

soCloud: distributed multi-cloud platform for deploying, executing and managing distributed applications

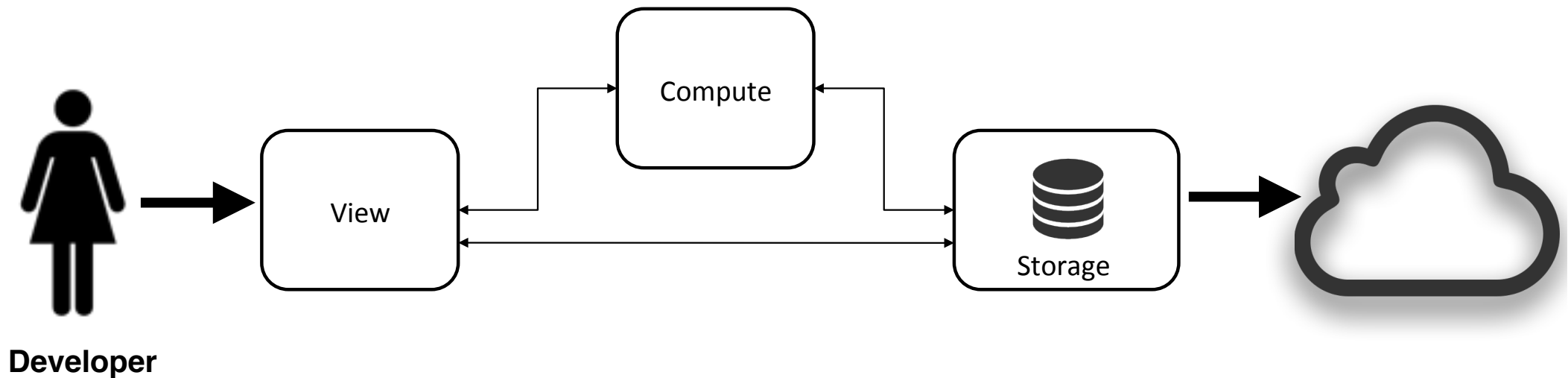
Fawaz PARAISSO
PhD Defense

Advisors: Lionel Seinturier, Philippe Merle

University Lille 1, Inria, SPIRALS research team

Context and motivation

Application fil rouge



Go to the Cloud !

So many problems !

Vendor Lock-in



Geo-location



Cloud-specific services



Failures



solution: Multi-Cloud

Do not put all your eggs in one basket

Multi-Cloud

Why not applying this precept of caution for cloud computing?

What is Multi-Cloud ?

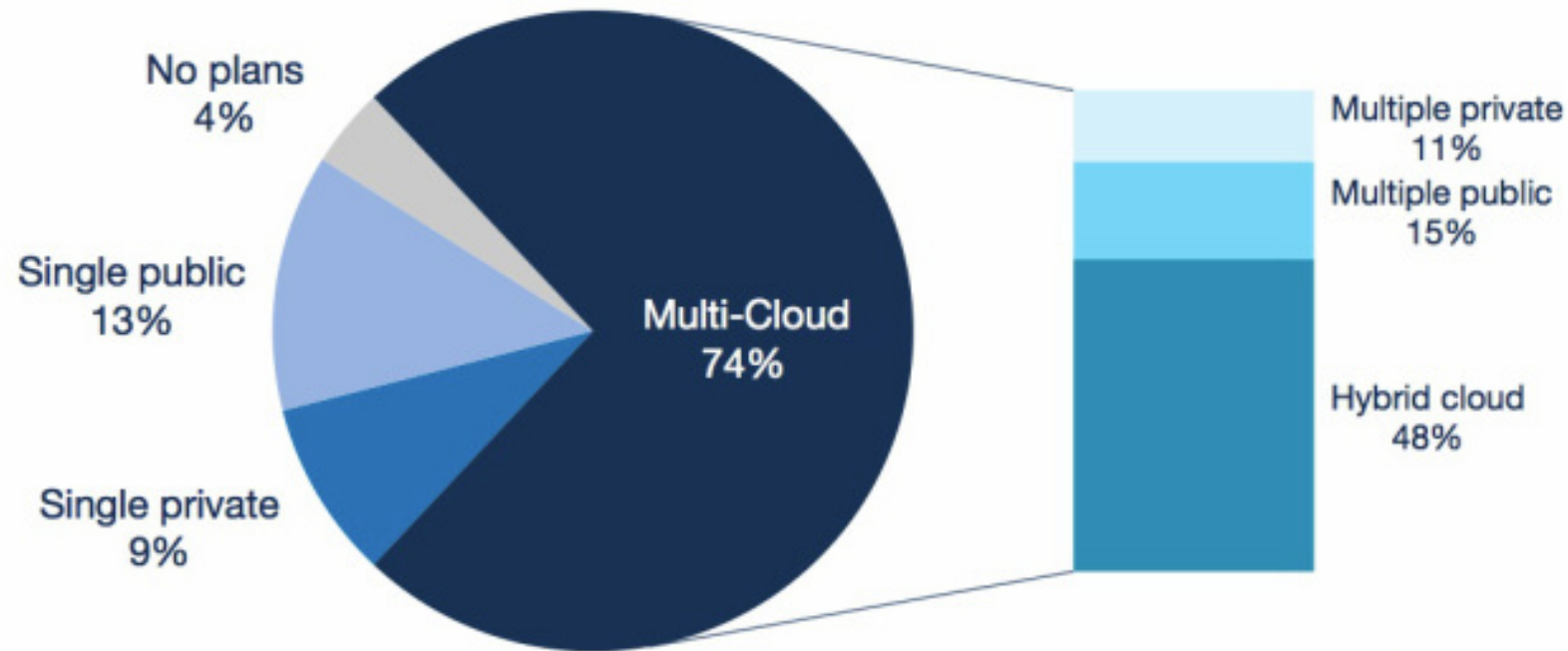
Definition

- **Multi-Cloud Computing**
 - ❖ using multiple cloud providers
 - ❖ independent
 - ❖ no agreement between providers

Context and motivation

Enterprise Cloud Strategy

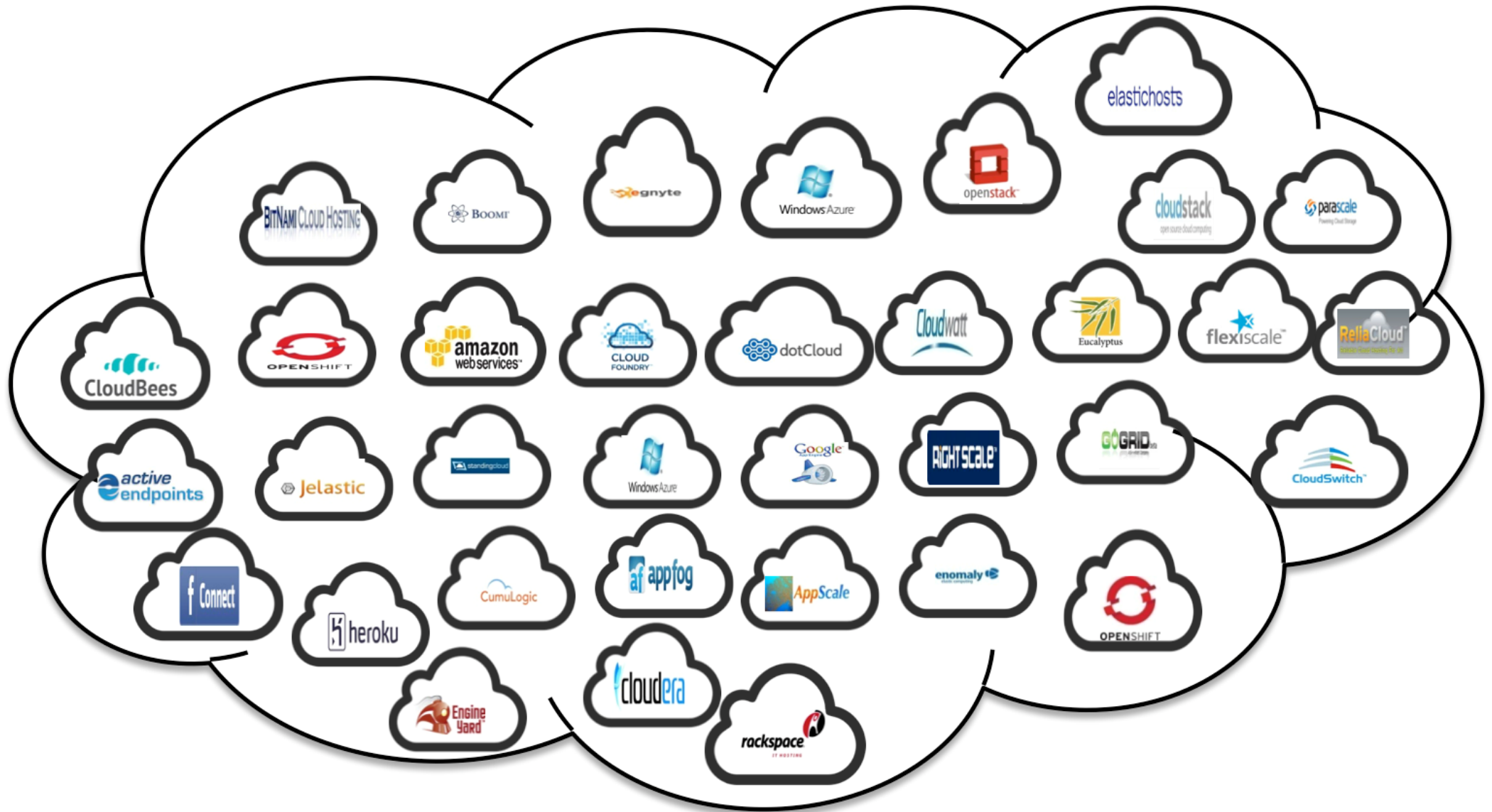
1000+ employees



Source: RightScale 2014 State of the Cloud Report

74% Enterprises have a **Multi-Cloud** strategies

Towards Multi-Cloud Computing



Multi-cloud is supposed to be the solution but...

