Management, Security and Sustainability for Cloud Computing

Carlos Becker Westphall

Networks and Management Laboratory

Federal University of Santa Catarina

MANAGEMENT FOR CLOUD COMPUTING

(Based on the reference – S. A. de Chaves, R. B. Uriarte, C. B. Westphall. Toward an Architecture for Monitoring Private Clouds. IEEE Communications Magazine. Dec. 2011.)

Outline

1. ABSTRACT
2. INTRODUCTION
3. BACKGROUND

3.1. Cloud Computing Service Models

3.2. Cloud Computing Deployment Models

3.3. Cloud Computing Standards

Outline

4. MONITORING ARCHITECTURE AND PCMONS

4.1. Architecture

4.2. Implemantation

5. CASE STUDY

6. RELATED WORK

6.1. Grid Monitoring

6.2. Cloud Monitoring

Outline

7. KEY LESSONS LEARNED

7.1. Related to Test-Bed Preparation

7.2. Design and Implementation

7.3. Standardization and Available Implementations

8. CONCLUSIONS AND FUTURE WORKS

9. SOME REFERENCES

1. ABSTRACT

This presentation describes:

- our experience with a private cloud;
- the design and implementation of a Private Cloud MONitoring System (PCMONS); and
- its application via a case study for the proposed architecture, using open source solutions and integrating with traditional tools like Nagios.

2. INTRODUCTION

- Cloud computing provides several technical benefits including flexible hardware and software allocation, elasticity, and performance isolation.
- Cloud management may be viewed as a specialization of distributed computing management, inheriting techniques from traditional computer network management.

2. INTRODUCTION

The intent of this presentation is to:

- Provide insight into how traditional tools and methods for managing network and distributed systems can be reused in cloud computing management.
- Introduce a Private Cloud MONitoring System (PCMONS) we developed to validate this architecture, which we intend to open source.

2. INTRODUCTION

- Help future adopters of could computing make good decisions on building their monitoring system in the cloud.
- We chose to address private clouds because they enable enterprises to reap cloud benefits while keeping their mission-critical data and software under their control and under the governance of their security policies.

3. BACKGROUND

3.1. Cloud Computing Service Models

- Software-as-a-Service (SaaS): The consumer uses the provider's applications, which are hosted in the cloud.
- Platform-as-a-Service (PaaS): Consumers deploy their own applications into the cloud infrastructure. Programming languages and applications development tools used must be supported by the provider.

3. BACKGROUND

3.1. Cloud Computing Service Models

- Infrastructure-as-a-Service (IaaS): Consumers are able to provision storage, network, processing, and other resources, and deploy and operate arbitrary software, ranging from applications to operating systems.
- This presetation focuses on IaaS model.

3. BACKGROUND

3.2. Cloud Computing Deployment Models

- Public: Resources are available to the general public over the Internet. In this case, "public" characterizes the scope of interface accessibility.
- Private: Resources are accessible within a private organization. This environment emphasizes the benefits of hardware investments.

3. BACKGROUND

3.2. Cloud Computing Deployment Models

- Community: Resources on this model are shared by several organizations with a common mission.

- Hybrid: This model mixes the techniques from public and private clouds. A private cloud can have its local infrastructure supplemented by computer capacity from public cloud.

3. BACKGROUND

3.3. Cloud Computing Standards

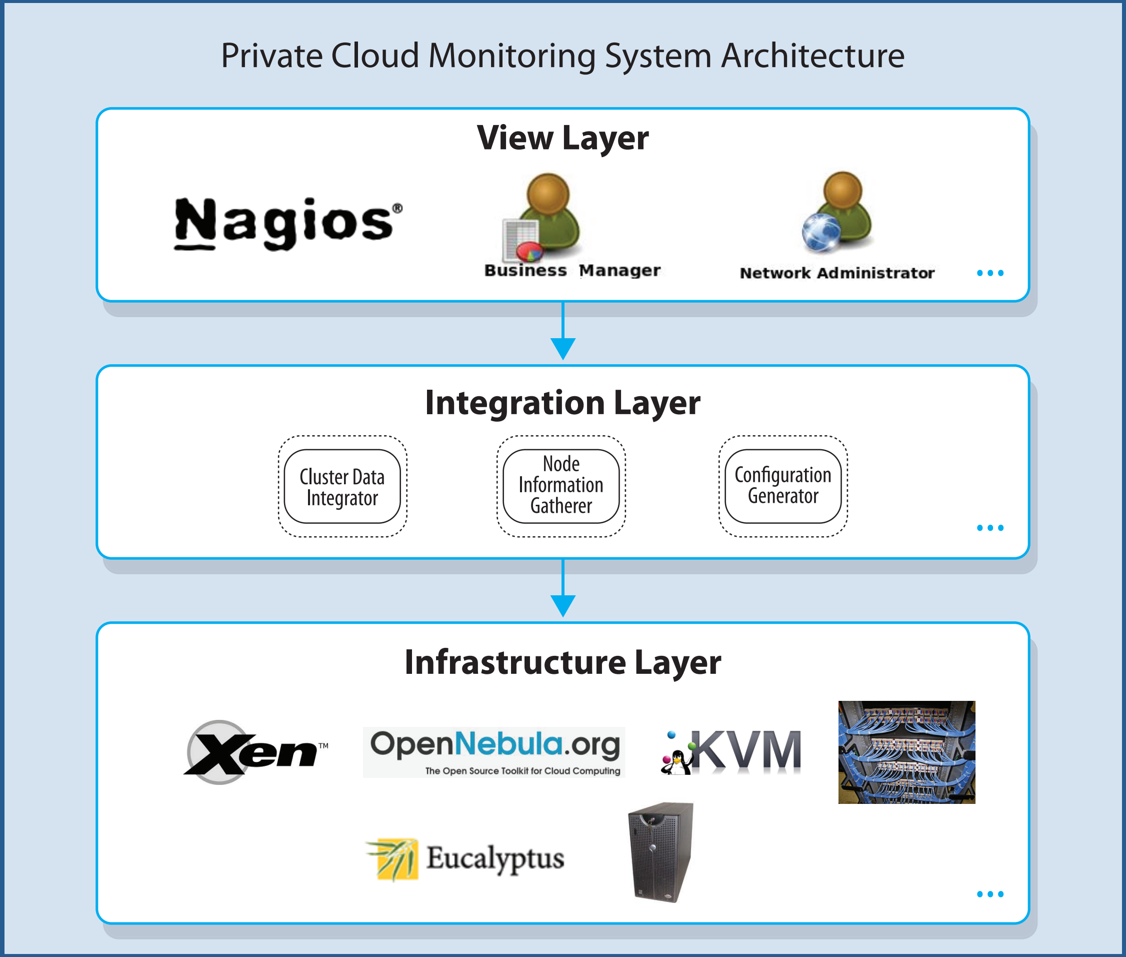
- Open Cloud Computing Interface: This Open Grid Forum group has a focus on specifications for interfacing "*aaS" cloud computing facilities.
- OCCI in Eucalyptus, OCCI in OpenStack, OCCI in OpenNebula...

3. BACKGROUND

3.3. Cloud Computing Standards

- Open Cloud Standards Incubator: This initiative, from Distributed Management Task Force (DMTF), focuses on interactions between cloud environments, their consumers, and developers.
- Example of document: "Use cases and Interactions for Managing Clouds".

4. MONITORING ARCHITECTURE AND PCMONS



Private Cloud Monitoring System Architecture

View Layer

Nagios

Business Manager

Network Administrator

...

Integration Layer

Cluster Data Integrator

Node Information Gatherer

Configuration Generator

...

Infrastructure Layer

4. MONITORING ARCHITECTURE AND PCMONS

4.1. Architecture

- Three layers address the monitoring needs of a private cloud.

Infrastructure layer:

- Basic facilities, services, and installations, such as hardware and networks;
- Available software: operating system, applications, licenses, hypervisors, and so on...

4. MONITORING ARCHITECTURE AND PCMONS

4.1. Architecture

Integration layer:

- The monitoring actions to be performed in the infrastructure layer must be systematized before passed to the appropriate service running in the integration layer.
- The integration layer is responsible for abstracting any infrastructure details.

4. MONITORING ARCHITECTURE AND PCMONS

4.1. Architecture

View layer:

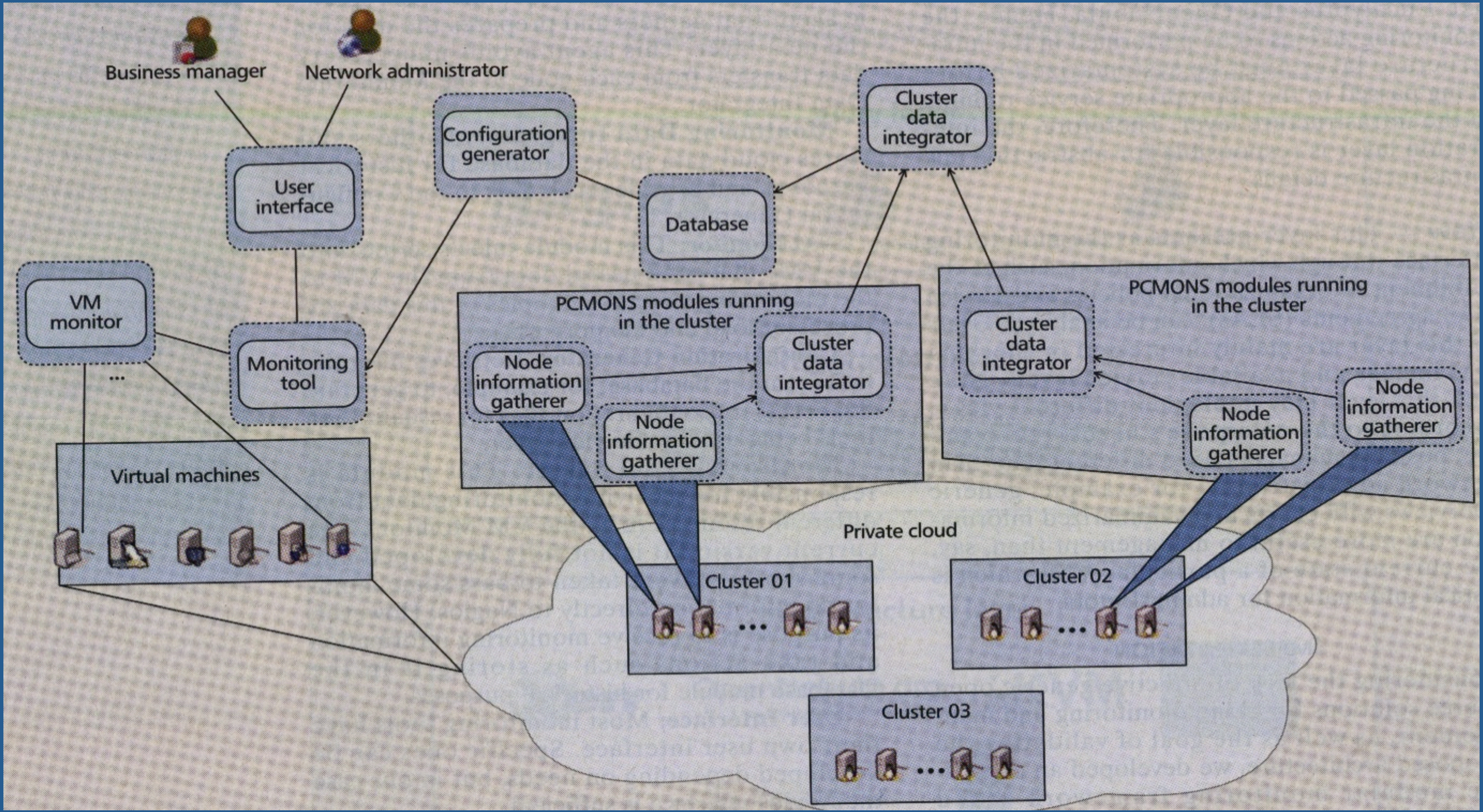
- This layer presents as the monitoring interface through which information, such as the fulfillment of organizational policies and service level agreements, can be analyzed.
- Users of this layer are mainly interested in checking VM images and available service levels.

4. MONITORING ARCHITECTURE AND PCMONS

4.2. Implementation

- The current PCMONS version acts principaly on the integration layer, by retrieving, gathering, and preparing relevant information for the visualization layer.
- The system is divided into the modules presented in the next figure and described below.

A typical deployment scenario for PCMONS



4. MONITORING ARCHITECTURE AND PCMONS

4.2 Implementation

- Node Information Gatherer: This module is responsible for gathering local information on a cloud node. It gathers information about local VMs and sends it to the Cluster Data Integrator.
- Cluster Data Integrator: It is a specific agent that gethers and prepares the data for the next level.

4. MONITORING ARCHITECTURE AND PCMONS

4.2 Implementation

- Monitoring Data Integrator: Gathers and stores cloud data in the database for historical purposes, and provides such data to the Configuration Generator.
- VM Monitor: This module injects scripts into the VMs that send useful data from the VM to the monitoring system.

4. MONITORING ARCHITECTURE AND PCMONS

4.2 Implementation

- Configuration Generator: Retrieves information from the database to generate configuration files for visualization tools.
- Monitoring Tool Server: Its purpose is to receive monitoring information and take actions such as storing it in the database module for historical purposes.

4. MONITORING ARCHITECTURE AND PCMONS

4.2 Implementation

- The Monitoting Tool Server generetes a configuration file that allows Nagions to monitor and display the monitoring information in its standard interface.
- Eucalyptus provides a simple Nagios script for basic monitoring of Eucalyptus components.

