute/network:	OK
Triggering actions on compute/network/storage kinds:	OK
Testing filter mechanisms using Categories:	OK
Testing correct behaviour on location and "normal" paths:	OK
Simple syntax checks:	OK

NOTE: Passing all tests only indicates that the service you are testing is OCCI compliant - IT DOES NOT GUARANTEE IT!

# Appliance format

Emerging standards OVF

- Defines the topology the service
- Contains the full software stack from OS to service
- Specify the HW of the VM



OpenNebula.org



Automatic deployment

Auto scaling rules

KPIs



# Storage & Core services

Enabling a real apps portability

- Applications usually use core services
  - Storage (S3, RDS, SimpleDB) – Messaging (SQS, Burrow) – Load Balancer (ELB), etc.
- Guaranteeing the portability means standardize the core services as well
  - Currently only storage is in the process of standardization

# Taxonomy

- 
- Clients acting in the role of using a Data Storage Interface**
- Clients can be in the cloud or enterprise and provide additional services (computing, data, etc.)
- Clients acting in the role of Managing Data/Storage**
- Management of the Cloud Storage can be standalone or part of the overall management of your cloud computing
- The diagram illustrates the interaction between clients and a central Data Storage Cloud. Clients acting in the role of using a Data Storage Interface (top) include Block Storage Client, Filesystem Client, Object Storage Client, XAM Client (XAM VIM for CDSI), and Database/Table Client. These clients interact with the Data Storage Cloud through various interfaces: iSCSI, FC, FCoE LUNs, Targets; POSIX (NFS, CIFS, WebDAV); SNIA Cloud Data Management Interface (CDMI); and Multiple, Proprietary Interfaces. The Data Storage Cloud contains Hard Data Containers, Soft Data Containers, and Database/Tables. Clients acting in the role of Managing Data/Storage (bottom) include a Data/Storage Management Client and a Cloud Data Management component. The Cloud Data Management component draws resources on demand from Data Services and Storage Services (represented by orange and green blocks) and interacts with Information Services (future) (represented by grey blocks). The SNIA Cloud Data Management Interface (CDMI) is also shown as a management interface.

# Storage interface

Emerging standard for content CDMI

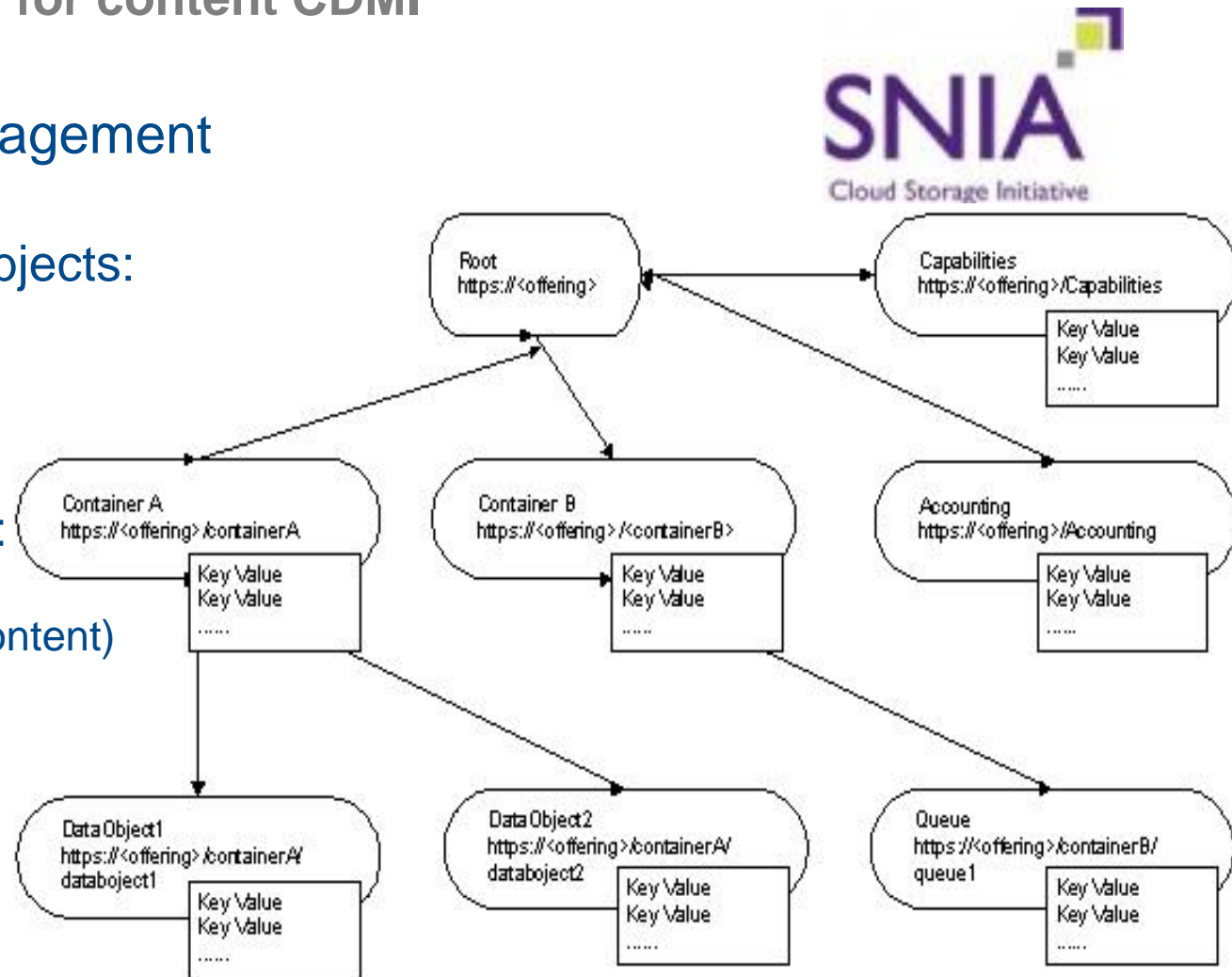
- Cloud Data Management Interface

- Hierarchy of objects:

- Container
    - Capabilities
    - Data
    - queue

- Two purposes:

- control path
    - data path (content)



# Storage interface

Emerging standard for Block-level

- Standardized APIs
  - iSCSI, CIFS, NFS, etc.
- Specialized hardware
  - EMC – NetApp – 3PAR
- Open source solution
  - DRDB - Lustre



# That is not enough

Need more standard

- Governance
  - Life-cycle management – CMDB – ADC – workflow management
- Billing & charging
  - Similar to DIAMETER Ro/Rf
- Metering
  - Measuring resource usage
- Cloud brokering
  - Managing a set of cloud providers

# QUESTIONS

**Euranova R&D**

<http://euranova.eu>

<http://twitter/sskhiri>

[sabri.skhiri@euranova.eu](mailto:sabri.skhiri@euranova.eu)