Outline

1. Context and motivation

2. Challenges

3. State of the art

4. Contributions

4.1. soCloud Model

4.2. soCloud Platform

5. Validation

6. Conclusion & Perspectives

Challenges

Multi-cloud Portability

Multi-cloud Provisioning

Multi-cloud Elasticity

Multi-cloud High-Availability

Challenges

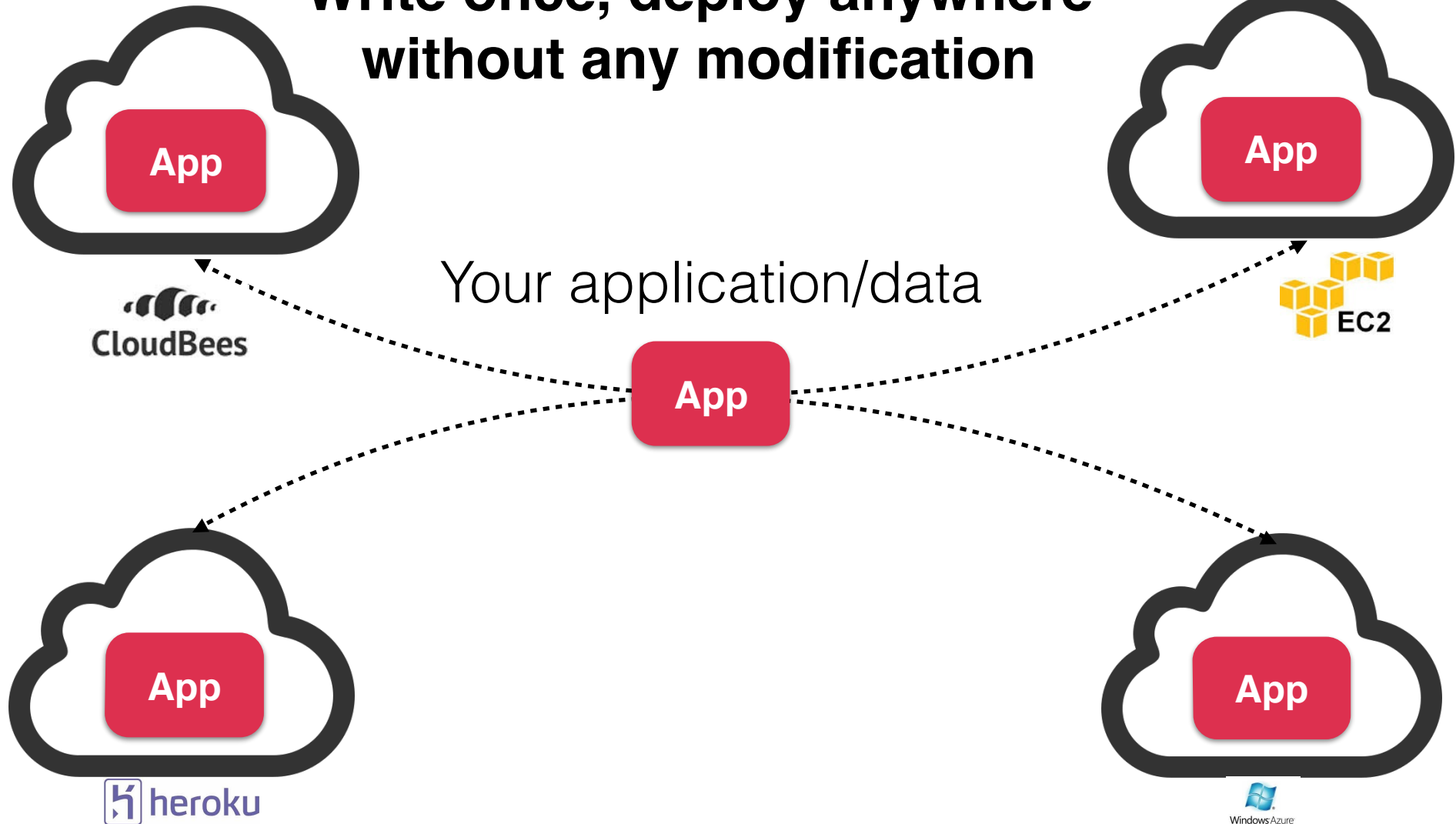
Multi-cloud Portability

Multi-cloud Provisioning

Multi-cloud Elasticity

Multi-cloud High availability

**Write once, deploy anywhere
without any modification**



Challenges

Multi-cloud Portability

Multi-cloud
Provisioning

Multi-cloud Elasticity

Multi-cloud High
availability

Resources

App Deployment

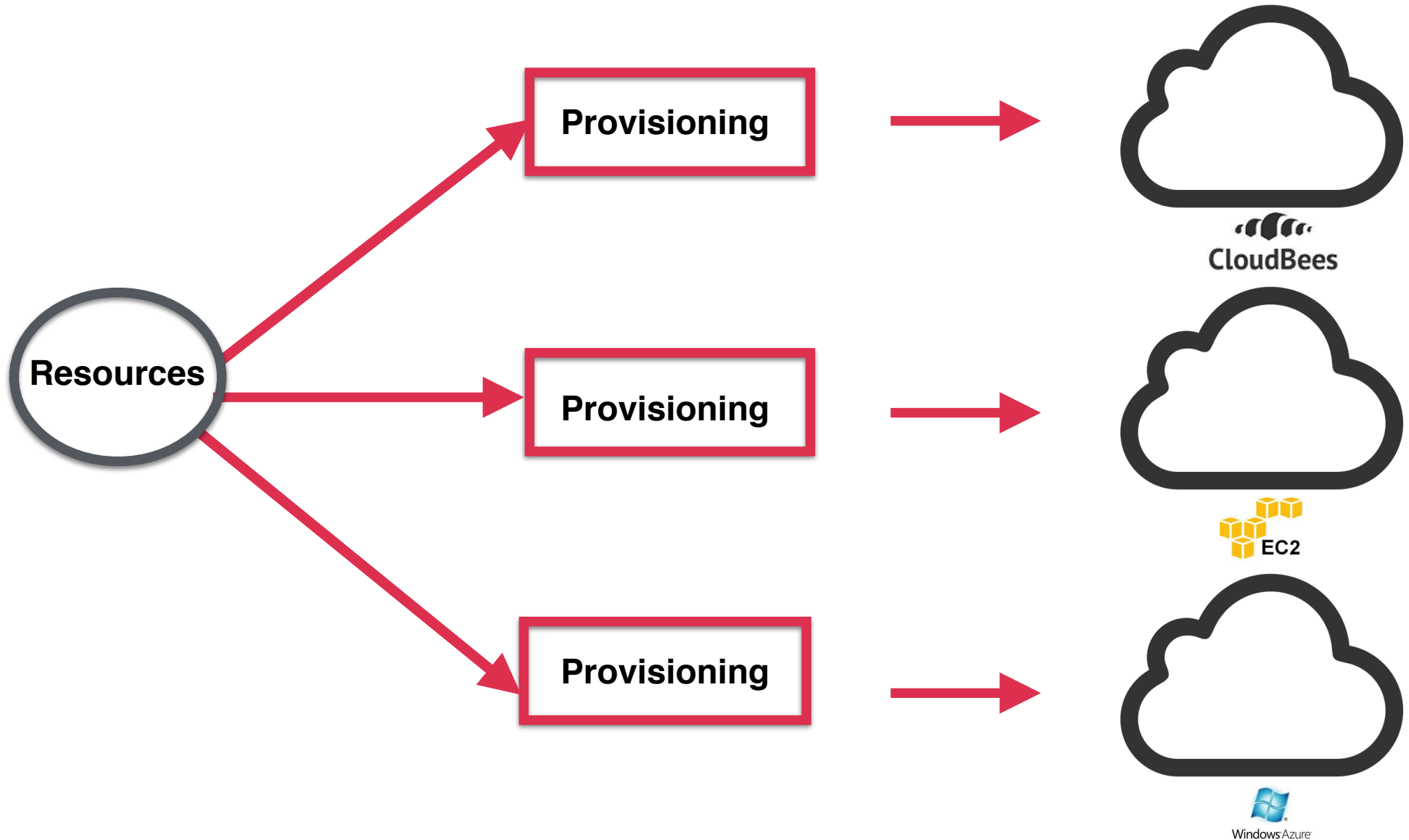
Challenges

Multi-cloud Portability

Multi-cloud Provisioning

Multi-cloud Elasticity

Multi-cloud High availability



Challenges

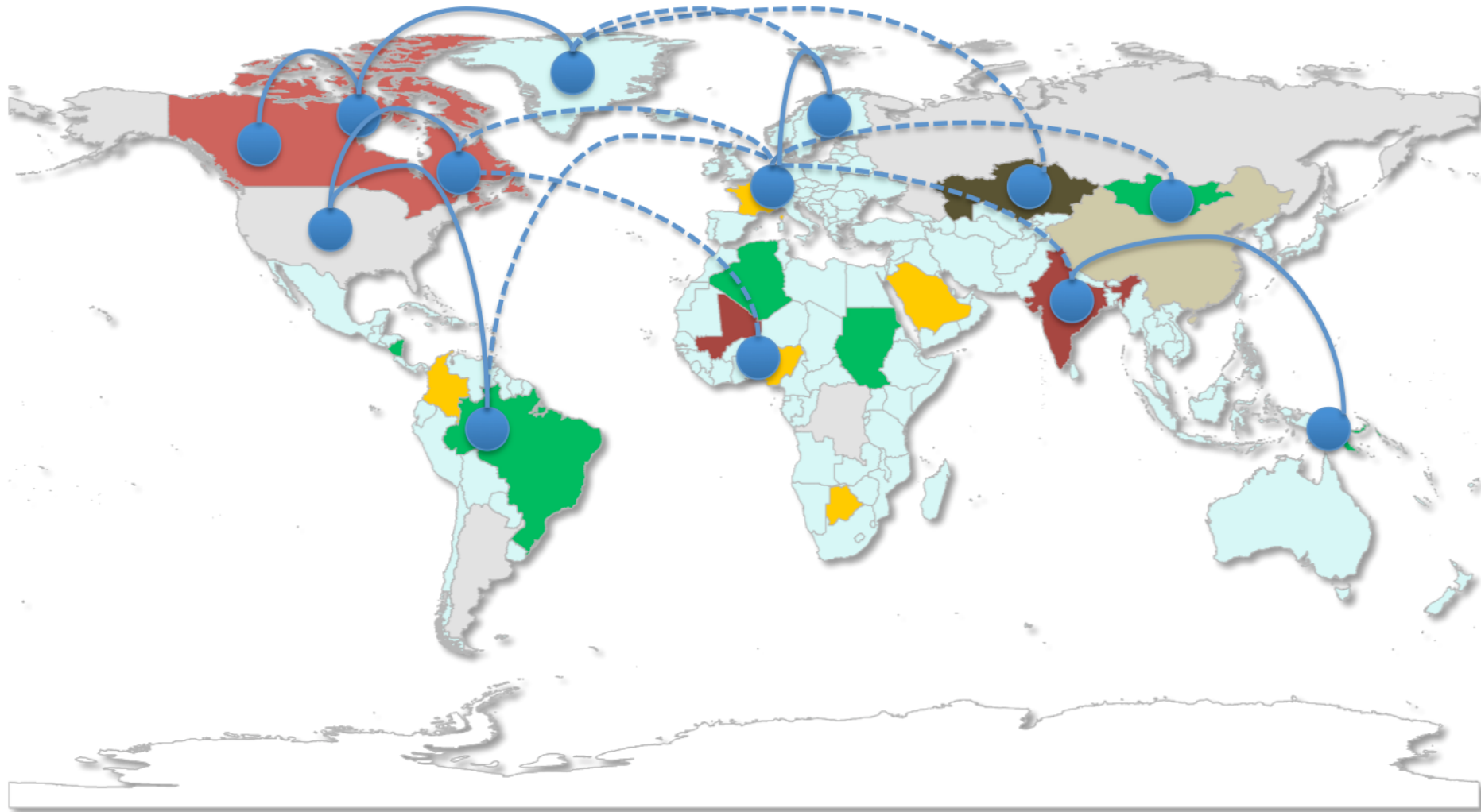
Multi-cloud Portability

Multi-cloud Provisioning

Multi-cloud Elasticity

Multi-cloud High availability

App deployment



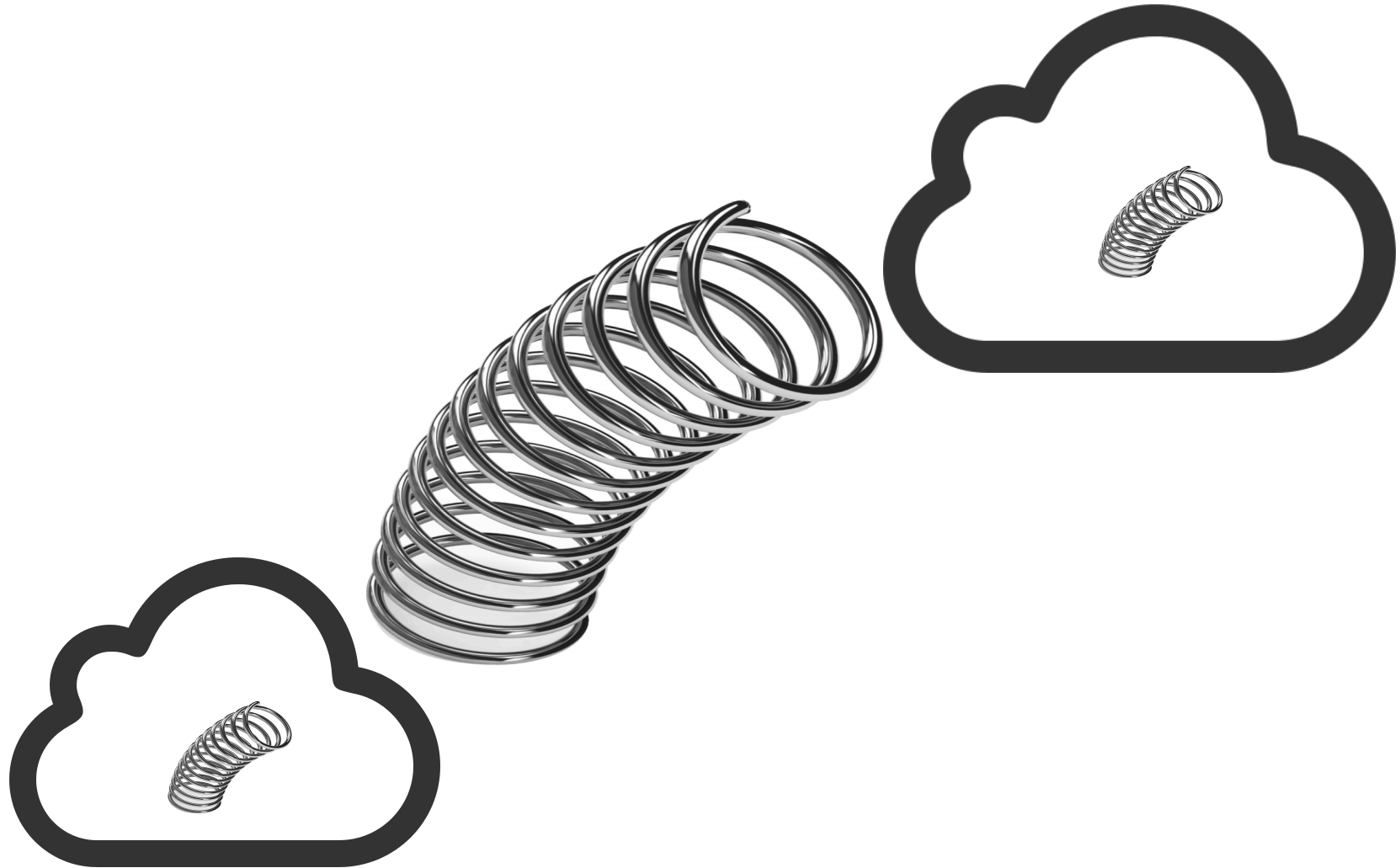
Challenges

Multi-cloud Portability

Multi-cloud Provisioning

Multi-cloud Elasticity

Multi-cloud High availability



Challenges

Multi-cloud Portability

Multi-cloud Provisioning

Multi-cloud Elasticity

Multi-cloud High availability



Windows Azure

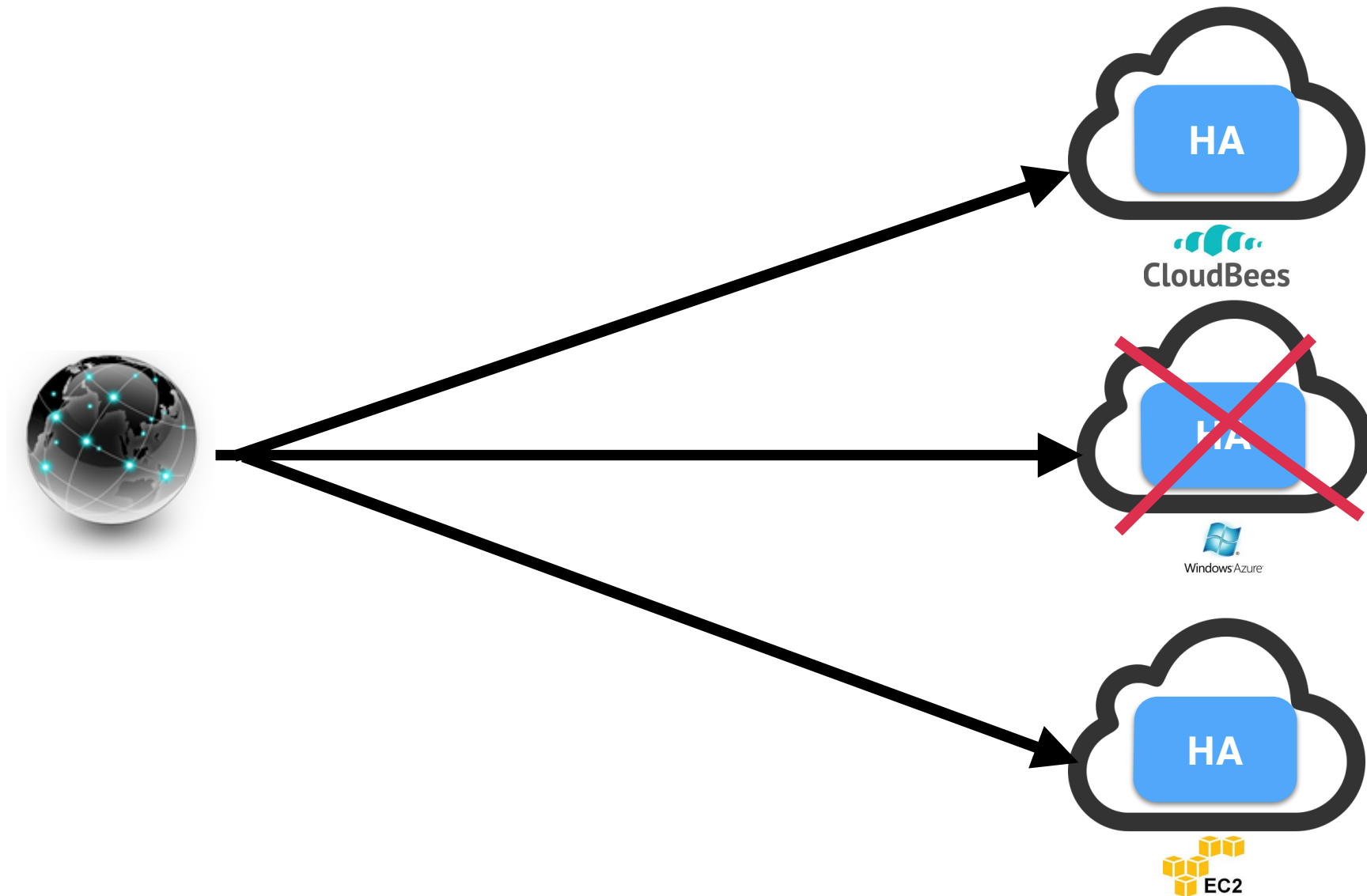
Challenges

Multi-cloud Portability

Multi-cloud Provisioning

Multi-cloud Elasticity

Multi-cloud High availability



Outline

1. Context and motivation

2. Challenges

3. State of the art

4. Contributions

4.1. soCloud Model

4.2. soCloud Platform

5. Validation

6. Conclusion & Perspectives