4. MONITORING ARCHITECTURE AND PCMONS

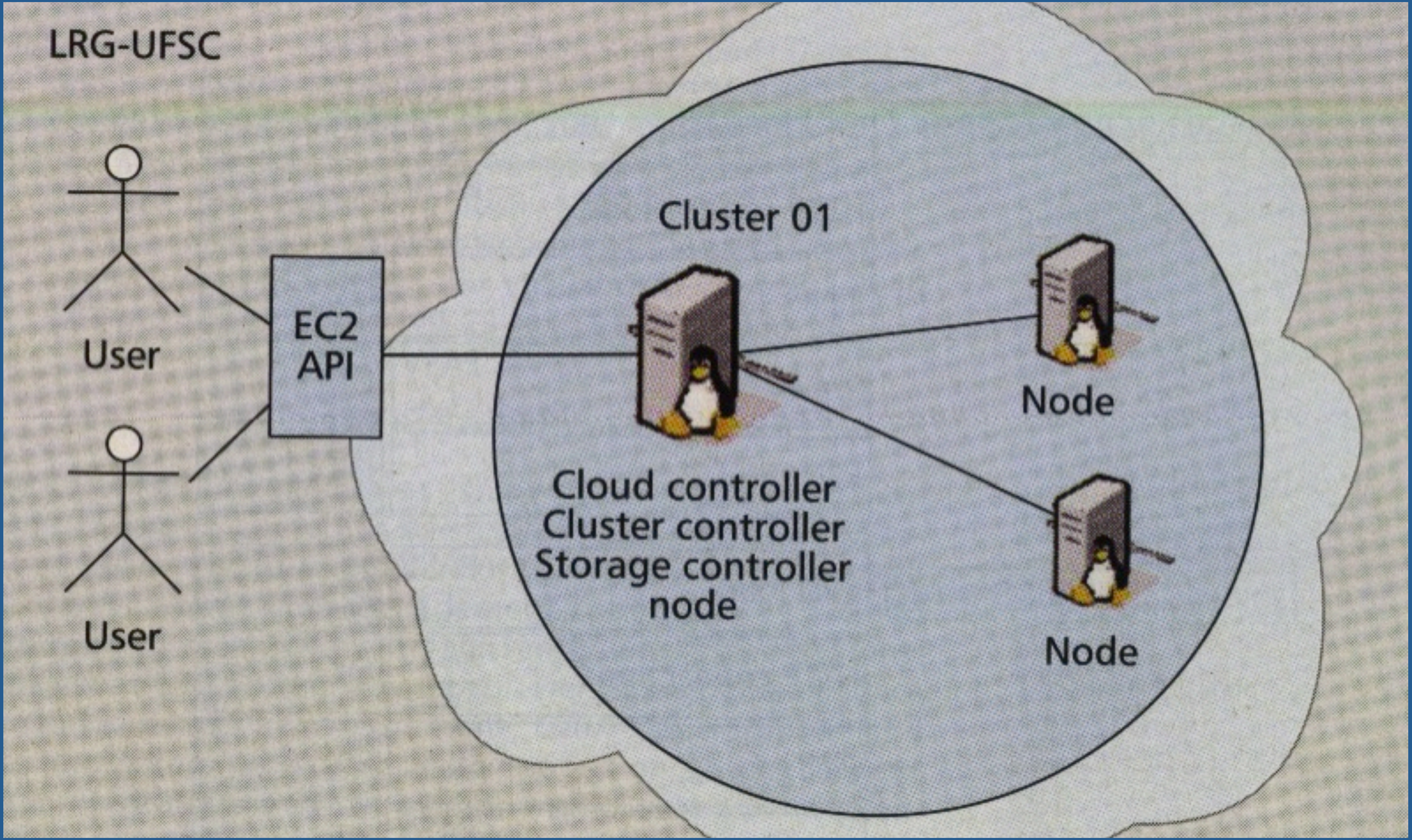
4.2 Implementation

- User Interface: Most monitoring tools have their own user interface. Specific ones can be developed depending on needs, but in our case the Nagios interface is sufficient.
- Database: Stores data needed by Configuration Generator and the Monitoring Data Integrator.

5. CASE STUDY

- We built an environment where VM images are available for users that instantiate a web server, thus simulating web hosting service provision.
- Instantiated VMs are Linux servers providing a basic set of tools, acting as web hosting servers.
- Apache Web Server, PHP language, SQLite.

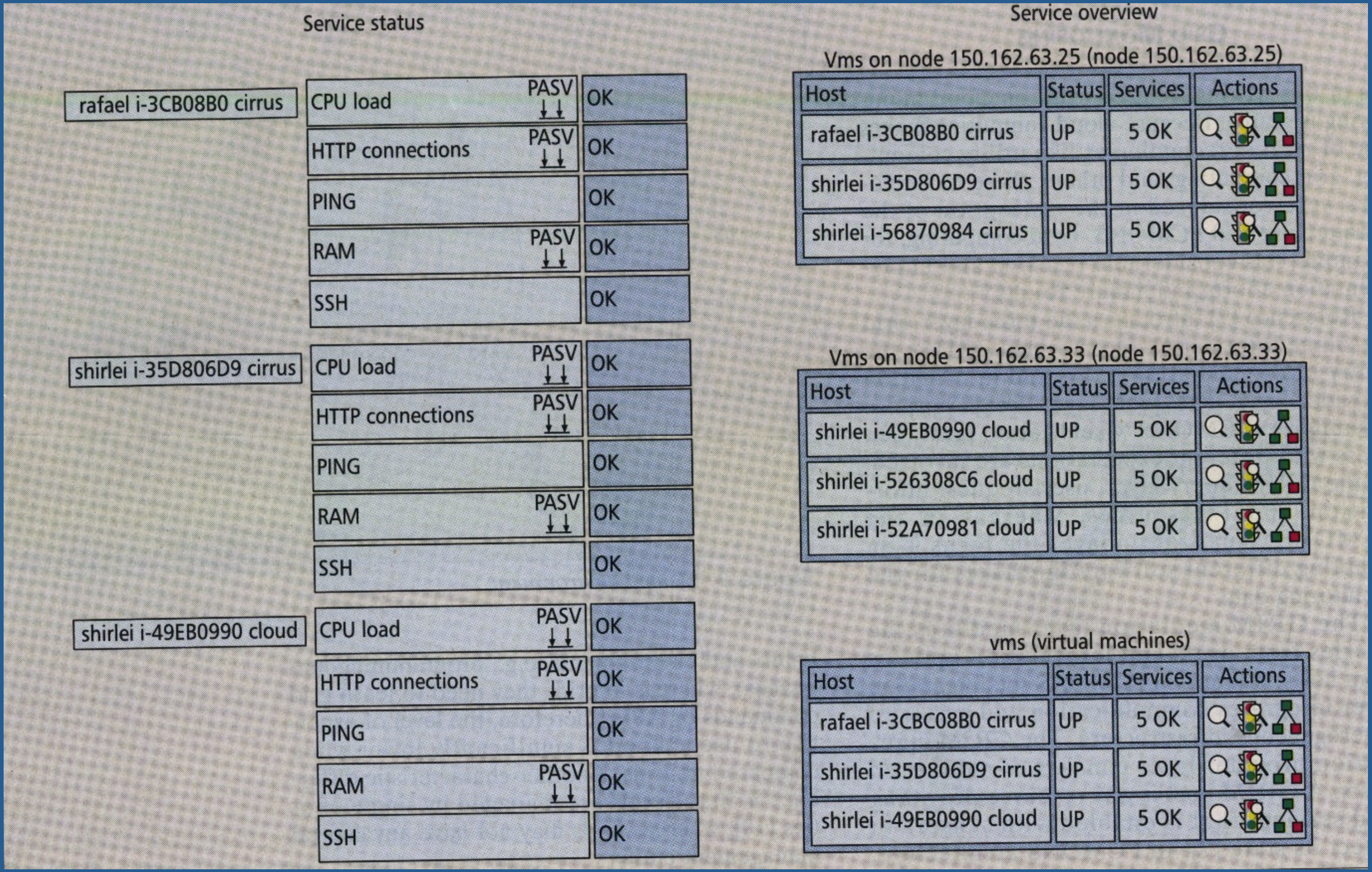
Testbed environment



5. CASE STUDY

- Open SUSE was chosen as the operating system of the physical machines (Xen and YaST).
- Eucalyptus (interface compatible with Amazon's EC2). VM images were downloaded from the Eucalyptus website.
- VM Monitor module is injectec into the VM during boot, allowing data monitoring.

Representative Nagios interface of the monitored cloud services



Service status

rafael i-3CB08B0 cirrus	CPU load	PASV ↓↓	OK
	HTTP connections	PASV ↓↓	OK
	PING		OK
	RAM	PASV ↓↓	OK
	SSH		OK
shirlei i-35D806D9 cirrus	CPU load	PASV ↓↓	OK
	HTTP connections	PASV ↓↓	OK
	PING		OK
	RAM	PASV ↓↓	OK
	SSH		OK
shirlei i-49EB0990 cloud	CPU load	PASV ↓↓	OK
	HTTP connections	PASV ↓↓	OK
	PING		OK
	RAM	PASV ↓↓	OK
	SSH		OK

Service overview

Vms on node 150.162.63.25 (node 150.162.63.25)

Host	Status	Services	Actions
rafael i-3CB08B0 cirrus	UP	5 OK	
shirlei i-35D806D9 cirrus	UP	5 OK	
shirlei i-56870984 cirrus	UP	5 OK	

Vms on node 150.162.63.33 (node 150.162.63.33)

Host	Status	Services	Actions
shirlei i-49EB0990 cloud	UP	5 OK	
shirlei i-526308C6 cloud	UP	5 OK	
shirlei i-52A70981 cloud	UP	5 OK	

vms (virtual machines)

Host	Status	Services	Actions
rafael i-3CBC08B0 cirrus	UP	5 OK	
shirlei i-35D806D9 cirrus	UP	5 OK	
shirlei i-49EB0990 cloud	UP	5 OK	

5. CASE STUDY

- First column shows object names (VM, PM, ROUTERS...). VM names are an aggregation of user name, VM ID, and name of PM where the VM is running.
- The other two columns show service names and their status (OK, Warning, Critical).
- It shows host group created by PCMONS and VM/VP mapping.

6. RELATED WORK

6.1. Grid Monitoring

- Reference [7] introduces the three-layer Grid Resource Information Monitoring (GRIM).
- Several design issues that should be considered when constructing a Grid Monitoring System (GMS) are preented in [8]. We have selected some and correlated them with PCMOMS.

6. RELATED WORK

6.1. Grid Monitoring

- Reference [9] identifies some differences between cloud monitoring and grid monitoring, especially in termes of interfaces and service provisioning.
- Another diference is that clouds are managed by single entities [10], whereas grids may not have any central management entity.

6. RELATED WORK

6.2. Cloud Monitoring

- Reference [11] defines general requirements for cloud monitoring and proposes a cloud monitoring framework.
- PCMONS supports two approches, agents and central monitoring, and is highly adaptable, making the migration to a privite cloud straighforward.

7. KEY LESSONS LEARNED

7.1. Related to Test-Bed Preparation

- Software platforms for cloud computing, such as Eucalyptus and OpenNebula, support a number of different hypervisors, each with its own characteristics.
- An example is the KVM hypervisor: it has great performance but requires hardware virtualization that not all processors provide.

7. KEY LESSONS LEARNED

7.2. Design and Implementation

- We opted for solutions well established in the market to facilatate the use of PCMONS in the running structures with little effort and prioritized an adaptable and extensible solution.
- We planned to define some basic common metrics for private clouds, but later found that metrics are often specific to each case.

7. KEY LESSONS LEARNED

7.3. Standardization and Available Implementations

- Before choosing a specific tool for private clouds, it is important to verify to what extent cloud standards are implemented by the tool.
- Some tools, such as OpenNebula, have begun implementing standardization efforts, including the OCCI API.

8. CONCLUSION AND FUTURE WORK

- This presentation summarizes some cloud computing concepts and our personal experience with this new paradigm.
- The current portfolio of open tools lacks open source, interoperable management and monitoring tools. To address this critical gap, we designed a monitoring architecture, and validade the architecture by developing PCMONS.

8. CONCLUSION AND FUTURE WORK

- To monitor specific metrics, especially in an interface-independent manner, a set of preconfigured monitoring plug-ins must be developed.
- For future work, we intend to improve PCMONS to monitor other metrics and suport other open source tools like OpenNebula, OpenStack...

9. REFERENCES

References indicated in this presentation:

- [7] W. Chung and R. Chang, “A New Mechanism for Resource Monitoring in Grid Computing,” Future Gen. Comp. Sys. Jan. 2009.
- [8] M. Yiduo et al., “Rapid and Automated Deployment of Monitoring Services in Grid Environments,” APSCC, 2007.
- [9] L. Wang et al., “Scientific Cloud Computing: Early Definition and Experience,” IEEE Int’l. Conf. High Perf. Computing and Commun., 2008.

9. REFERENCES

References indicated in this presentation:

- [10] M. Brock and A. Goscinski, “Grids vs. Clouds,” IEEE 2010 5th Int’l. Conf. Future Info. Tech., 2010.
- [11] P. Hasselmeyer and N. d’Heureuse, “Towards Holistic Multi-Tenant Monitoring for Virtual Data Centers,” IEEE/IFIP NOMS Wksps., 2010.

SECURITY FOR CLOUD COMPUTING

(Based on the reference: – M. A. P. Leandro, T. J. Nascimento, D. R. Santos, C. M. Westphall, C. B. Westphall. Multi-Tenancy Authorization System with Federated Identity to Cloud Environment Using Shibboleth. International Conference on Networks. Feb. 2012.)

Content at a Glance

- Introduction and Related Works
- Cloud Computing
- Identity Management
- Shibboleth
- Federated Multi-Tenancy Authorization System on Cloud
  - Scenario
  - Implementation of the Proposed Scenario
  - Analysis and Test Results within Scenario
- Conclusions and Future Works

Introduction

- Cloud computing systems: reduced upfront investment, expected performance, high availability, infinite scalability, fault-tolerance.
- IAM (Identity and Access Management) plays an important role in controlling and billing user access to the shared resources in the cloud.

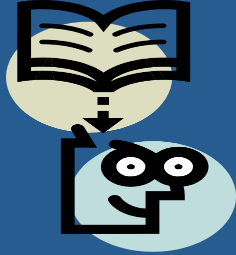
Introduction

- IAM systems need to be protected by federations.
- Some technologies implement federated identity, such as the SAML (Security Assertion Markup Language) and Shibboleth system.
- The aim of this paper is to propose a multi-tenancy authorization system using Shibboleth for cloud-based environments.