State of the art

Approach	Layer	Multi-Cloud Portability	Multi-Cloud Provisioning	Multi-Cloud Elasticity	Multi-Cloud High-Availability
moSAIC	PaaS				
STRATOS	IaaS				
MODAClouds	PaaS				
CompatibleOne	IaaS				
Cloud4SOA	PaaS				

State of the art

Approach	Layer	Multi-Cloud Portability	Multi-Cloud Provisioning	Multi-Cloud Elasticity	Multi-Cloud High-Availability
moSAIC	PaaS			+	
STRATOS	IaaS			+	
MODAClouds	PaaS			-	
CompatibleOne	IaaS			-	
Cloud4SOA	PaaS			-	

State of the art

Approach	Layer	Multi-Cloud Portability	Multi-Cloud Provisioning	Multi-Cloud Elasticity	Multi-Cloud High-Availability
moSAIC	PaaS			+	
STRATOS	IaaS			+	
MODAClouds	PaaS			-	
CompatibleOne	IaaS			-	
Cloud4SOA	PaaS			-	
soCloud	PaaS			++	

State of the art

Approach	Layer	Multi-Cloud Portability	Multi-Cloud Provisioning	Multi-Cloud Elasticity	Multi-Cloud High-Availability
moSAIC	PaaS	+	+	+	-
STRATOS	IaaS	-	+	+	-
MODAClouds	PaaS	+	-	-	+
CompatibleOne	IaaS	+	+	-	+
Cloud4SOA	PaaS	+	+	-	-
soCloud	PaaS	++	++	++	++

Outline

1. Context and motivation

2. Challenges

3. State of the art

4. Contributions

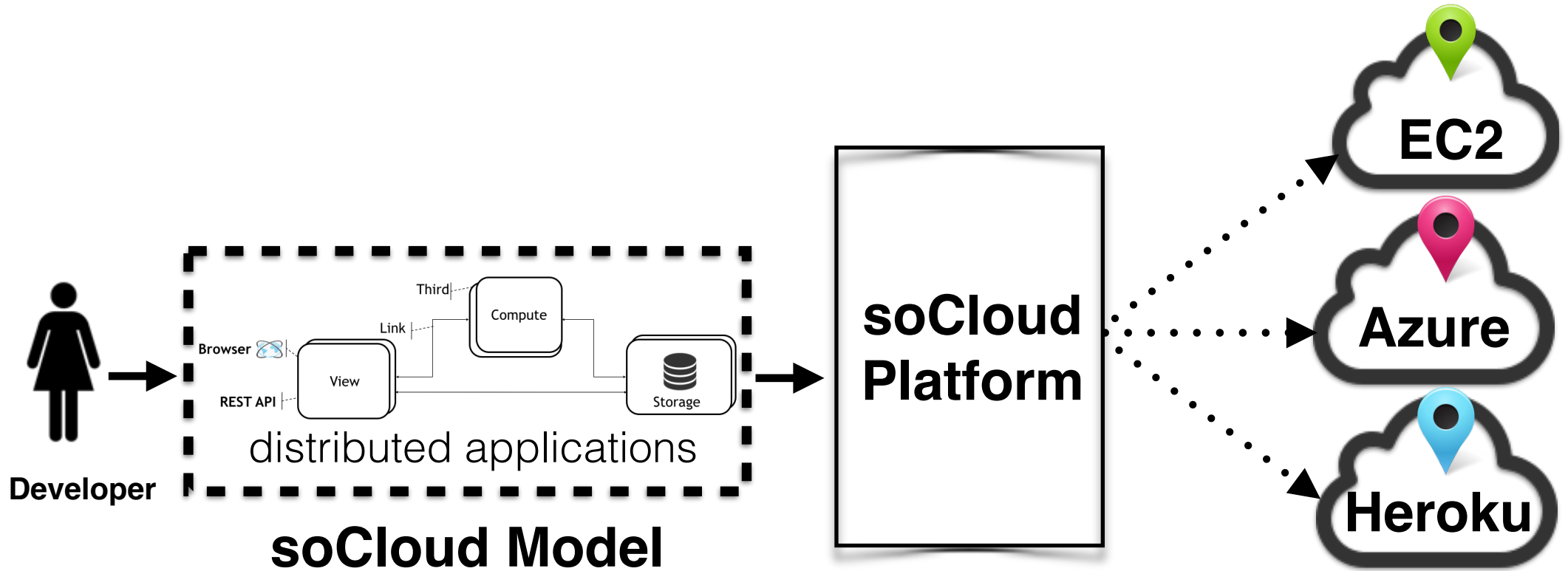
4.1 .soCloud Model

4.2 .soCloud Platform

5. Validation

6. Conclusion & Perspectives

soCloud Overview



Outline

1.Context and motivation

2.Challenges

3.State of the art

4.Contributions

4.1.soCloud Model

4.2.soCloud Platform

5.Validation

6.Conclusion & Perspectives

Objectives

“Provides a **model** to design a distributed applications in a simple and concise manner for a Multi-Cloud environment”

Features

Identify requirements for engineering distributed application for the Multi-Cloud environments

**Multi-Cloud
Portability**

**Abstraction
Standard
Structure**

**Multi-Cloud
Provisioning**

**Placement
Resources
Granularity**

**Multi-Cloud
Elasticity**

**DSL
Simple**

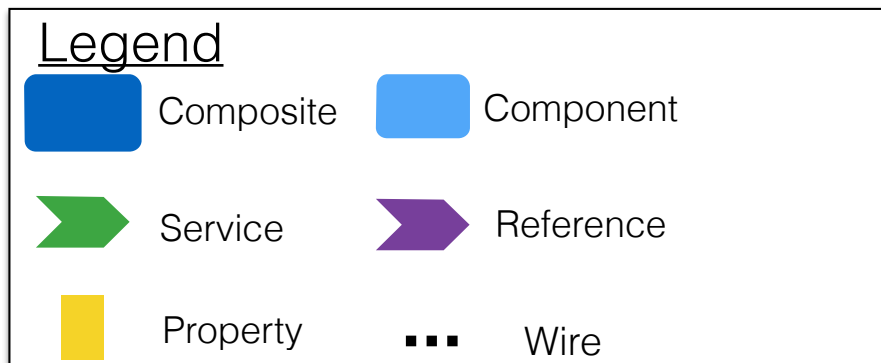
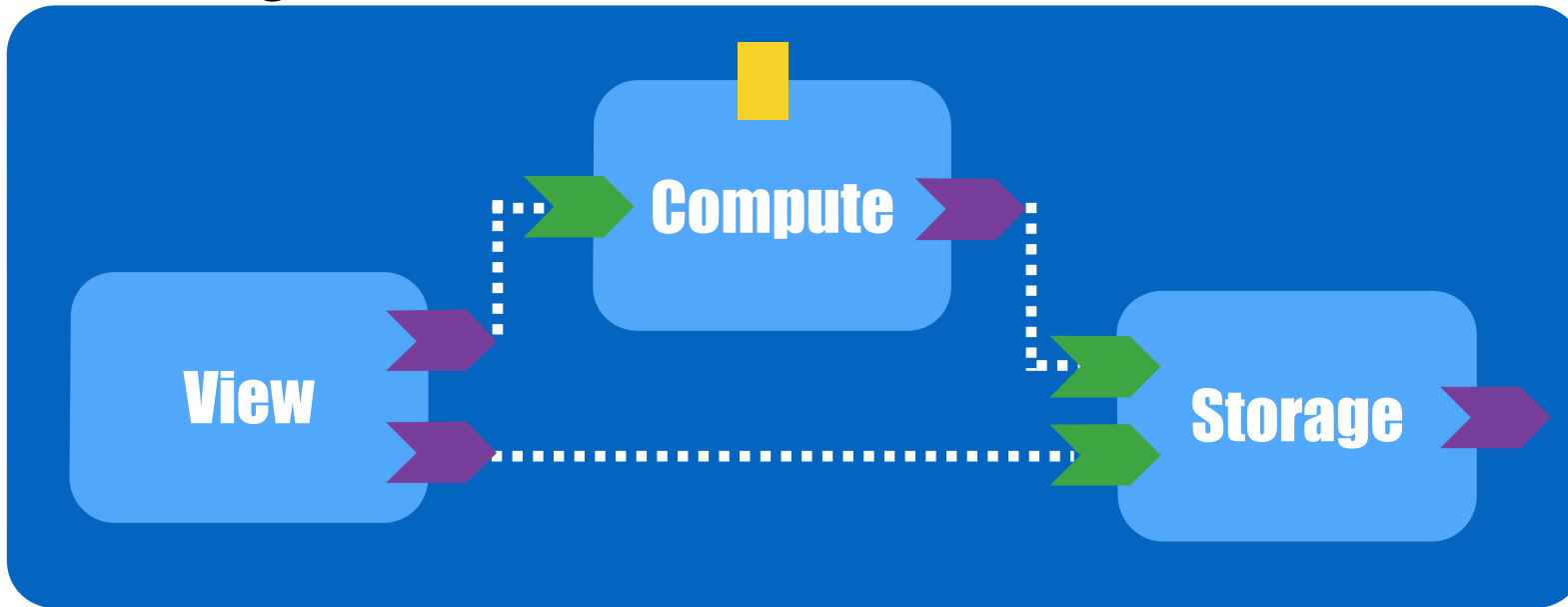
**Multi-Cloud
High-availability**

**Failures
Diversity**

Extended SCA Model

SCA

- **S**ervice **C**omponent **A**rchitecture (**SCA**)
 - ❖ Set of OASIS specifications
 - ❖ Distributed applications
 - ❖ Using SOA



SCA Contribution



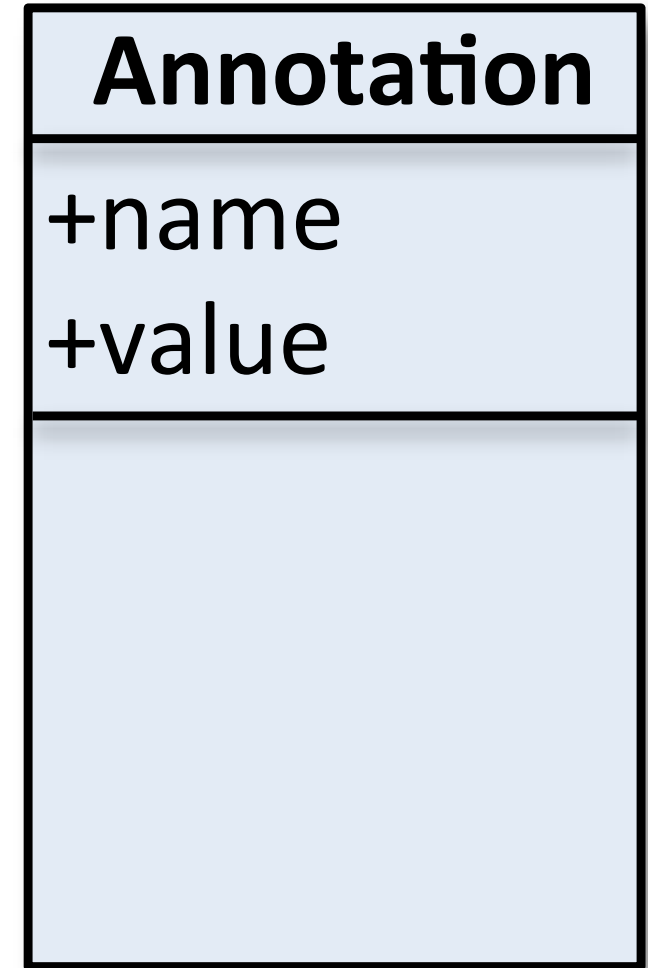
ZIP File

soCloud Model based on SCA

Why **extend** SCA model ?



Extension 1



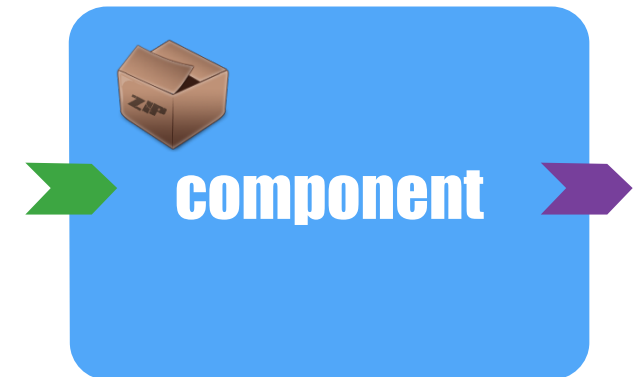
Extension 2

soCloud Model: implementation

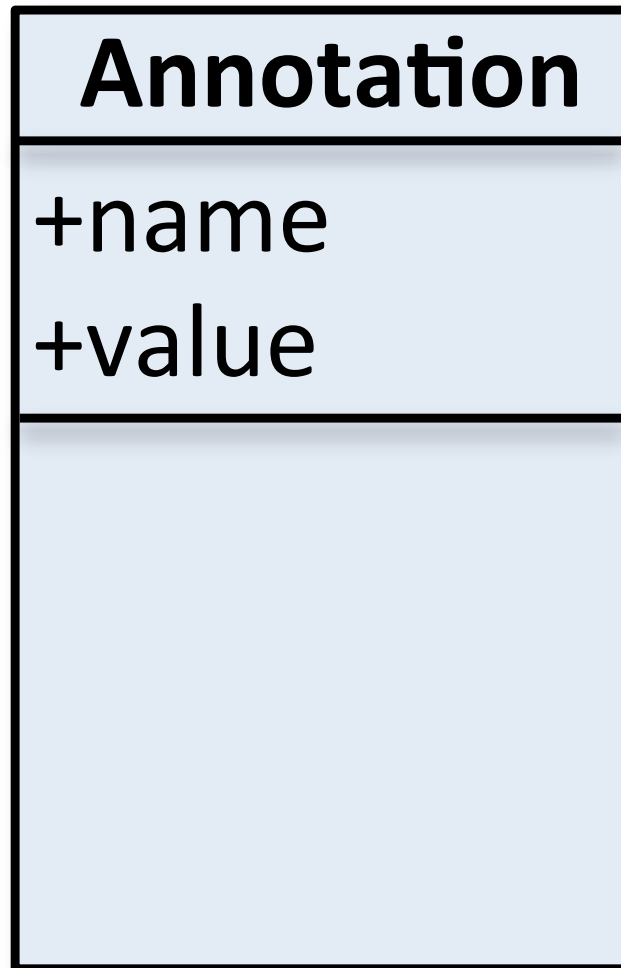


Extension 1

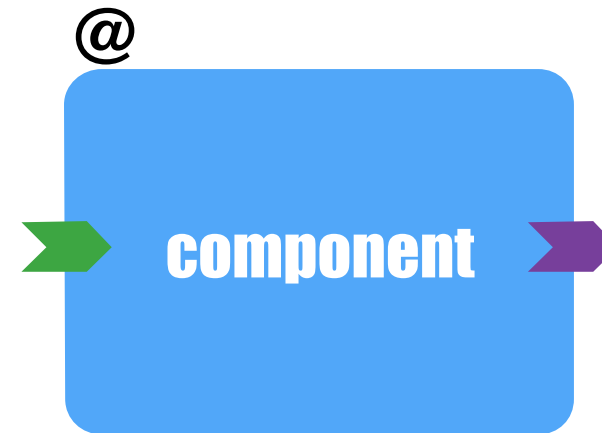
- Provides **high level conceptual** view to a component
- Allows the **deployment** of the component as execution unit
- **Structured** components of distributed applications