Related Work

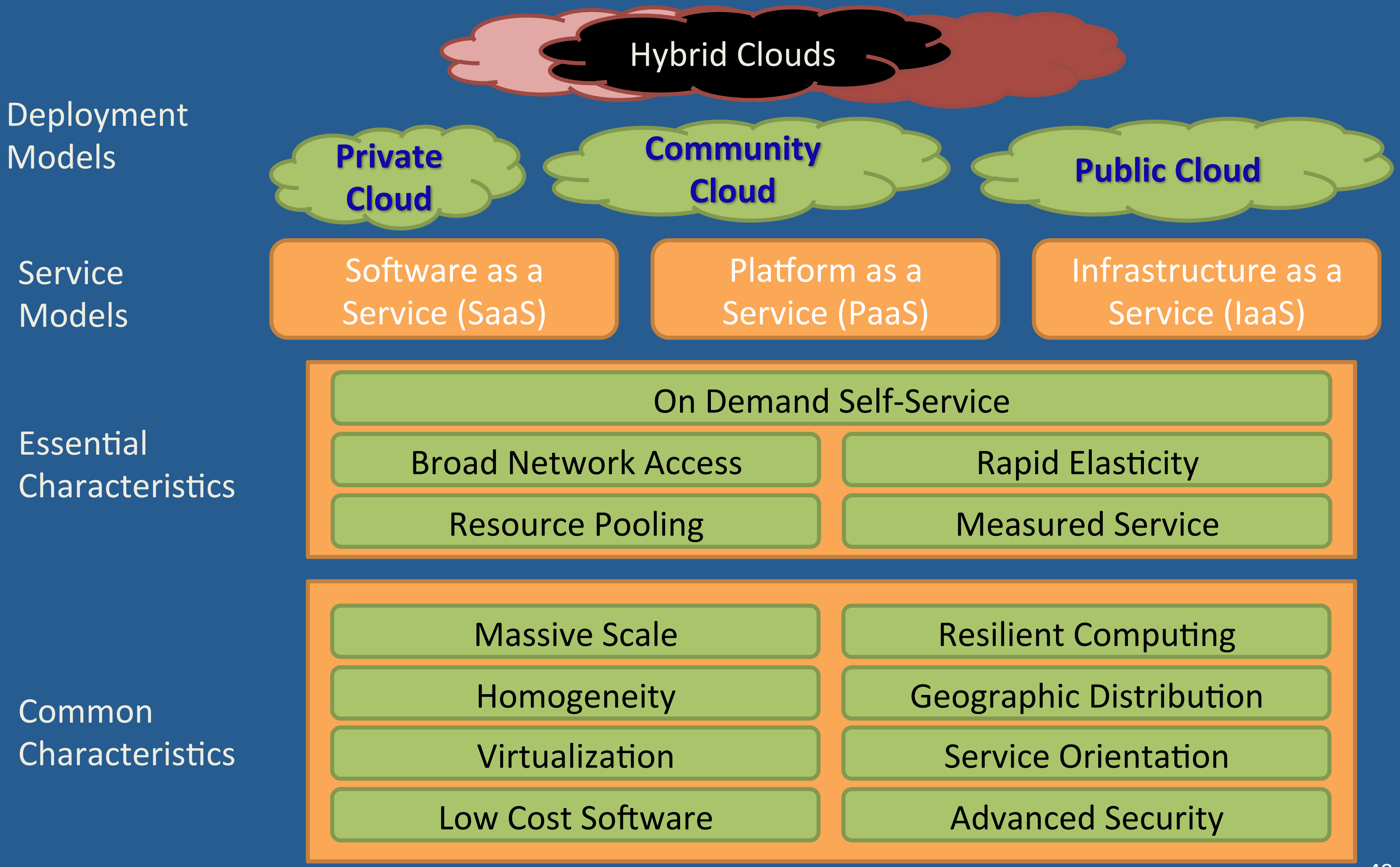
- R. Ranchal et al. 2010 - an approach for IDM is proposed, which is independent of Trusted Third Party (TTP) and has the ability to use identity data on untrusted hosts.
- P. Angin et al. 2010 - an entity-centric approach for IDM in the cloud is proposed. They proposed the cryptographic mechanisms used in R. Ranchal et al. without any kind of implementation or validation.

This Work



- Provide identity management and access control and aims to: (1) be an independent third party; (2) authenticate cloud services using the user's privacy policies, providing minimal information to the Service Provider (SP); (3) ensure mutual protection of both clients and providers.
- This paper highlights the use of a specific tool, Shibboleth, which provides support to the tasks of authentication, authorization and identity federation.
- The main contribution of our work is the implementation in cloud and the scenario presented.

The NIST Cloud Definition Framework



Hybrid Clouds

Deployment Models

Private Cloud

Community Cloud

Public Cloud

Service Models

Software as a Service (SaaS)

Platform as a Service (PaaS)

Infrastructure as a Service (IaaS)

Essential Characteristics

On Demand Self-Service

Broad Network Access

Rapid Elasticity

Resource Pooling

Measured Service

Common Characteristics

Massive Scale

Resilient Computing

Homogeneity

Geographic Distribution

Virtualization

Service Orientation

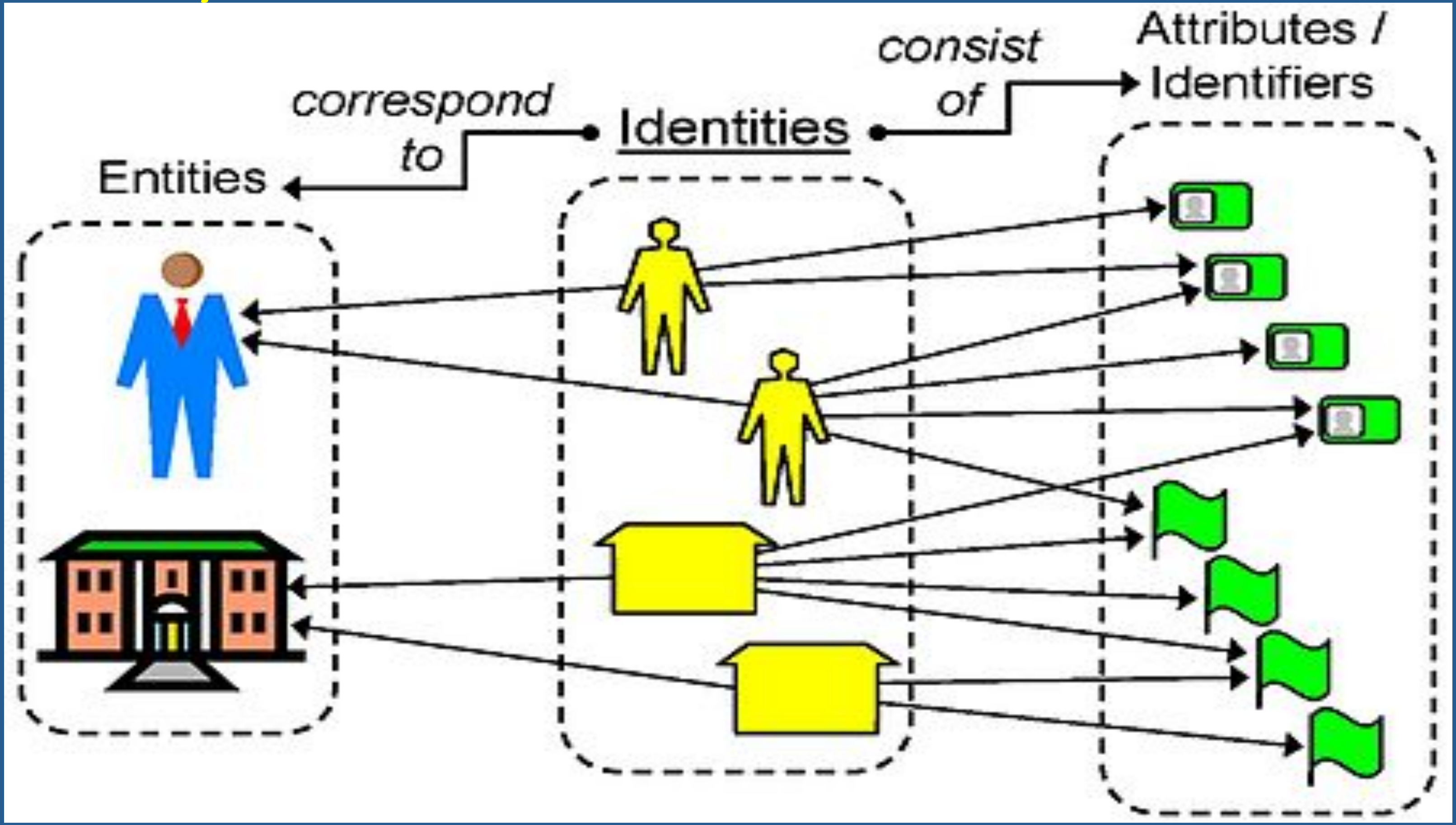
Low Cost Software

Advanced Security

Based upon original chart created by Alex Dowbor

Identity Management

- Digital identity is the representation of an entity in the form of attributes.



http://en.wikipedia.org/wiki/Identity_management

Identity Management

- Identity Management (IdM) is a set of functions and capabilities used to ensure identity information, thus assuring security.
- An Identity Management System (IMS) provides tools for managing individual identities.
- An IMS involves:
  - User
  - Identity Provider (IdP)
  - Service Provider (SP)

IMS

- *Provisioning:* addresses the provisioning and deprovisioning of several types of user accounts.
- *Authentication:* ensures that the individual is who he/she claims to be.
- *Authorization:* provide different access levels for different parts or operations within a computing system.
- *Federation:* it is a group of organizations or SPs that establish a circle of trust.

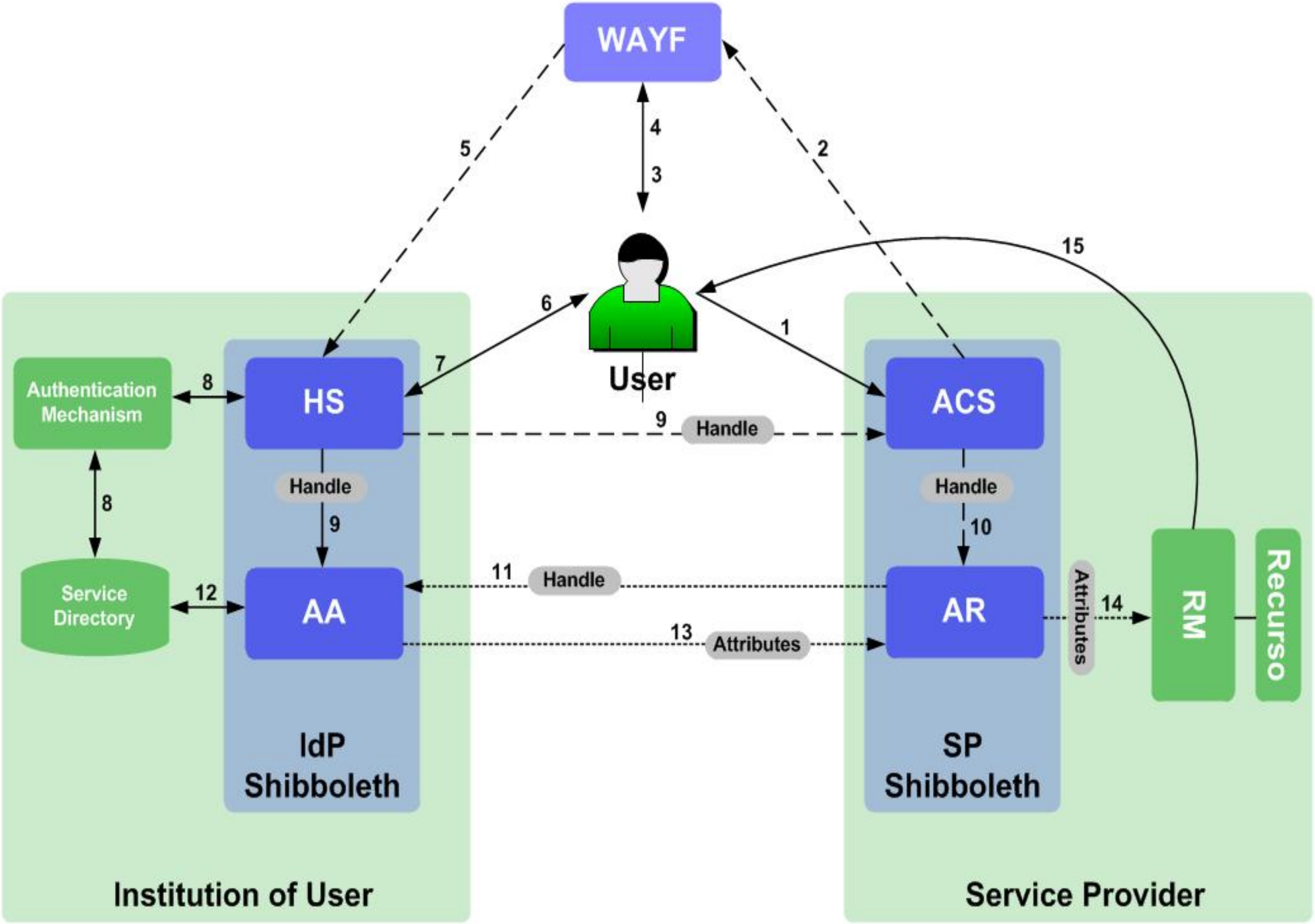
Shibboleth

A Project of the Internet2 Middleware Initiative

- The OASIS SAML (Security Assertion Markup Language) standard defines precise syntax and rules for requesting, creating, communicating, and using SAML assertions.
- The Shibboleth is an authentication and authorization infrastructure based on SAML that uses the concept of federated identity. The Shibboleth system is divided into two entities: the IdP and SP.

Shibboleth

- The IdP is the element responsible for authenticating users: Handle Service (HS), Attribute Authority (AA), Directory Service, Authentication Mechanism.
- The SP Shibboleth is where the resources are stored: Assertion Consumer Service (ACS), Attribute Requester (AR), Resource Manager (RM).
- The WAYF ("Where Are You From", also called the Discovery Service) is responsible for allowing an association between a user and organization.