soCloud Model: annotations

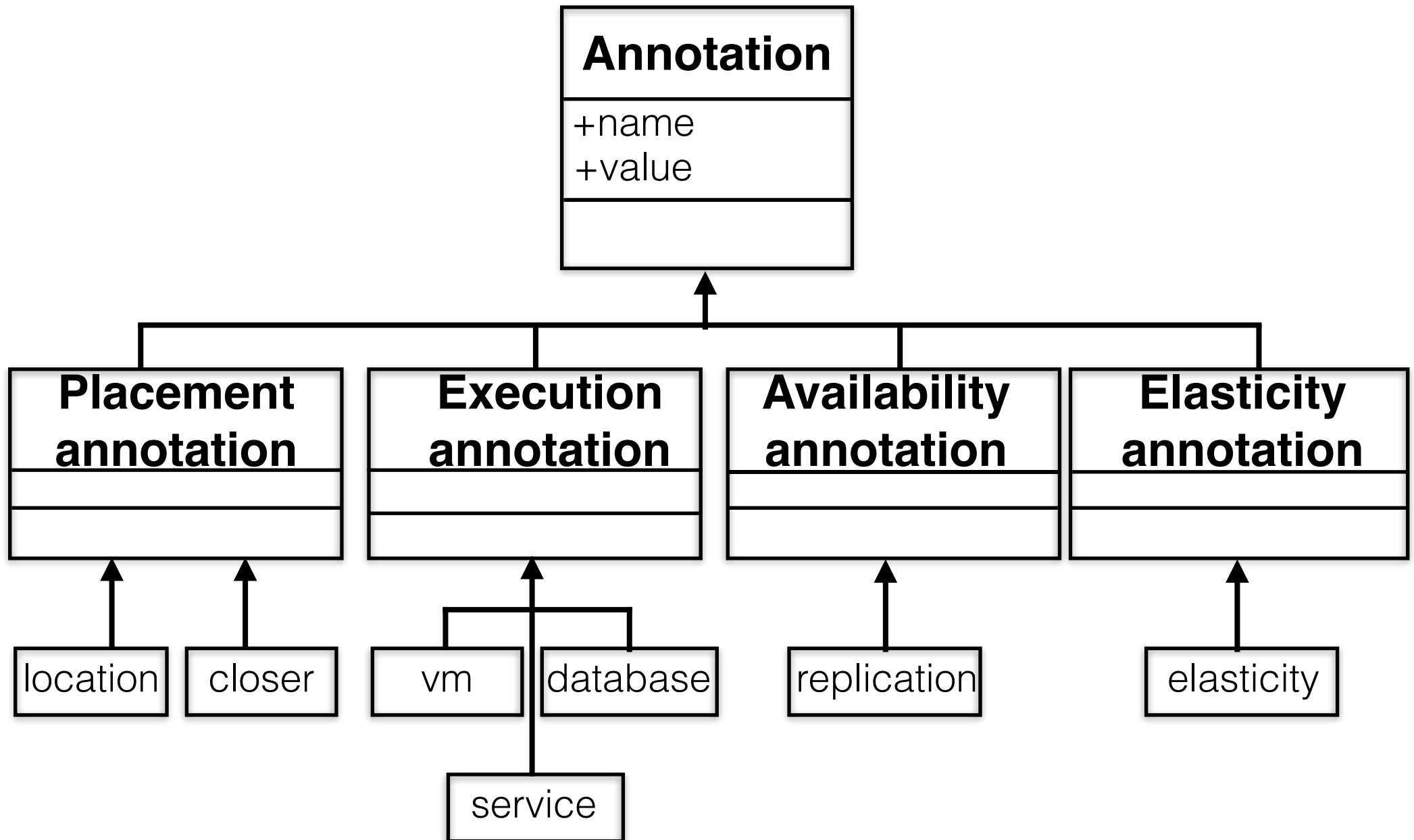


Extension 2

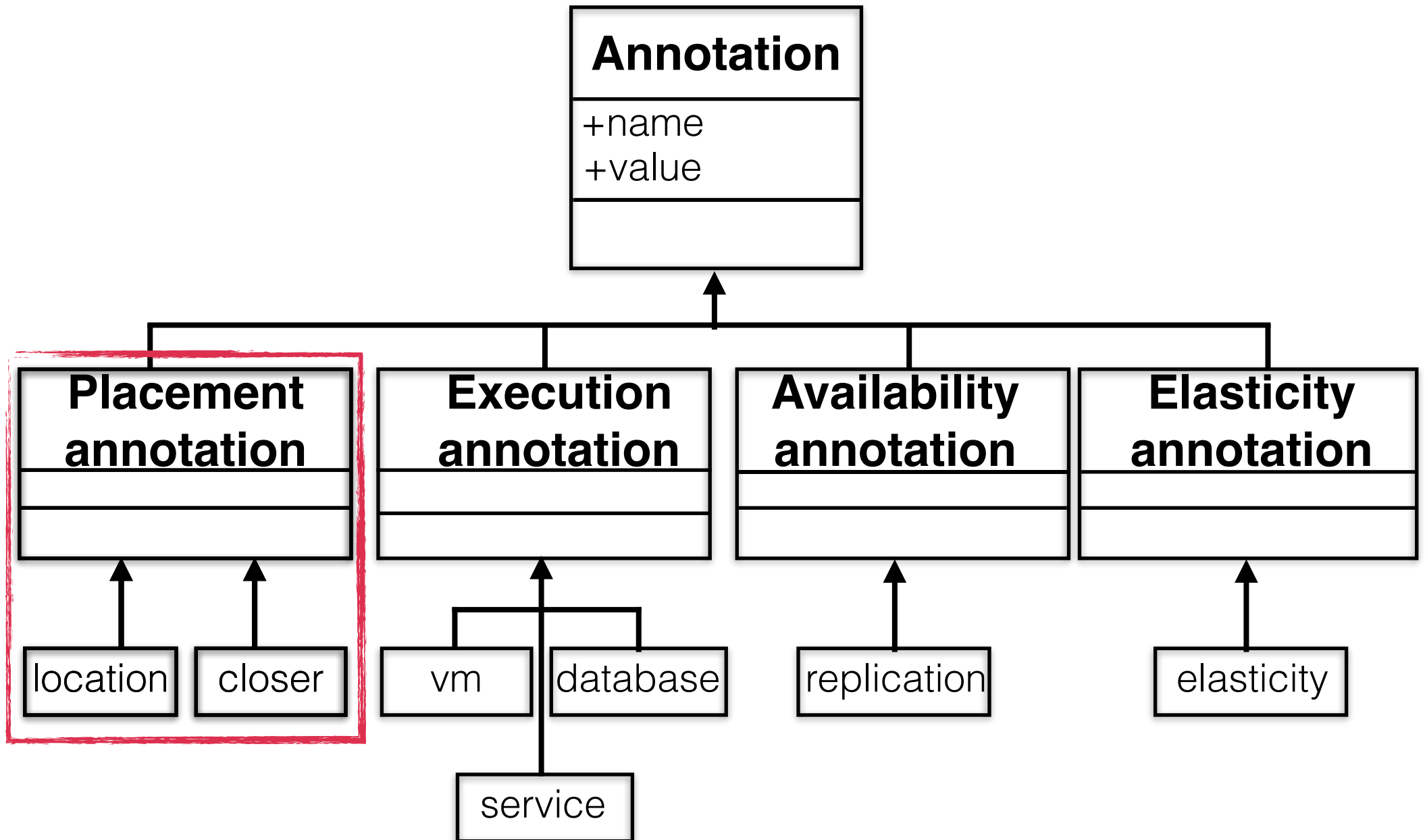


- Allowing to associate **non-functional** requirements to a **component**
- The **SCA model** does not allow us to take into account these non-functional requirements

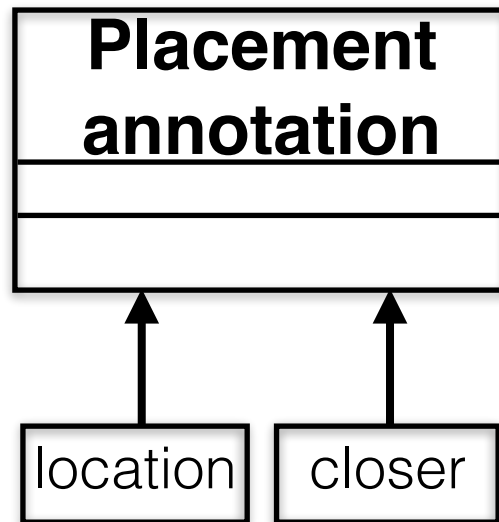
soCloud Model: annotations



soCloud Model: annotations



soCloud Model: annotations

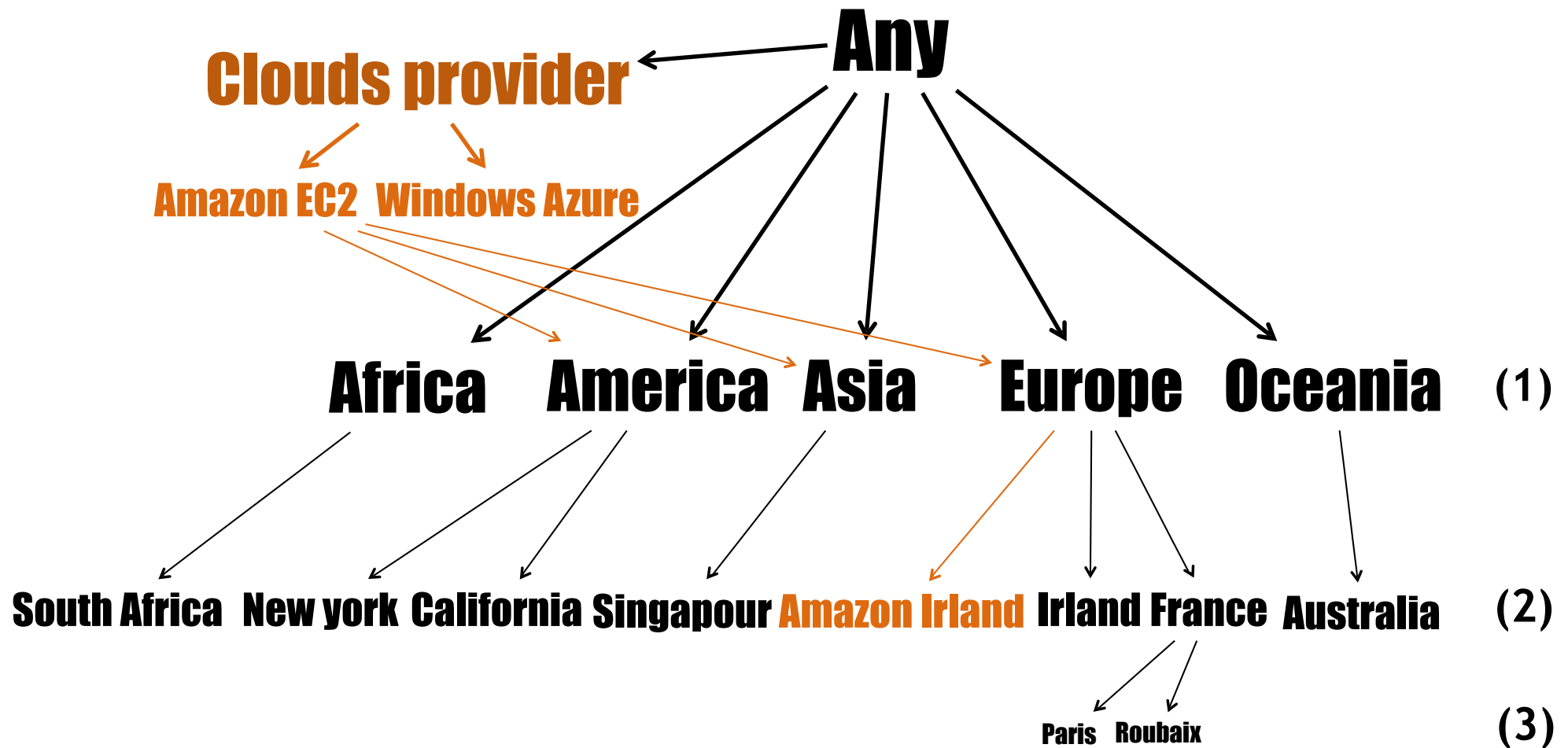


1 @location = 'value'

2 @closer = 'value'

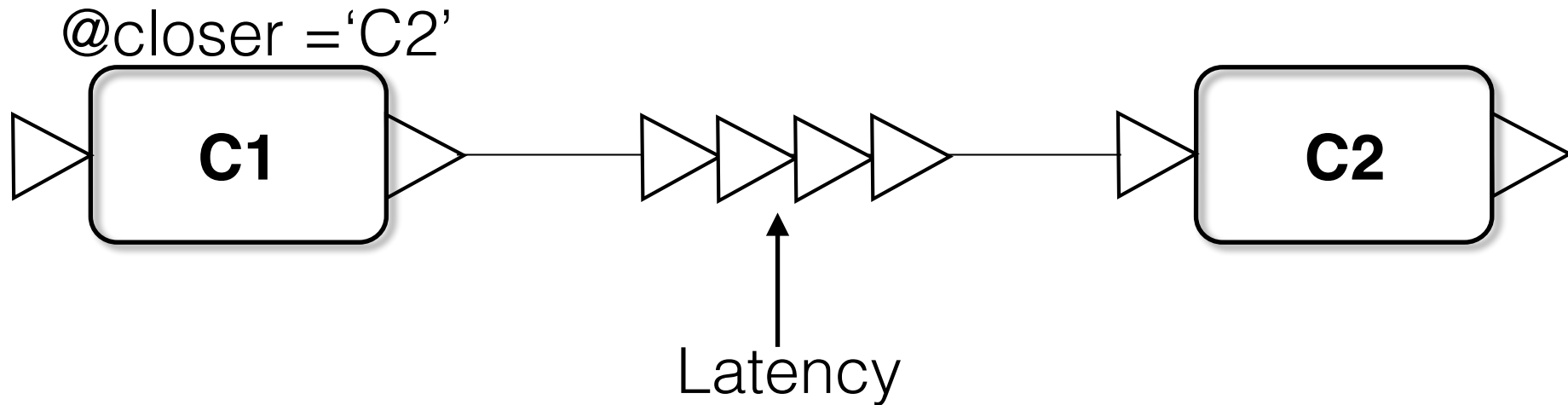
soCloud Model: annotations

@location = 'value'