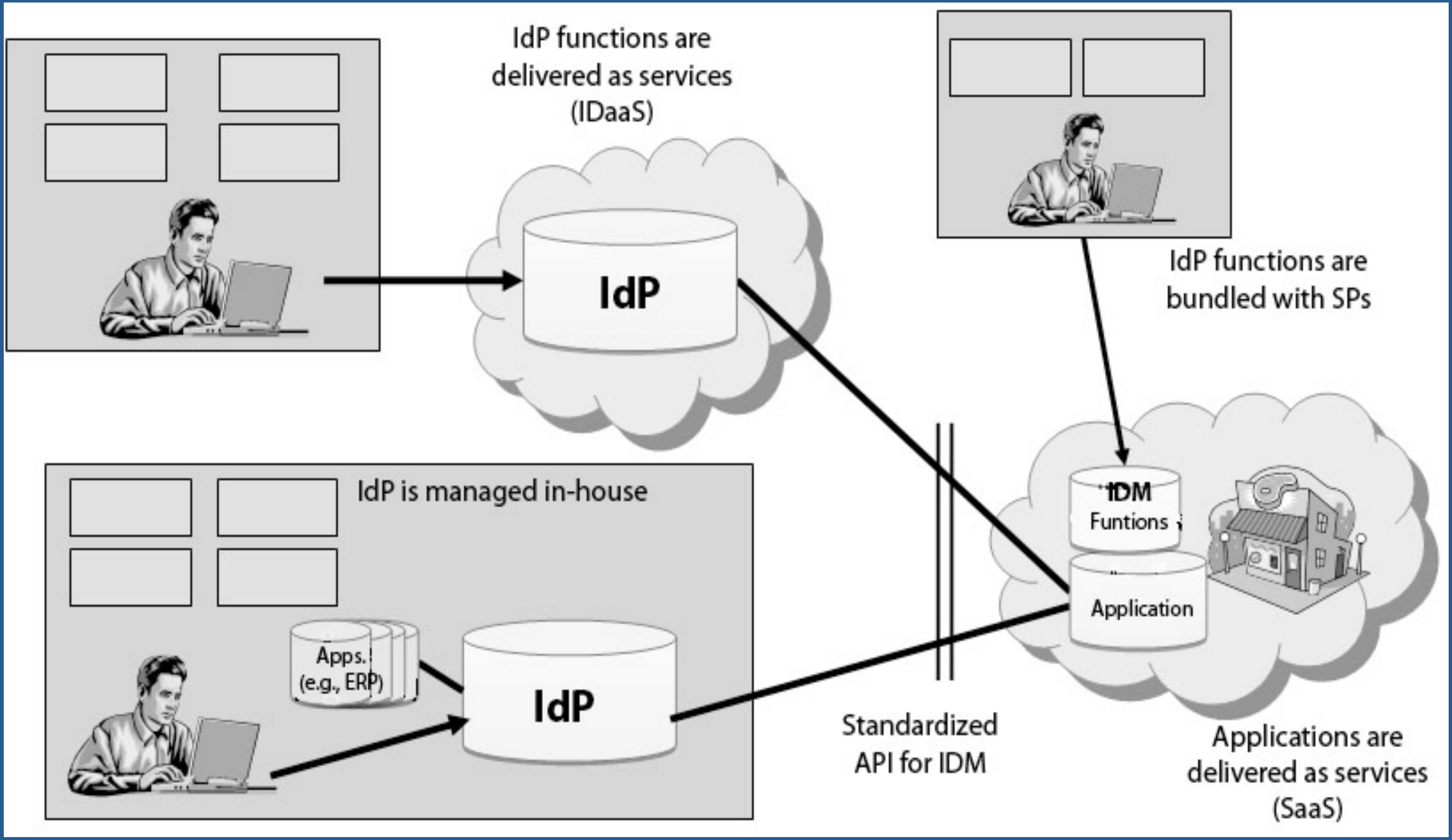

WAYF
5
4
3
2
15
6
1
User
7
Authentication Mechanism
8
HS
9
Handle
ACS
Handle
Handle
9
10
8
Service Directory
12
AA
11
Handle
AR
Attributes
14
RM
Recurso
13
Attributes
IdP Shibboleth
SP Shibboleth
Institution of User
Service Provider

In Step 1, the user navigates to the SP to access a protected resource. In Steps 2 and 3, Shibboleth redirects the user to the WAYF page, where he should inform his IdP. In Step 4, the user enters his IdP, and Step 5 redirects the user to the site, which is the component HS of the IdP. In Steps 6 and 7, the user enters his authentication data and in Step 8 the HS authenticate the user. The HS creates a handle to identify the user and sends it also to the AA. Step 9 sends that user authentication handle to AA and to ACS. The handle is checked by the ACS and transferred to the AR, and in Step 10 a session is established. In Step 11 the AR uses the handle to request user attributes to the IdP. Step 12 checks whether the IdP can release the attributes and in Step 13 the AA responds with the attribute values. In Step 14 the SP receives the attributes and passes them to the RM, which loads the resource in Step 15 to present to the user.

Federated Multi-Tenancy Authorization System on Cloud

- IdM can be implemented in several different types of configuration:
  - IdM can be implemented in-house;
  - IdM itself can be delivered as an outsourced service. This is called Identity as a Service (IDaaS);
  - Each cloud SP may independently implement a set of IdM functions.
- In this work, it was decided to use the first case configuration: in-house.

Configurations of IDM systems on cloud computing environments


IdP functions are delivered as services (IDaaS)
IdP
IdP functions are bundled with SPs
IDM Funtions
Application
IdP is managed in-house
Apps. (e.g., ERP)
IdP
Standardized API for IDM
Applications are delivered as services (SaaS)

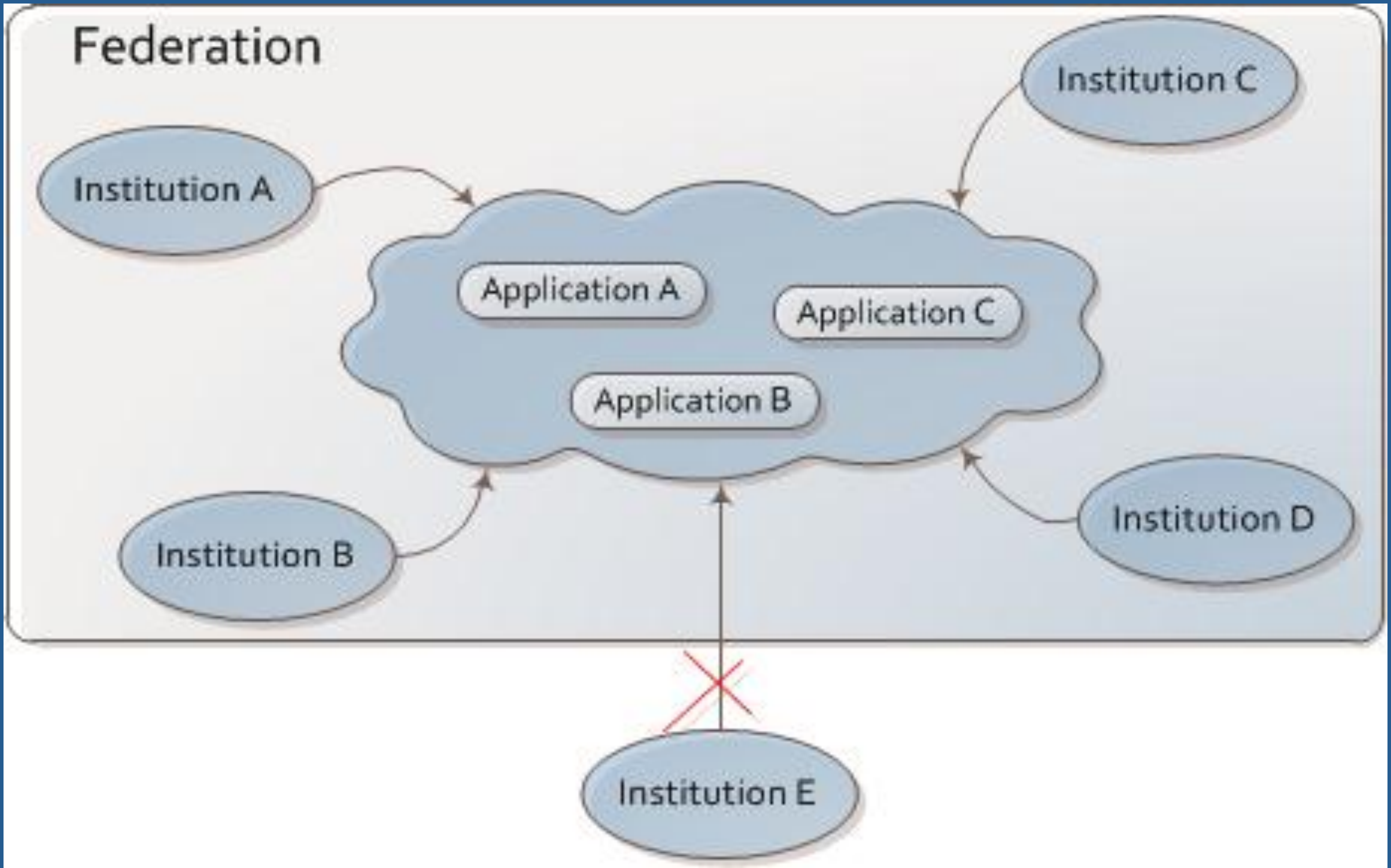
Federated Multi-Tenancy Authorization System on Cloud

- This work presents an authorization mechanism to be used by an academic institution to offer and use the services offered in the cloud.
- The part of the management system responsible for the authentication of identity will be located in the client organization.
- The communication with the SP in the cloud (Cloud Service Provider, CSP) will be made through identity federation.
- The access system performs authorization or access control in the environment.
- The institution has a responsibility to provide the user attributes for the deployed application SP in the cloud.
- The authorization system should be able to accept multiple clients, such as a multi-tenancy.

Scenario

- A service is provided by an academic institution in a CSP, and shared with other institutions. In order to share services is necessary that an institution is affiliated to the federation.
- For an institution to join the federation it must have configured an IdP that meets the requirements imposed by the federation.
- Once affiliated with the federation, the institution will be able to authenticate its own users, since authorization is the responsibility of the SP.

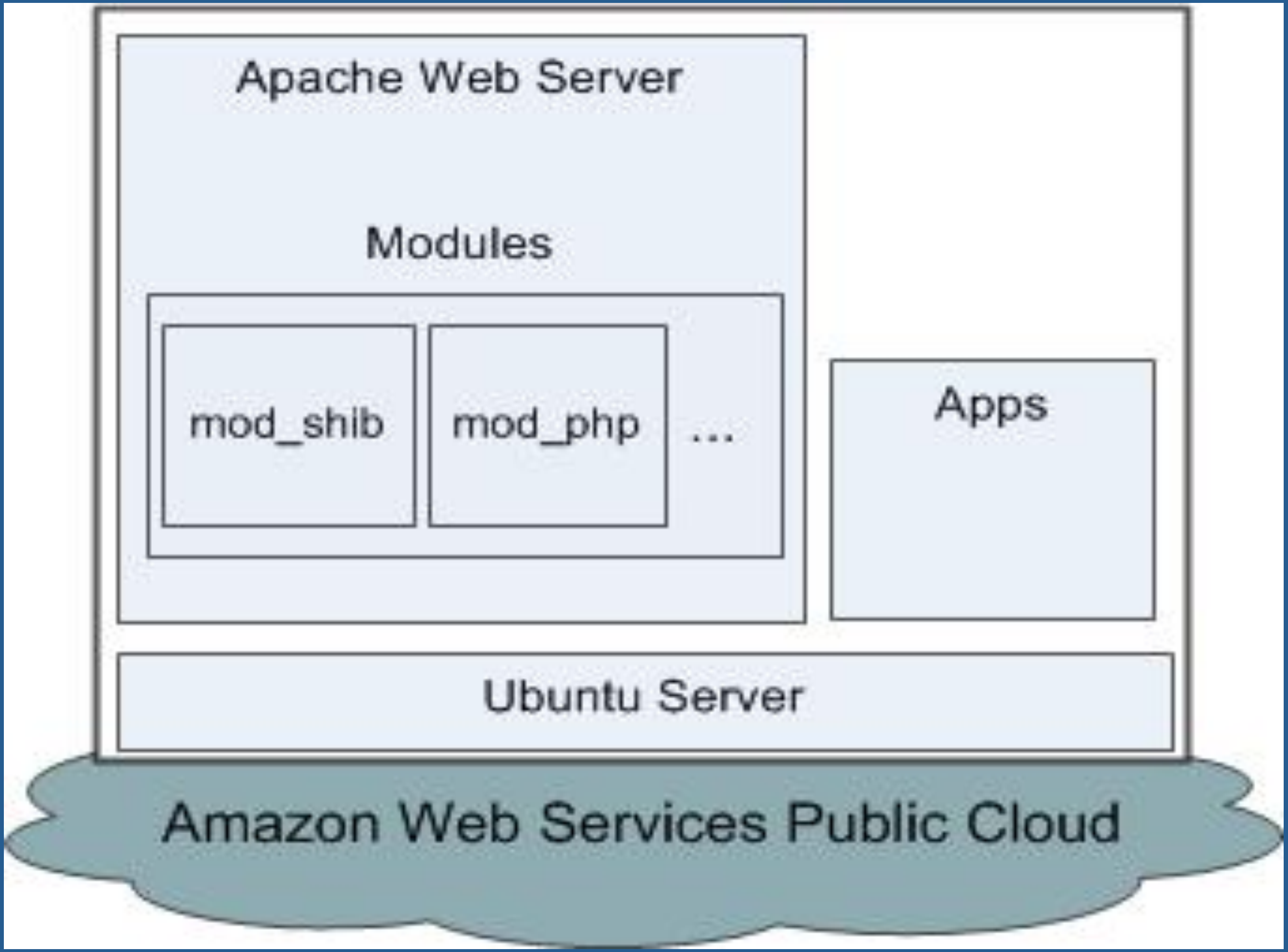
Scenario - Academic Federation sharing services in the cloud



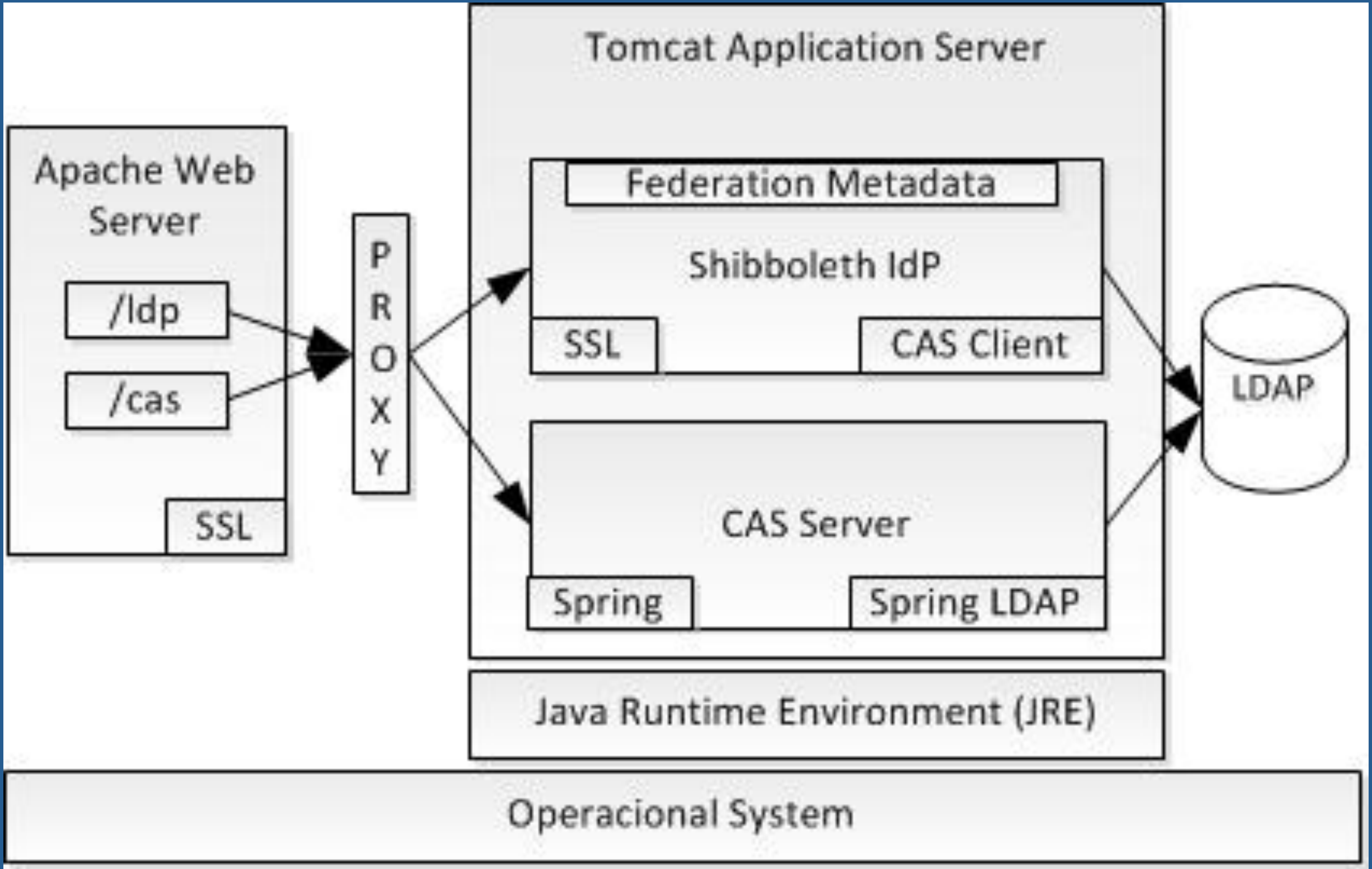
Implementation of the Proposed Scenario

- A SP was primarily implemented in the cloud:
  - an Apache server on a virtual machine hired by the Amazon Web Services cloud.
  - Installation of the Shibboleth SP.
  - Installation of DokuWiki, which is an application that allows the collaborative editing of documents.
  - The SP was configured with authorization via application, to differentiate between common users and administrators of Dokuwiki.

Implementation of the Proposed Scenario – Cloud Service Provider



Implementation of the Proposed Scenario – cloud IdP



Implementation of the Proposed Scenario

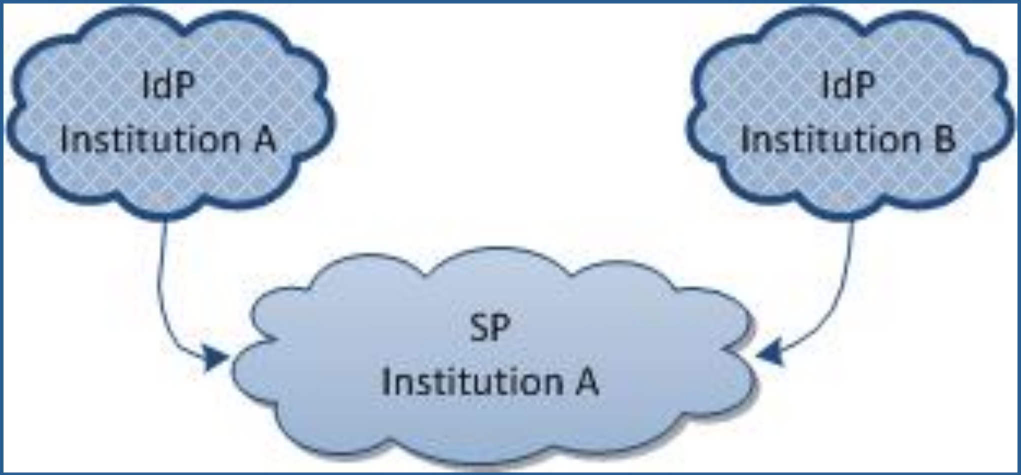
- The JASIG CAS Server was used to perform user authentication through login and password, and then passes the authenticated users to Shibboleth.
- The CAS has been configured to search for users in a Lightweight Directory Access Protocol (LDAP). To use this directory OpenLDAP was installed in another virtual machine, also running on Amazon's cloud.
- To demonstrate the use of SP for more than one client, another IdP was implemented, also in cloud, similar to the first. To support this task Shibboleth provides a WAYF component.

Analysis and Test Results within Scenario

- In this resulting structure, each IdP is represented in a private cloud, and the SP is in a public cloud.

The results highlighted two main use cases:

- *Read access to documents*
- *Access for editing documents*



Conclusions

- The use of federations in IdM plays a vital role.
- This work was aimed at an alternative solution to a IDaaS. IDaaS is controlled and maintained by a third party.
- The infrastructure obtained aims to: (1) be an independent third party, (2) authenticate cloud services using the user's privacy policies, providing minimal information to the SP, (3) ensure mutual protection of both clients and providers.

Conclusions

- This paper highlights the use of a specific tool, Shibboleth, which provides support to the tasks of authentication, authorization and identity federation.
- Shibboleth was very flexible and it is compatible with international standards.
- It was possible to offer a service allowing public access in the case of read-only access, while at the same time requiring credentials where the user must be logged in order to change documents.

Future Work

- We propose an alternative authorization method, where the user, once authenticated, carries the access policy, and the SP should be able to interpret these rules.
- The authorization process will no longer be performed at the application level.
- Expanding the scenario to represent new forms of communication.
- Create new use cases for testing.
- Use pseudonyms in the CSP domain.

Some References

- E. Bertino, and K. Takahashi, Identity Management - Concepts, Technologies, and Systems. ARTECH HOUSE, 2011.

- "Security Guidance for Critical Areas of Focus in Cloud Computing," CSA. Online at: http://www.cloudsecurityalliance.org.

- "Domain 12: Guidance for Identity and Access Management V2.1.," Cloud Security Alliance. - CSA. Online at: https://cloudsecurityalliance.org/guidance/csaguide-dom12-v2.10.pdf.

- D. W. Chadwick, Federated identity management. Foundations of Security Analysis and Design V, Springer-Verlag: Berlin, Heidelberg 2009 pp. 96–120.

Some References

- A. Albeshri, and W. Caelli, “Mutual Protection in a Cloud Computing environment,” Proc. 12th IEEE Intl. Conf. on High Performance Computing and Communications (HPCC 10), pp. 641-646.

- R. Ranchal, B. Bhargava, A. Kim, M. Kang, L. B. Othmane, L. Lilien, and M. Linderman, “Protection of Identity Information in Cloud Computing without Trusted Third Party,” Proc. 29th IEEE Intl. Symp. on Reliable Distributed Systems (SRDS 10), pp. 368–372.

- P. Angin, B. Bhargava, R. Ranchal, N. Singh, L. B. Othmane, L. Lilien, and M. Linderman, “An Entity-Centric Approach for Privacy and Identity Management in Cloud Computing,” Proc. 29th IEEE Intl. Symp. on Reliable Distributed Systems (SRDS 10), pp. 177–183.

SUSTAINABILITY FOR CLOUD COMPUTING

(Based on the reference: - J. Werner, G. A. Geronimo, C. B. Westphall, F. L. Koch, R. R. de Freitas, C. M. Westphall. Environment, Services and Network Management for Green Clouds. CLEI Electronic Journal. Aug. 2012.)

Summary

1 - Introduction

2 - Motivation

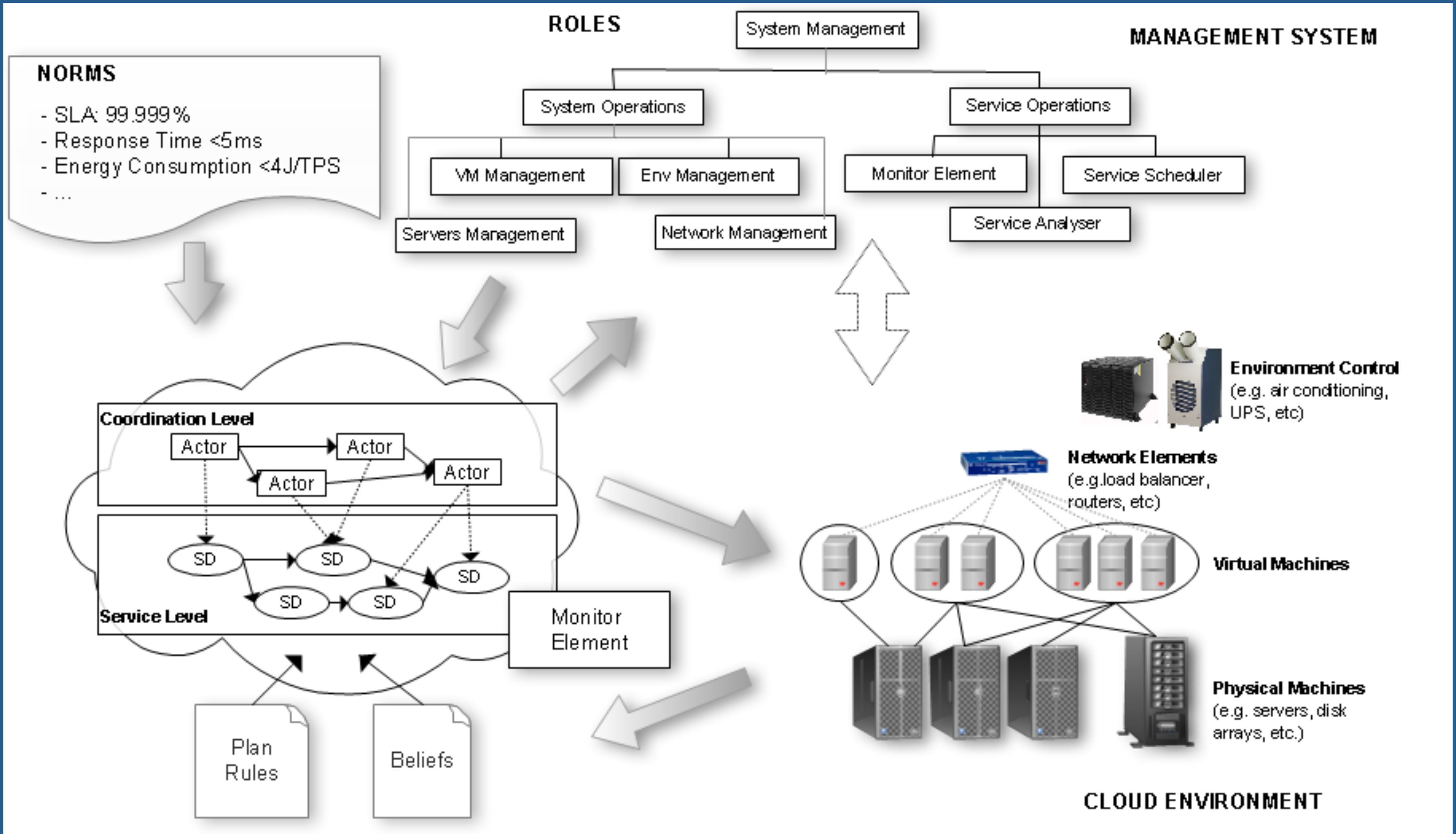
3 - Proposals and Solutions

4 - Case Studies

5 - Results

6 - Conclusions

7 - Future Works



NORMS

- SLA: 99.999%
- Response Time <5ms
- Energy Consumption <4J/TPS
- ...

ROLES

MANAGEMENT SYSTEM

System Management

System Operations

Service Operations

VM Management

Env Management

Servers Management

Network Management

Monitor Element

Service Scheduler

Service Analyser

Coordination Level

Actor

Actor

Actor

Actor

Service Level

SD

SD

SD

SD

SD

Monitor Element

Plan Rules

Beliefs

Environment Control (e.g. air conditioning, UPS, etc)

Network Elements (e.g.load balancer, routers, etc)

Virtual Machines

Physical Machines (e.g. servers, disk arrays, etc.)

CLOUD ENVIRONMENT

1 Introduction

- We propose an integrated solution for environment, services and network management based on organization theory model.

- This work introduces the system management model, analyses the system's behavior, describes the operation principles, and presents case studies and some results.

1 Introduction

- We extended CloudSim to simulate the organization model approach and implemented the migration and reallocation policies using this improved version to validate our management solution.

- Organization:

  2 introduces a motivating scenario.