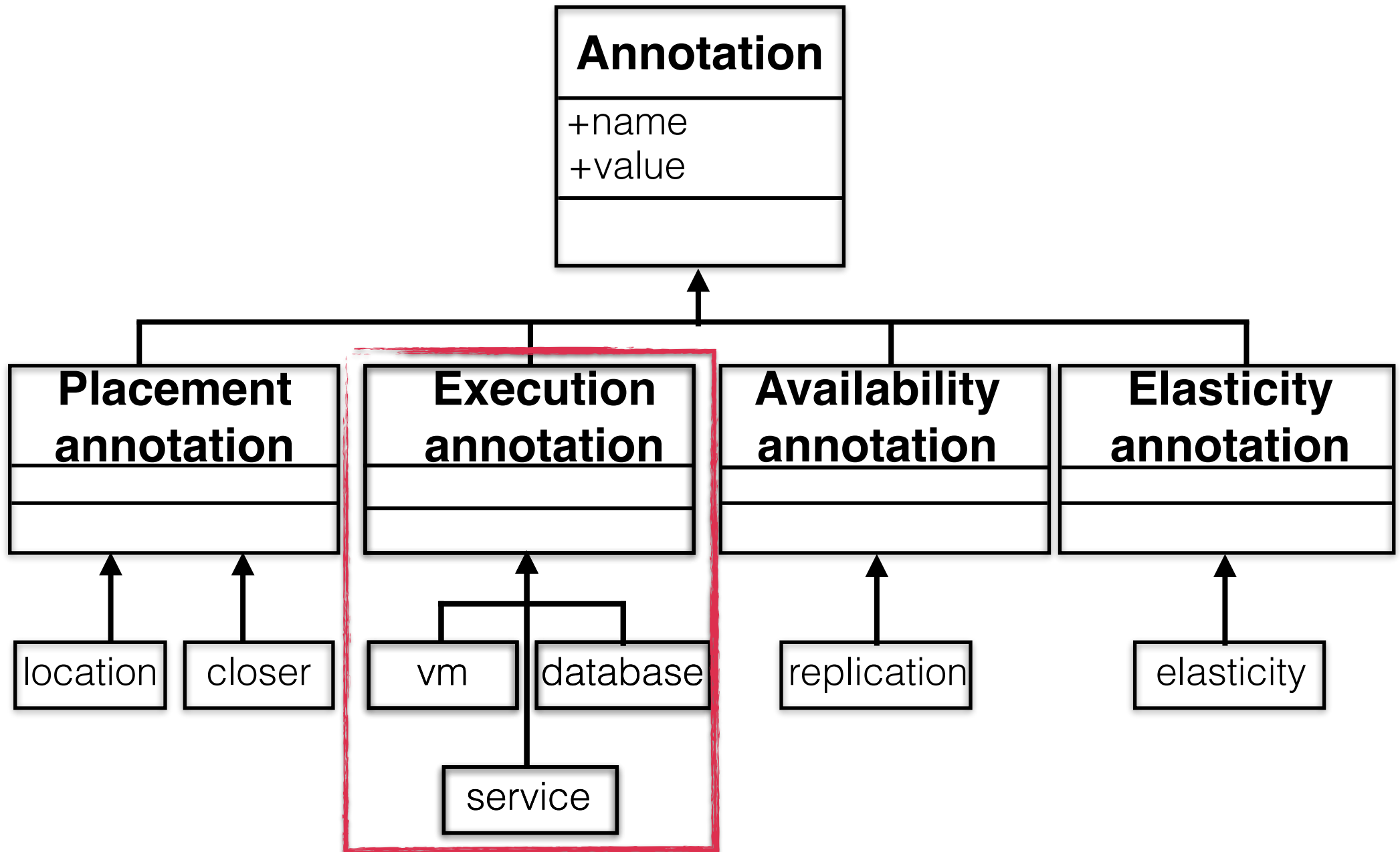


soCloud Model: annotations

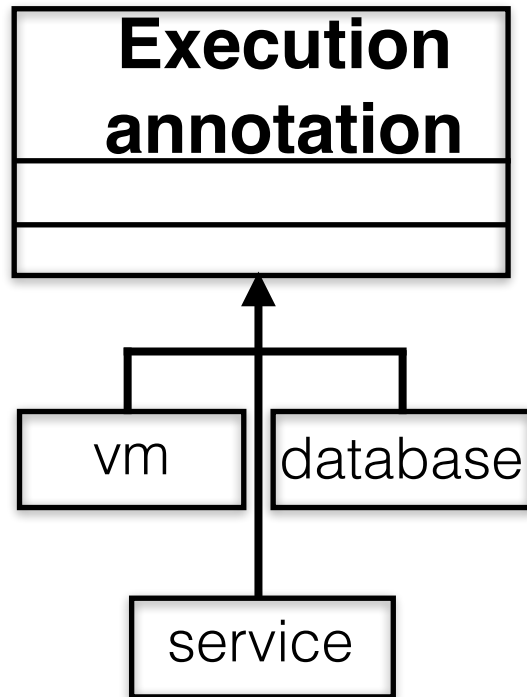
2 @closer = 'value'



soCloud Model: annotations



soCloud Model: annotations



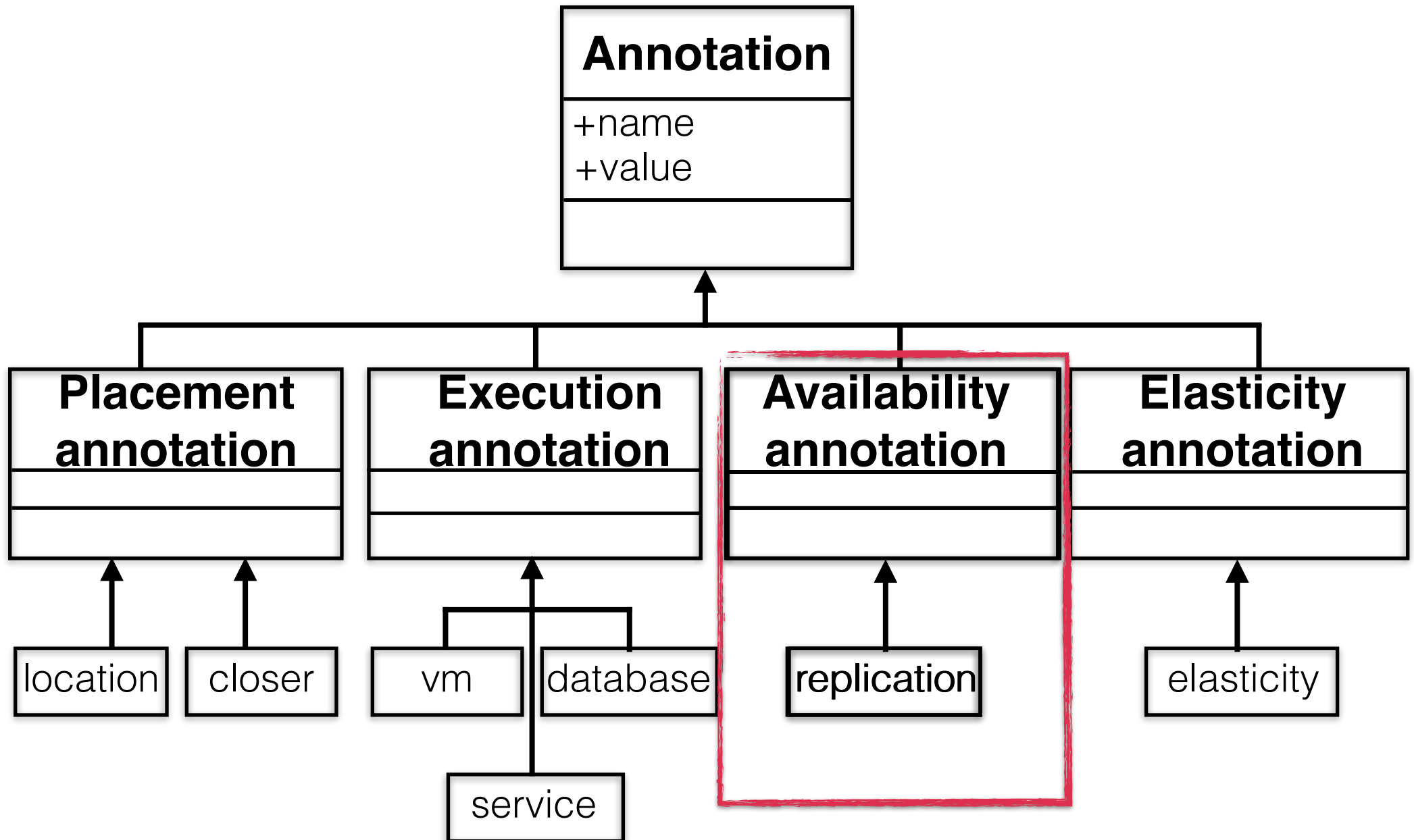
- 1 @vm = 'type_vm'
- 2 @database = 'name -> version'
- 3 @service = 'name -> version'

Optional ←

Example

- 1 @vm = 'micro'
- 2 @database = 'MySQL'
- 3 @service = 'IronMQ -> 2.8.9'

soCloud Model: annotations

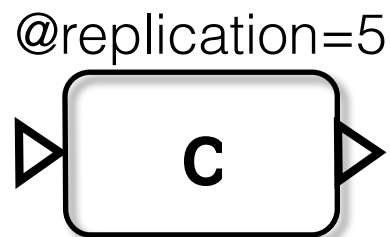


soCloud Model: annotations

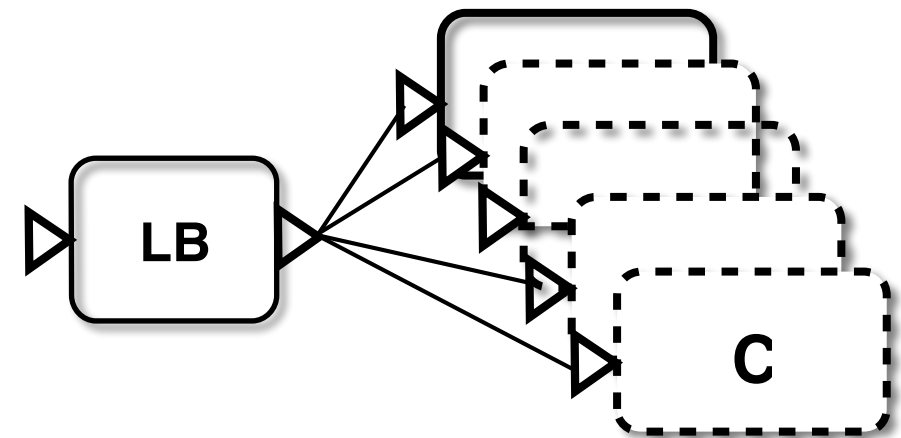
1 @replication = 'number'