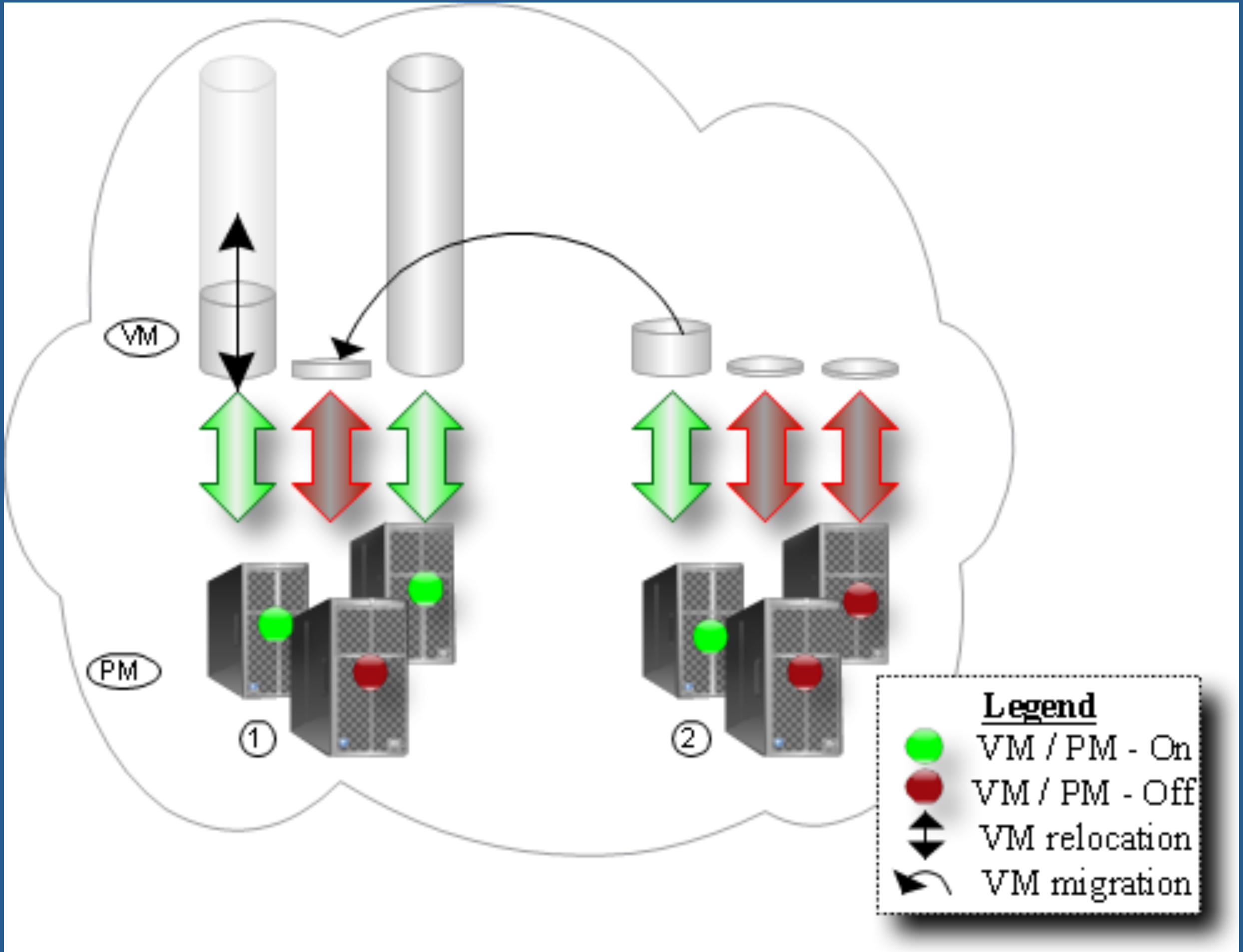
  3 outlines the system design.

  4 presents case studies.

  5 presents some results.

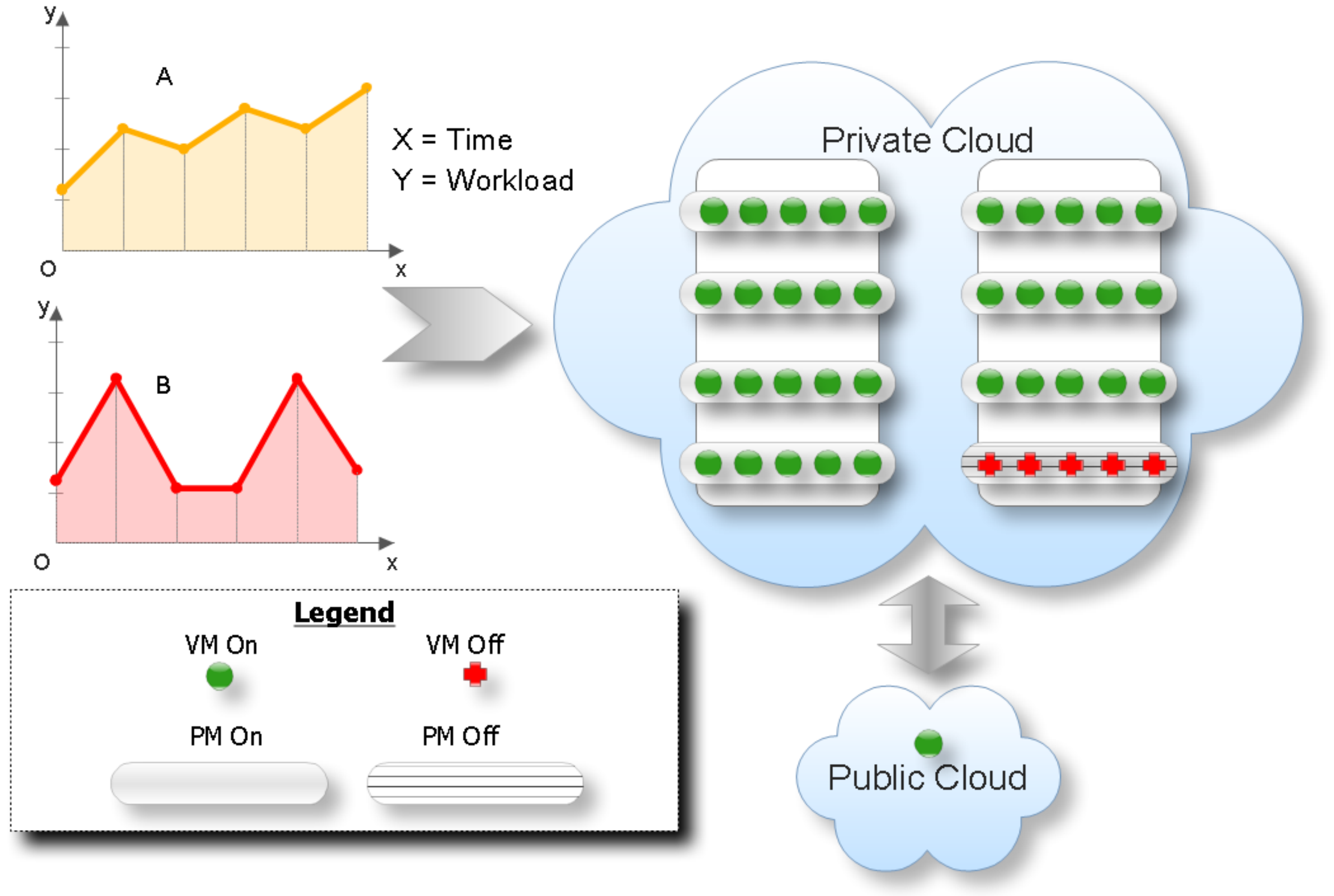
2 Motivation

- Our research was motivated by a practical scenario at our university's data center.
- Organization theory model for integrated management of the green clouds focusing on:
- (i) optimizing resource allocation through predictive models;


VM
PM
1
2
Legend
VM / PM - On
VM / PM - Off
VM relocation
VM migration

2 Motivation

- (ii) coordinating control over the multiple elements, reducing the infrastructure utilization;
- (iii) promoting the “balance” between local and remote resources; and
- (iv) aggregating energy management of network devices.

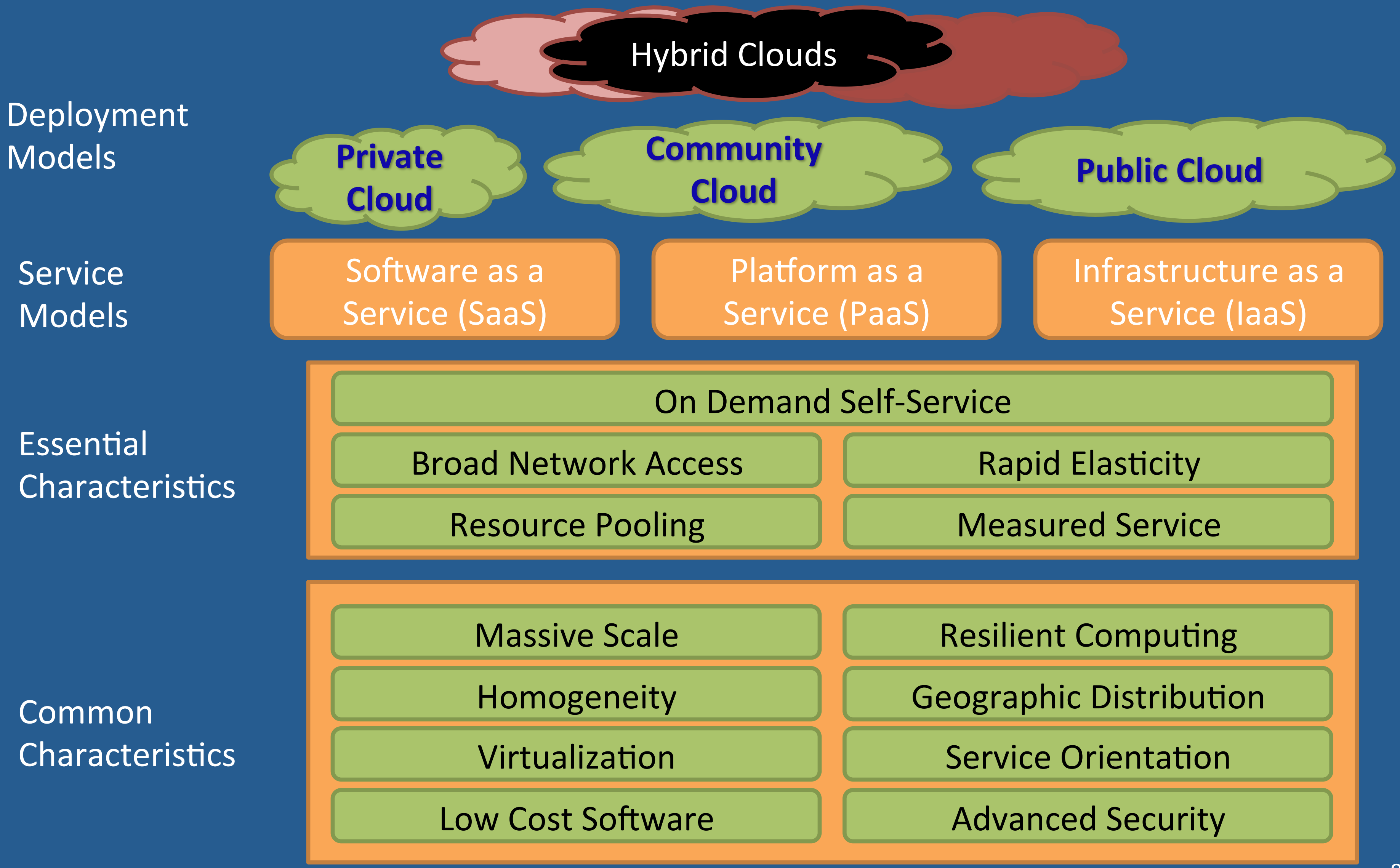

y
A
X = Time
Y = Workload
O
x
y
B
O
x
Private Cloud
Public Cloud
Legend
VM On
VM Off
PM On
PM Off

2 Motivation (*Concepts & Analysis)*

Cloud computing

- This structure describes the most common implementation of cloud; and
- It is based on server virtualization functionalities, where there is a layer that abstracts the physical resources of the servers and presents them as a set of resources to be shared by VMs.

The NIST Cloud Definition Framework



Hybrid Clouds

Deployment Models

- Private Cloud
- Community Cloud
- Public Cloud

Service Models

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

Essential Characteristics

- On Demand Self-Service
- Broad Network Access
- Rapid Elasticity
- Resource Pooling
- Measured Service

Common Characteristics

- Massive Scale
- Resilient Computing
- Homogeneity
- Geographic Distribution
- Virtualization
- Service Orientation
- Low Cost Software
- Advanced Security

Based upon original chart created by Alex Dowbor

2 Motivation (*Concepts & Analysis)*

Green cloud

- The green cloud is not very different from cloud computing, but it infers a concern over the structure and the social responsibility of energy consumption; and
- Hence aiming to ensure the infrastructure sustainability without breaking contracts.

2 Motivation (*Concepts & Analysis)*

Analysis

- Table I relates (1) the 3 possible combinations between VMs and PMs, with (2) the average activation delay, and (3) the chances of the services not being processed (risk); and
- It also presents the energy consumed according to each scenario.

2 Motivation (*Concepts & Analysis)*

PM State	VM State	Time	Risks	Watts	Consumption
Down	Down	30s	High	0Ws	None
Up	Down	10s	Medium	200Ws	Medium
Up	Up	0s	None	215Ws	High

Relation between situations & Risks & Activation delay & Consumption

(ASSUNÇÃO, M. D. et al. ENERGY 2010)

2 Motivation (*Related Works)*

- E. Pinheiro, et al. "Load balancing and unbalancing for power and performance in cluster-based systems" in Proceedings of the Workshop on Compilers and Operating Systems for Low Power. 2001.
- Pinheiro et al. have proposed a technique for managing a cluster of physical machines that minimizes power consumption while maintaining the QoS level.

2 Motivation (*Related Works)*

- The main technique to minimize power consumption is to adjust the load balancing system to consolidate the workload in some resources of the cluster to shut down the idle resources.
- At the end, besides having an economy of 20% compared to fulltime online clusters, it saves less than 6% of the whole consumption of the data center.