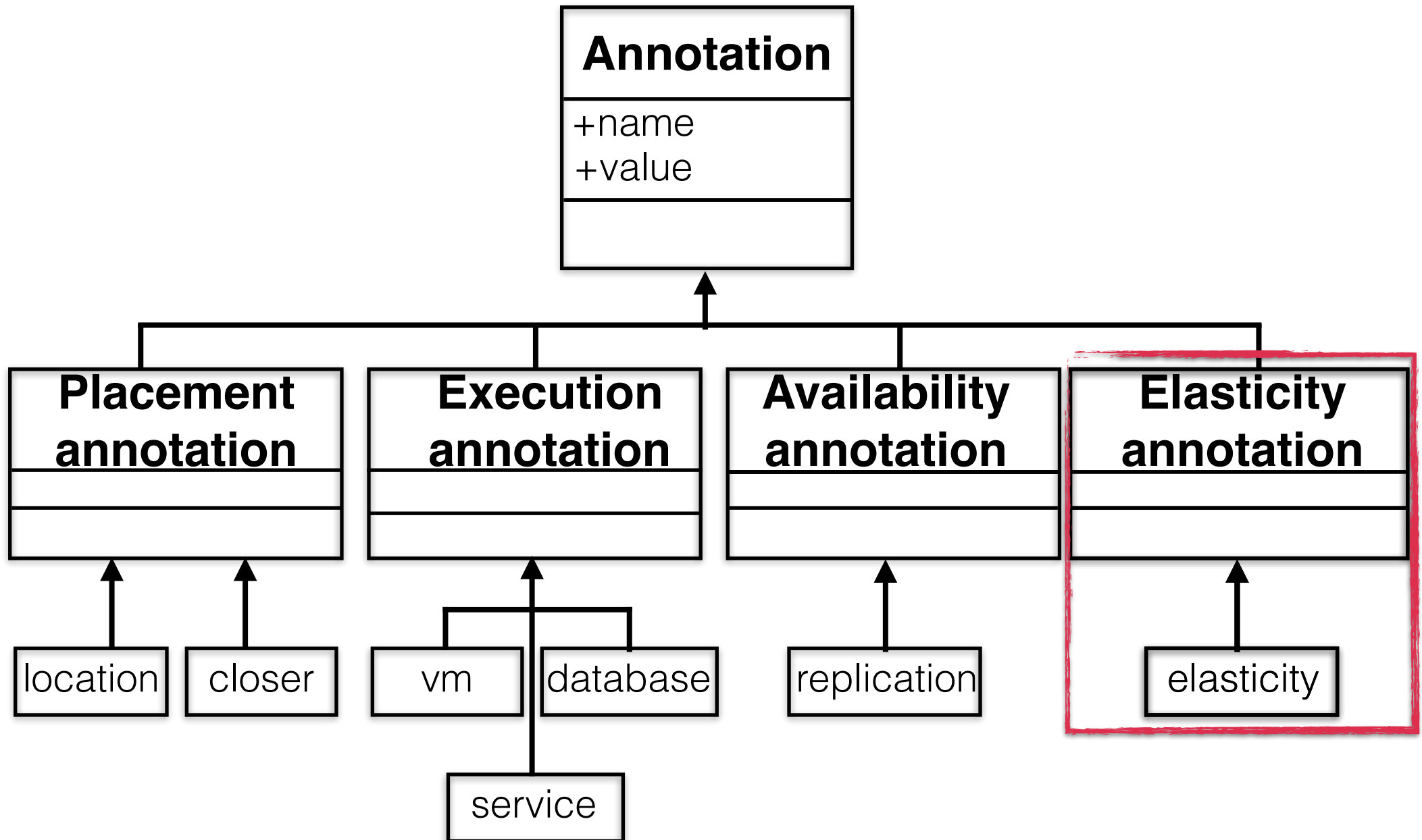
Example



become

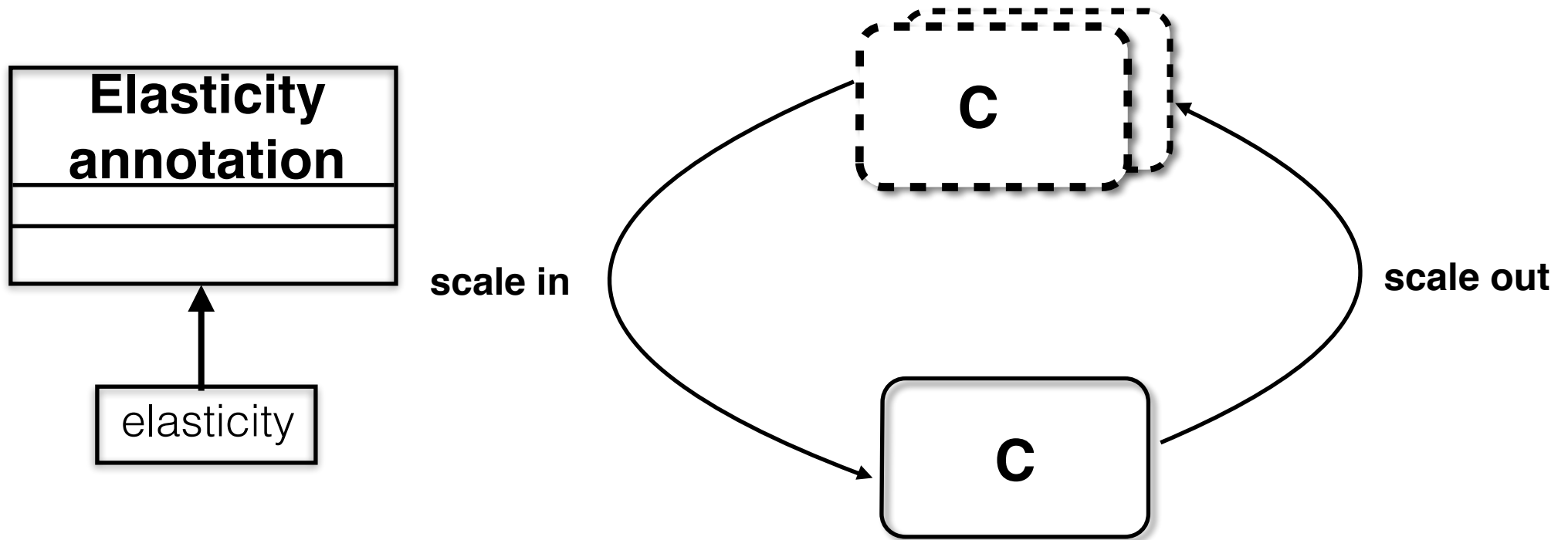


soCloud Model: annotations



soCloud Model: annotations

1 `@elasticity = 'description'`



A **DSL** for describing elasticity

soCloud Model: elasticity language

Event **Action** Condition

scaling up when (**average (cpuUsage, 120s) > 80%**)

minimize availability when (**totalCost(costCompute, 24 h) > 900**)

Elasticity is expressed on the **Resources, Cost, Quality**

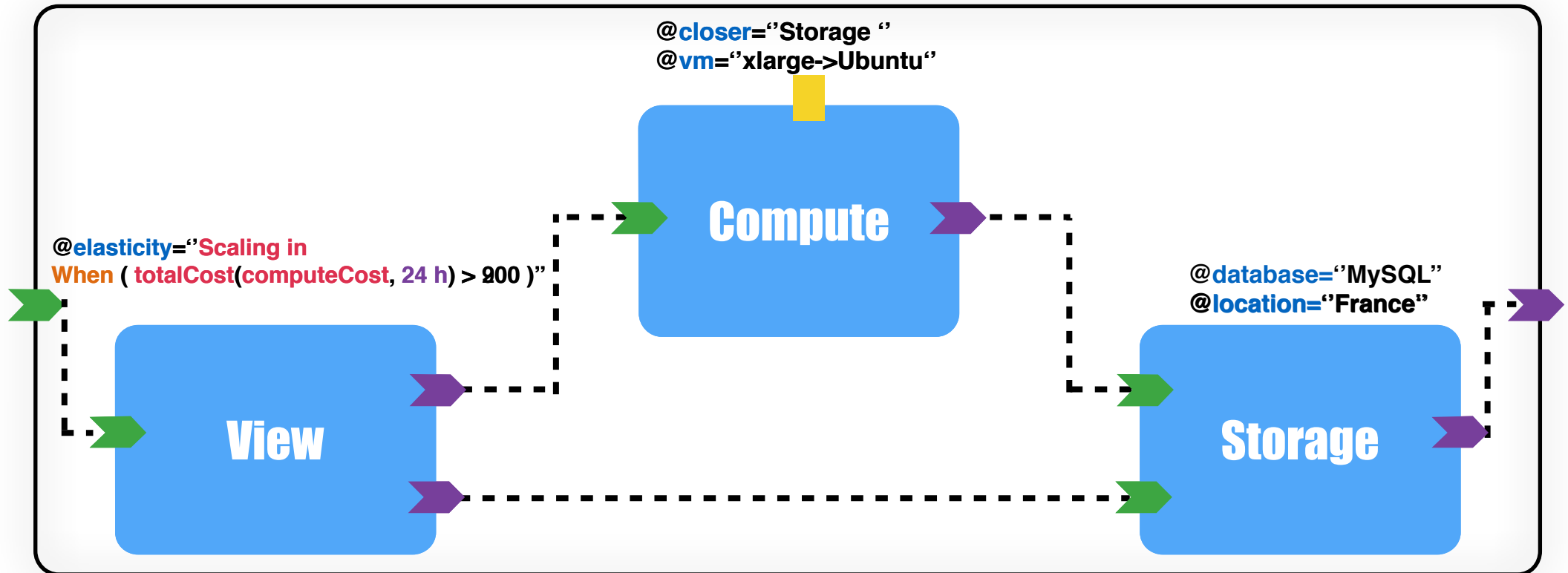
soCloud Model: elasticity language

Trigger

scaling in 5 at (20:00 Friday)

soCloud Model: annotations

3-Tiers Application



Summary

- We show how we use **annotation** to describe **non-functional** properties and manage each component as **unit** of execution
- New language is proposed to effectively **express** the **elasticity**

- Paraiso F, Merle P and Seinturier L : *soCloud : A service- oriented component-based PaaS for managing portability, provisioning, elasticity et high availability across multiple clouds*. **Springer Computing Journal** (Submitted)
- Haderer N, Paraiso F, Ribeiro C, Merle P, Rouvoy R and Seinturier L : *A Cloud-based Infrastructure for Crowd-sourcing Data from Mobile Devices*. **Springer Book** (To appear)

Outline

1. Context and motivation

2. Challenges

3. State of the art

4. Contributions

4.1. soCloud Model

4.2. soCloud Platform

5. Validation

6. Conclusion & Perspectives

soCloud Platform

- The **expectations** in term of **execution support** for distributed applications built with soCloud Model are **different**
- We need to provide a Platform that manages:
 - ❖ Multi-Cloud environments
 - ❖ Distributed applications in Multi-Cloud environments

soCloud Platform: concept

We need to build Multi-Cloud Platform that:

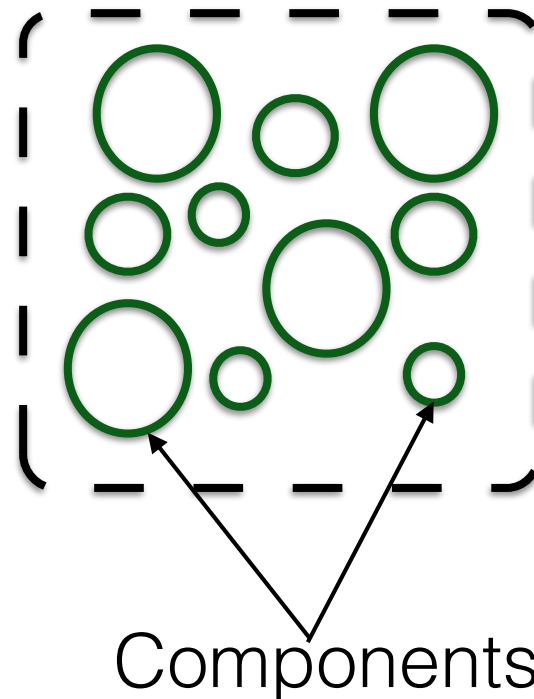
- **react** to load —> **Scalable**
- **react** to event —> **Event-Driven**
- **react** to failure —> **Fault-Tolerance**
- **react** to change —> **Responsive**
- **self** management —> **Autonomic**
- is **flexible** —> **Component-based**