2 Motivation (*Related Works)*

- R. N. Calheiros, et al. "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms" Software: Practice and Experience. 2011.
- Calheiros et al. have developed a framework for cloud computing simulation. It has four main features:

2 Motivation (*Related Works)*

- (i) it allows for modeling and instantiation of major cloud computing infrastructures,
- (ii) it offers a platform providing flexibility of service brokers, scheduling and allocations policies,
- (iii) its virtualization engine can be customized, thus providing the capability to simulate heterogeneous clouds, and

2 Motivation (*Related Works)*

- (iv) it is capable of choosing the scheduling strategies for the resources.
- R. Buyya, et al. "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services" Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing. 2010.

2 Motivation (*Related Works)*

- Buyya et al. suggested creating federated clouds, called Interclouds, which form a cloud computing environment to support dynamic expansion or contraction.
- The simulation results revealed that the availability of these federated clouds reduces the average turn-around time by more than 50%.

2 Motivation (*Related Works)*

- It is shown that a significant benefit for the application's performance is obtained by using simple load migration policies.
- R. Buyya, et al. "Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges" in Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications.

2 Motivation (*Related Works)*

- Buyya et al. aimed to create architecture of green cloud. In the proposals some simulations are executed comparing the outcomes of proposed policies, with simulations of DVFS (Dynamic Voltage and Frequency Scaling).
- They leave other possible research directions open, such as optimization problems due to the virtual network topology, increasing response time for the migration of VMs because of the delay between servers or virtual machines when they are not located in the same data centers.

2 Motivation (*Related Works)*

- L. Liu, et al. "Greencloud: a new architecture for green data center" in Proceedings of the 6th international conference industry session on autonomic computing. 2009.
- Liu et al. presented the GreenCloud architecture to reduce data center power consumption while guaranteeing the performance from user perspective.

2 Motivation (*Related Works)*

- P. Mahavadevan, et al. "On Energy Efficiency for Enterprise and Data Center Networks" in IEEE Communications Magazine. 2011.
- Mahadevan et al. described the challenges relating to life cycle energy management of network devices, present a sustainability analysis of these devices, and develop techniques to significantly reduce network operation power.

2 Motivation (*Problem Scenario)*

- To understand the problem scenario, we introduce the elements, interactions, and operation principles in green clouds.
- The target in green clouds is: how to keep resources turned off as long as possible?
- The interactions and operation principles of the scenario are:

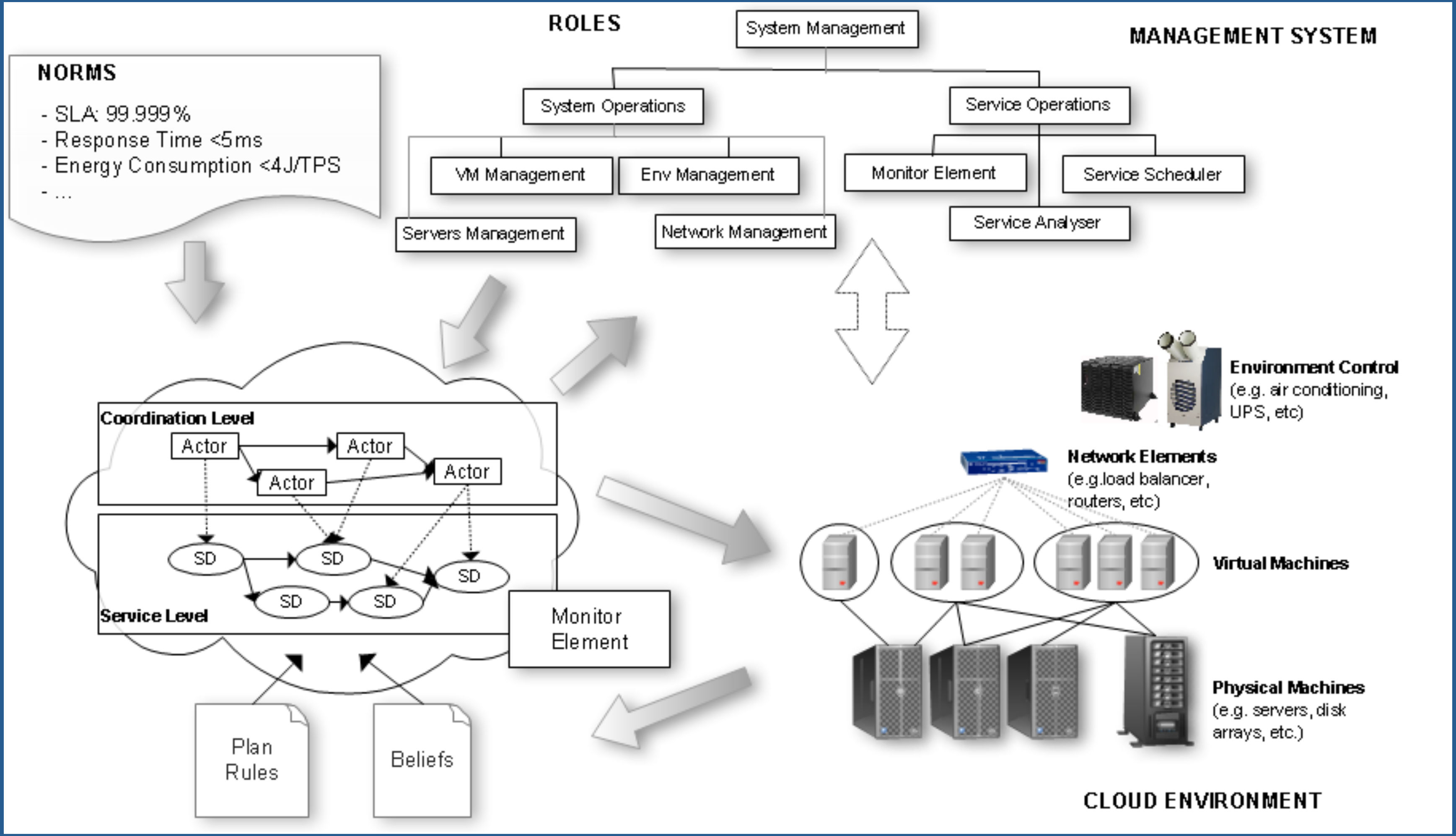
2 Motivation (*Problem Scenario)*

- (i) there are multiple applications generating different load requirements over the day;
- (ii) a load "balance" system distributes the load to active servers in the processing pool;
- (iii) the resources are grouped in clusters that include servers and local environmental control units; and

2 Motivation (*Problem Scenario)*

- (iv) the management system can turn on/off machines overtime, but the question is when to activate resources on-demand?
- In other words, taking too much delay to activate resources in response to a surge of demand (too reactive) may result in the shortage of processing power for a while.

3 Proposals and Solutions



3 Proposals and Solutions

- The four roles that operations system may be classified as are: VM management; Servers management; Network management; and Environment management.
- The three roles that service system may be classified as are: Monitor element; Service scheduler; and Service analyser.

3 Proposals and Solutions

- We can take as example of Planning Rules the following notions:
- (i) if the PM presents a high load, to decrease the load, we will move the VM with more processing to another PM; and
- (ii) if the datacenter presents a high load, to decrease the general load, we will turn on more PMs.

3 Proposals and Solutions

- We can take as example of Beliefs the following notions:
- (i) the activation of a VM type A increases the consumption in B KWh; and
- (ii) the VM type A supports C requests per second.

4 Case Studies

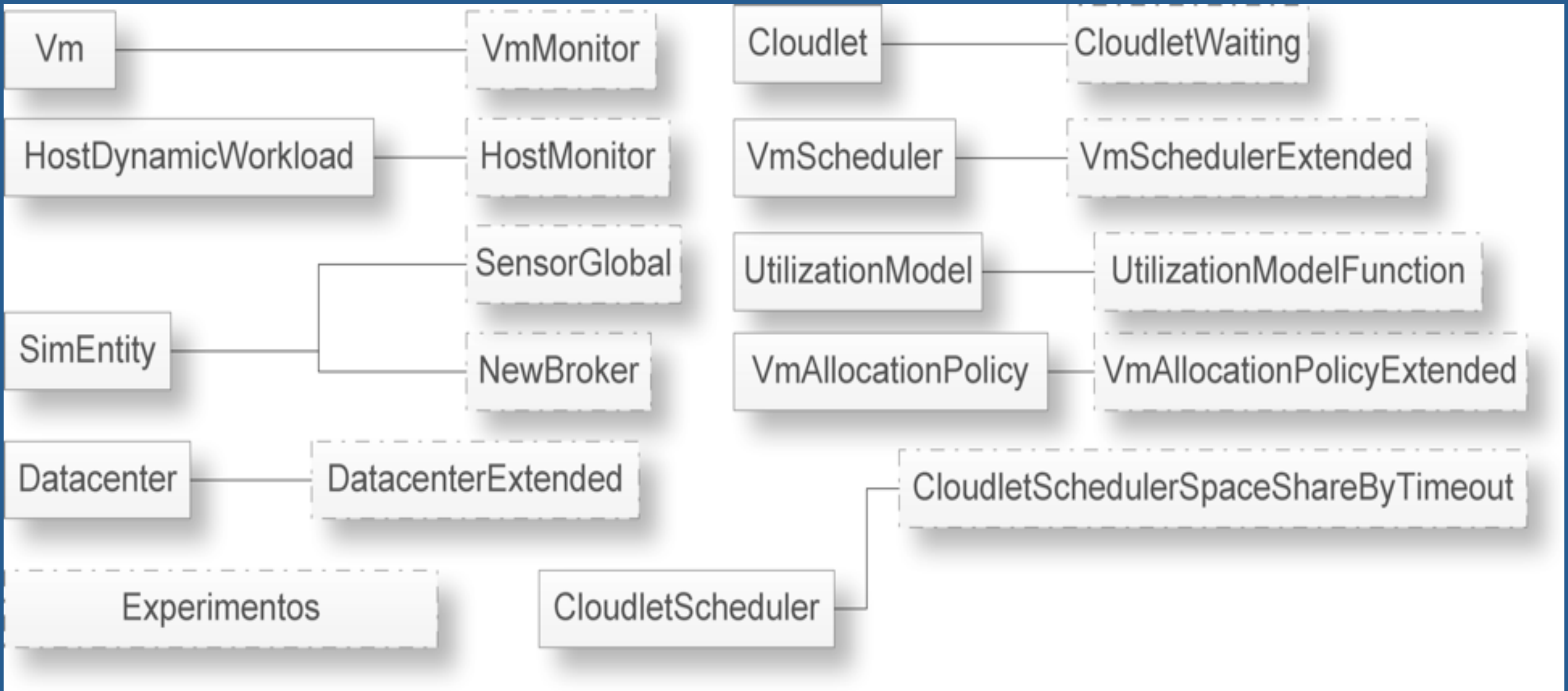
- We modeled the system using Norms (NM), Beliefs (BL) and Plan Rules (PR), inferring that we would need (NM) to reduce energy consumption.
- Based on inferences from NM, BL and PR agents would monitor the system and determine actions dynamically.

5 Results

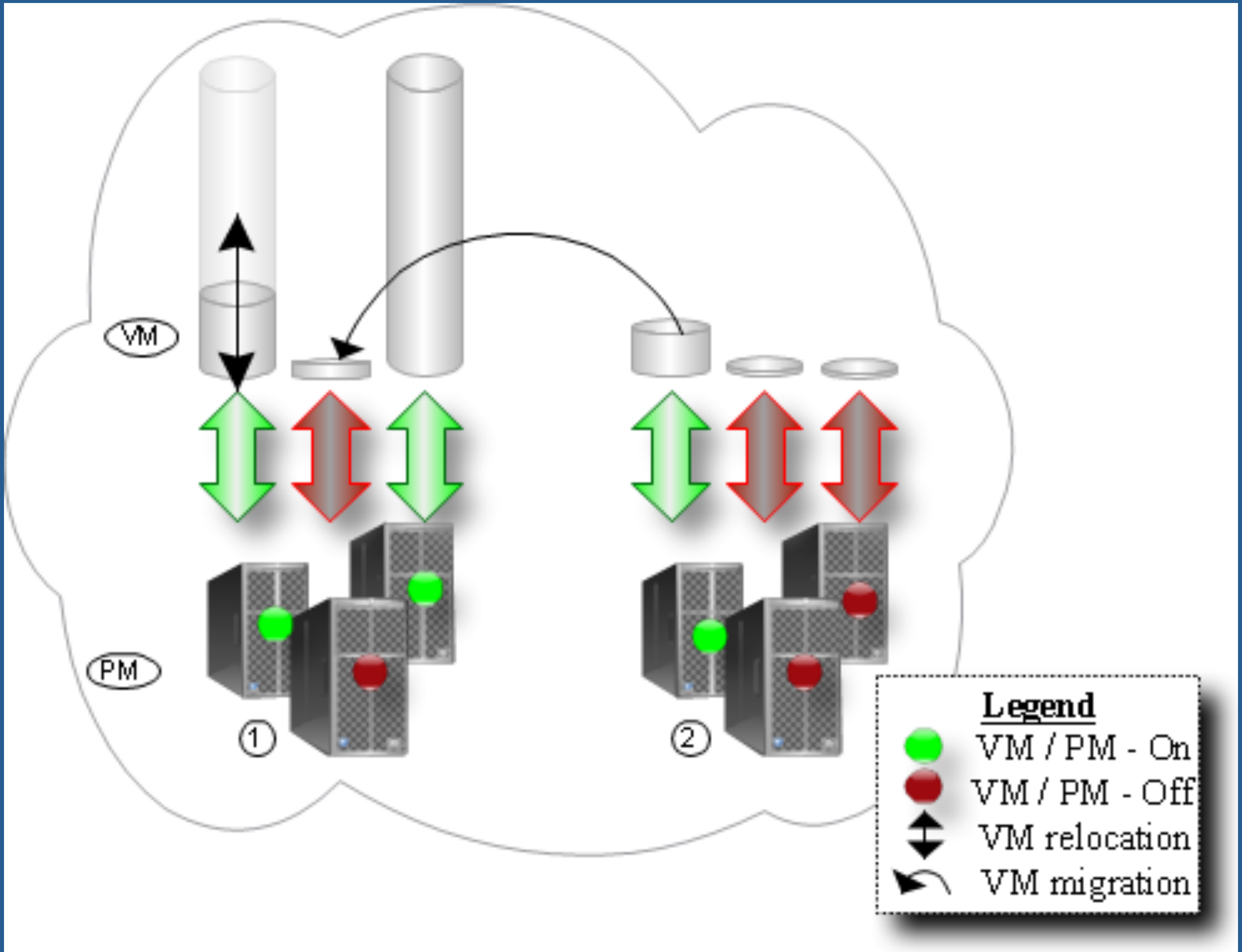
The main components implemented in the improved version at CloudSim are as follows:

HostMonitor: controls the input and output of physical machines; VmMonitor: controls the input and output of virtual machines; NewBroker: controls the size of requests; SensorGlobal: controls the sensors; CloudletSchedulerSpaceShareByTimeout: controls the size and simulation time; VmAllocationPolicyExtended: allocation policy; VmSchedulerExtended: allocates the virtual machines; UtilizationModelFunction: checks the format of requests; CloudletWaiting: controls the time of the request; and DatacenterExtended: controls the datacenter.

5 Results


Vm
VmMonitor
Cloudlet
CloudletWaiting
HostDynamicWorkload
HostMonitor
VmScheduler
VmSchedulerExtended
SensorGlobal
UtilizationModel
UtilizationModelFunction
SimEntity
NewBroker
VmAllocationPolicy
VmAllocationPolicyExtended
Datacenter
DatacenterExtended
CloudletSchedulerSpaceShareByTimeout
Experimentos
CloudletScheduler

5 Results

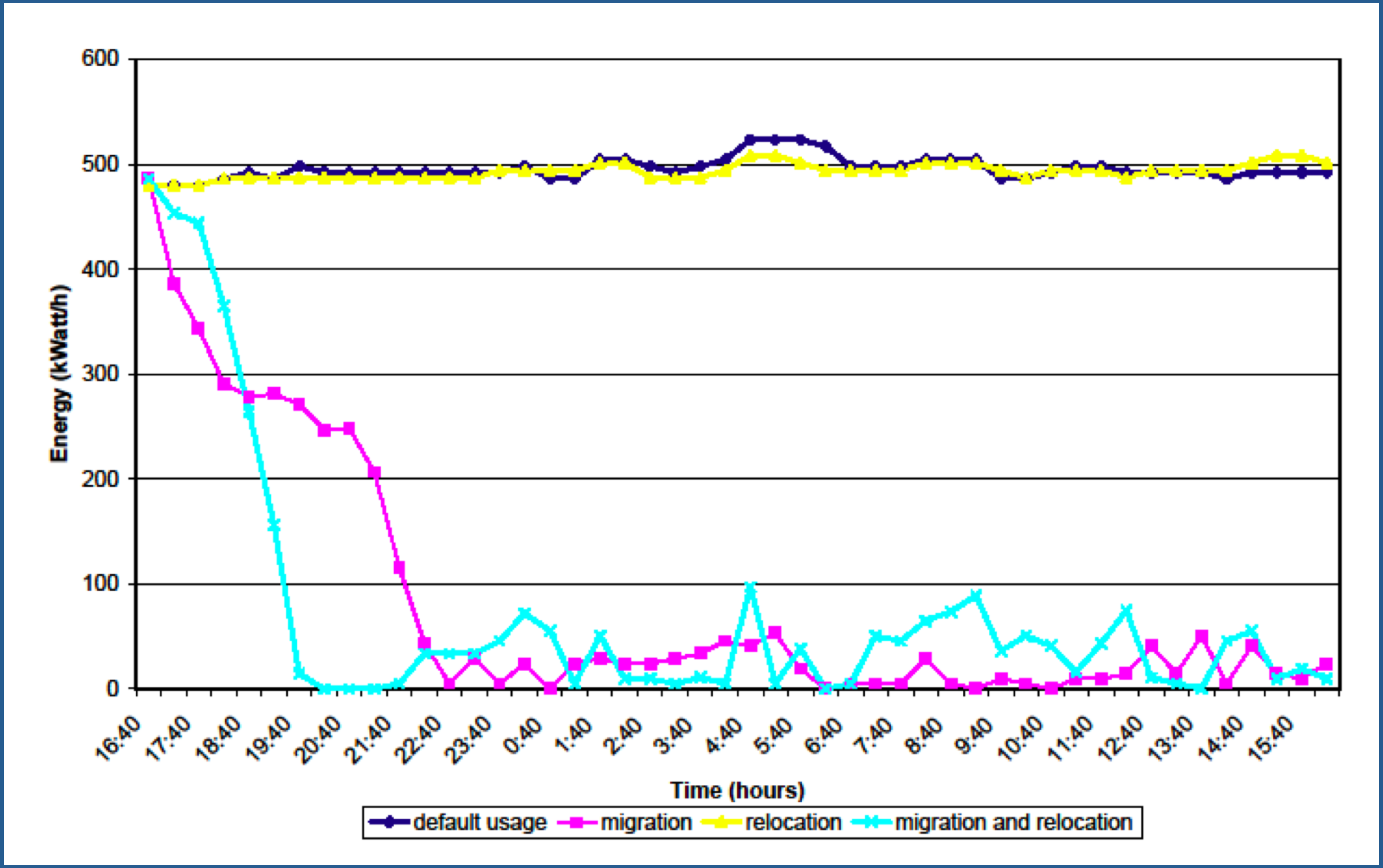


5 Results

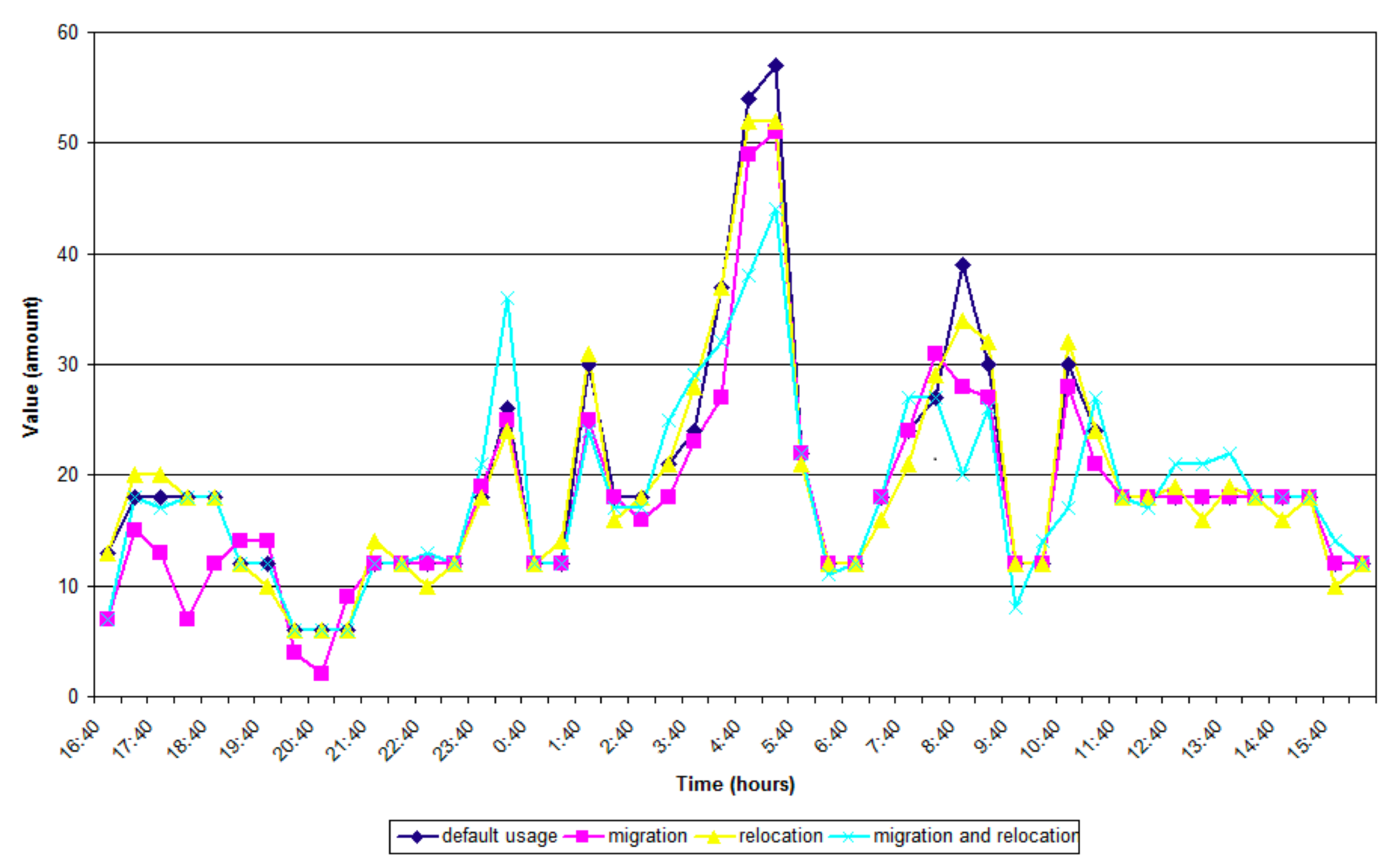
Parameter	Value
VM – Image size	1GB
VM - RAM	256MB
PM - Engine	Xen
PM - RAM	8GB
PM - Frequency	3.0GHZ
PM - Cores	2

PROPOSED SCENARIO CHARACTERISTCS

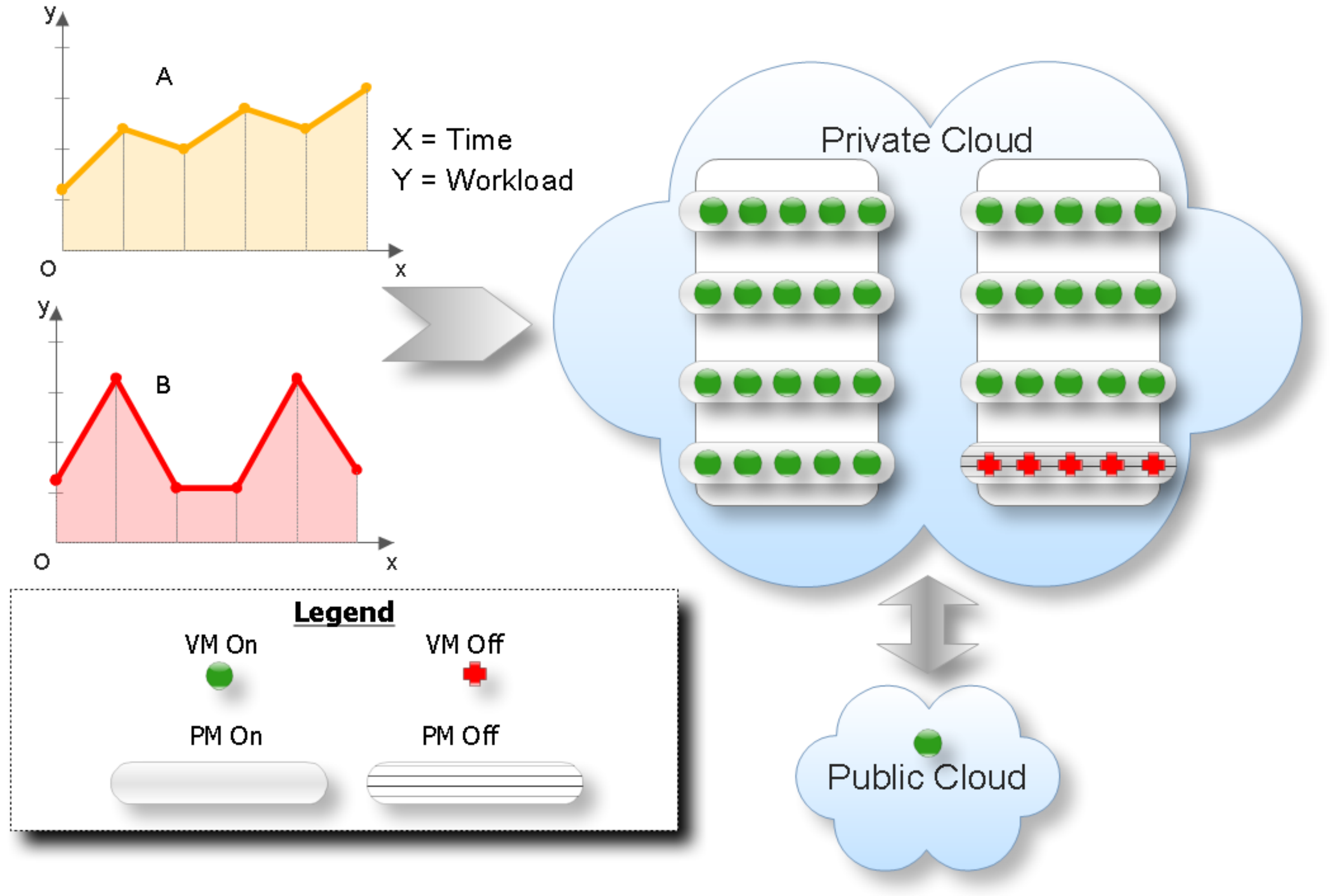
5 Results *(consumption)*



5 Results *(SLA violations)*



5 Results *(Hybrid strategy)*



5 Results *(Hybrid strategy)*

Strategy	Cost	Consumption
On-demand	- 3.2 %	- 23.5 %
Idle resources	- 49.0 %	- 59.0 %

REDUCTION OF COST AND POWER CONSUMPTION

6 Conclusions

- Tests were realized to prove the validity of the system by utilizing the CloudSim simulator from the University of Melbourne in Australia.
- We have implemented improvements related to service-based interaction.
- We implemented migration policies and relocation of virtual machines by monitoring and controlling the system.

6 Conclusions

We achieved the following results in the test environment:

- Dynamic physical orchestration and service orchestration led to 87,18% energy savings, when compared to static approaches; and
- Improvement in load “balancing” and high availability schemas provide up to 8,03% SLA error decrease.

7 Future Works

- As future work we intend to simulate other strategies to get a more accurate feedback of the model, using other simulation environment and testing different approaches of beliefs and plan rules.
- Furthermore, we would like to exploit the integration of other approaches from the field of artificial intelligence, viz. bayesian networks, advanced strategies of intention reconsideration, and improved coordination in multi-agent systems.