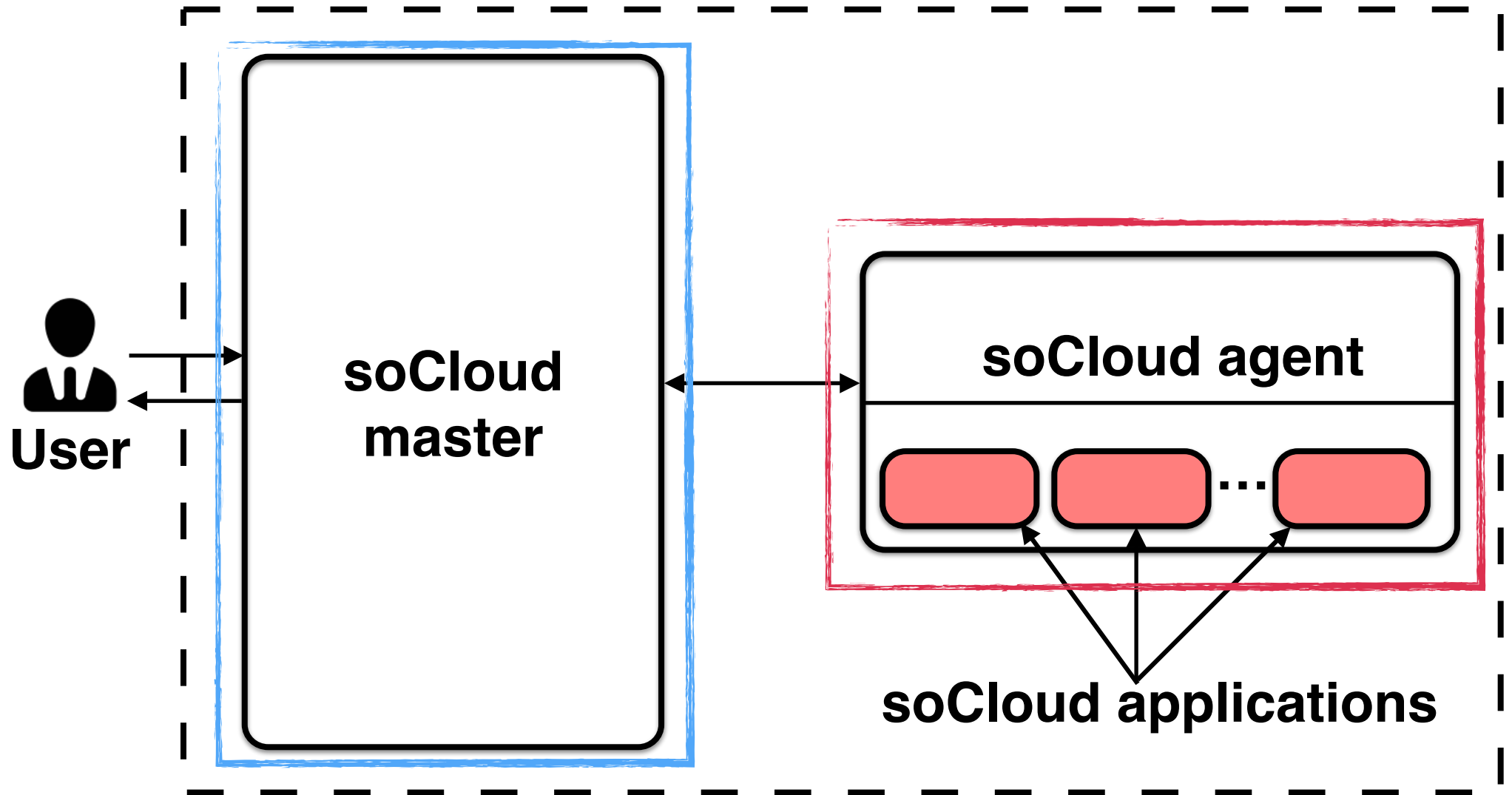
Reactive, flexible and **self** management platform

soCloud Platform

- soCloud Platform is a distributed **component-based** PaaS for managing
 - ❖ Portability
 - ❖ Provisioning
 - ❖ Elasticity
 - ❖ High-availability



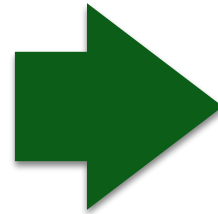
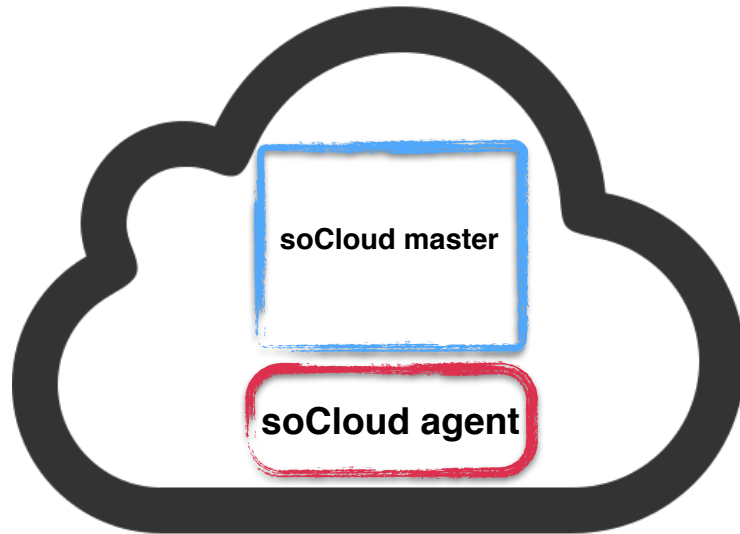
soCloud platform high level view



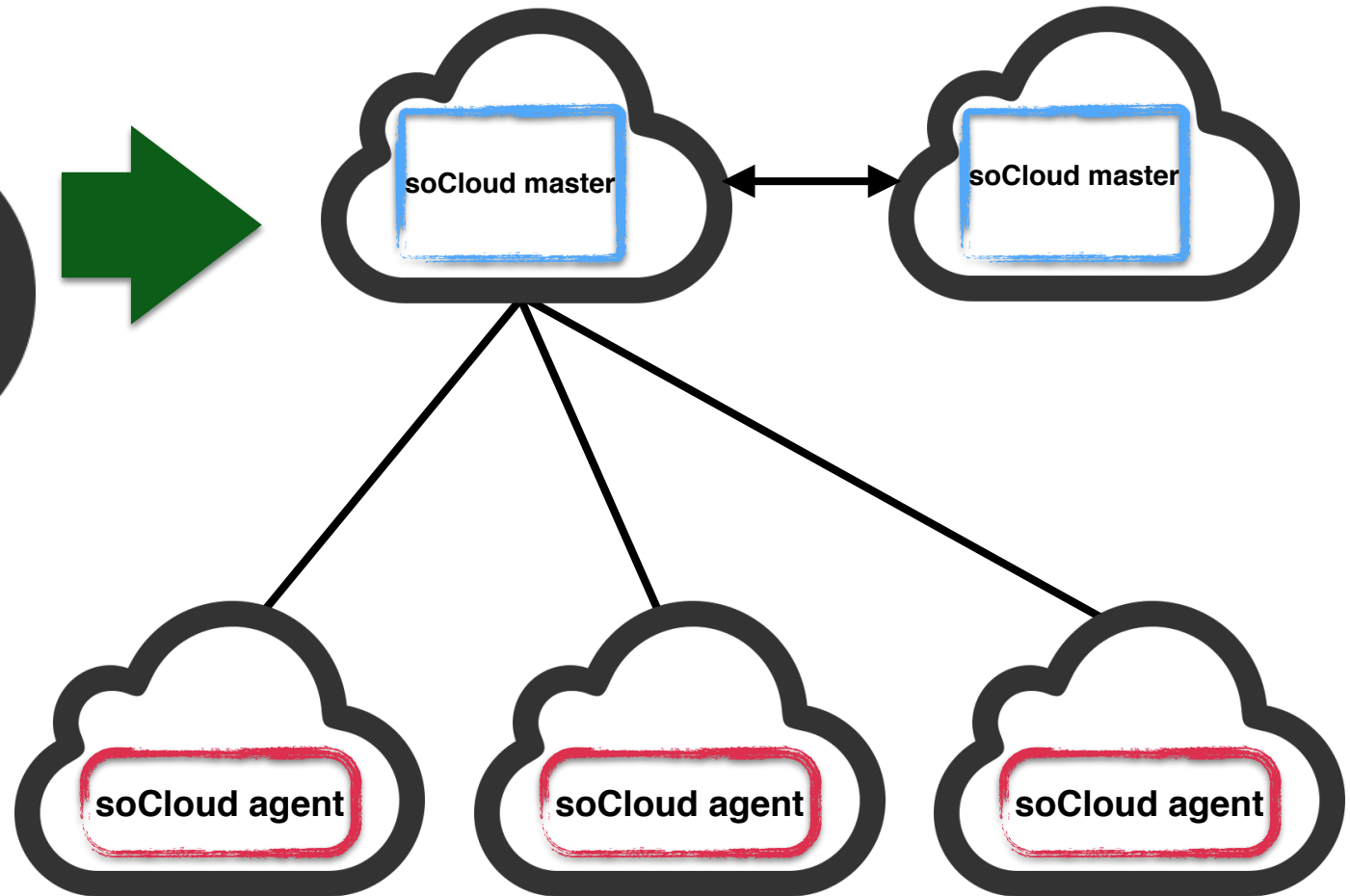
soCloud Platform: Multi-Cloud centric Architecture

Trend in the soCloud Platform Architecture

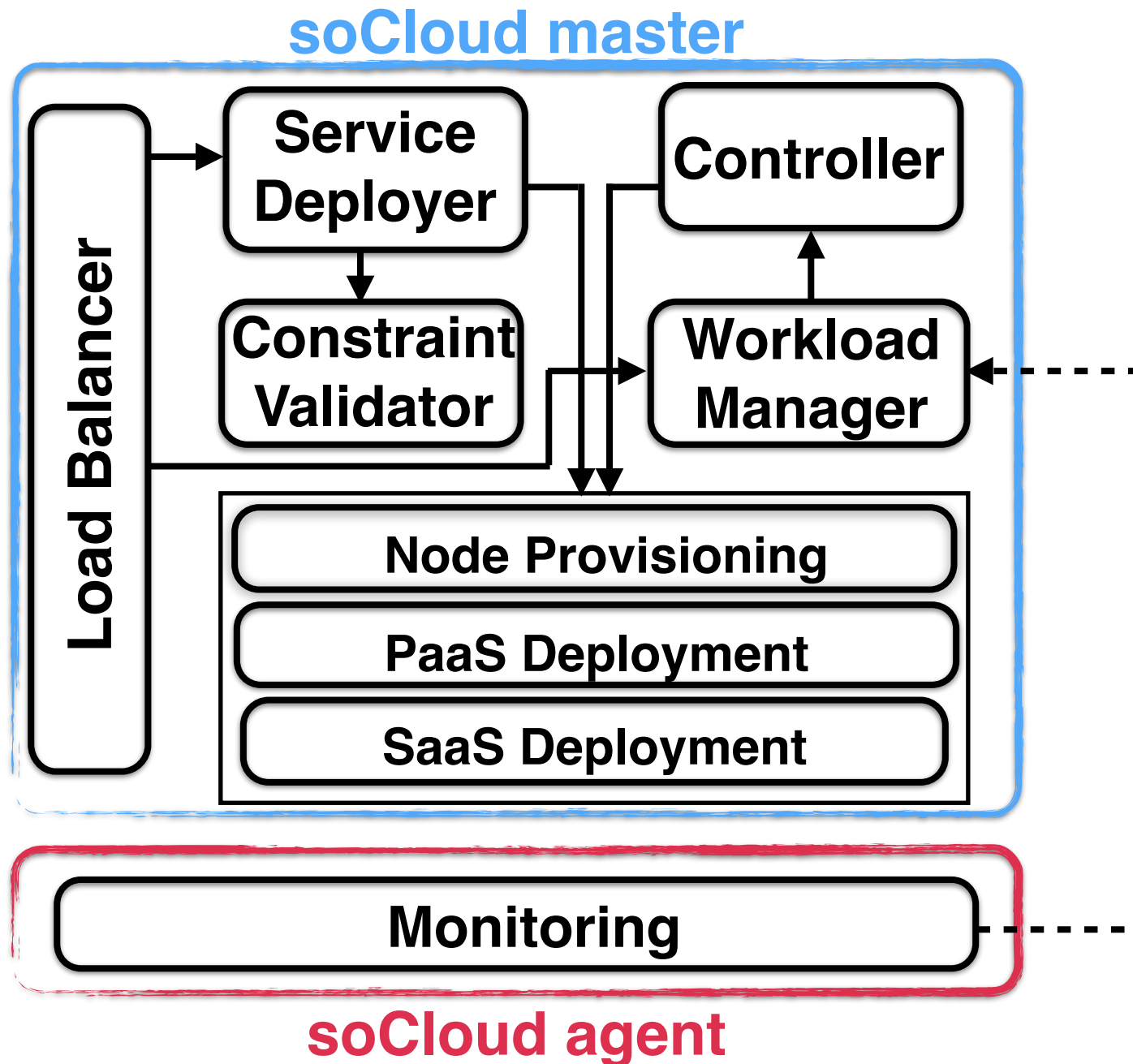
single Cloud centric Architecture



Multi-Cloud centric Architecture



soCloud Platform detail view



soCloud Platform: Fault Tolerance

Let it **Crash**

Application level

Replication in
different clouds

Platform level

Replication in
different clouds

soCloud Platform: Fault Tolerance

To achieve this

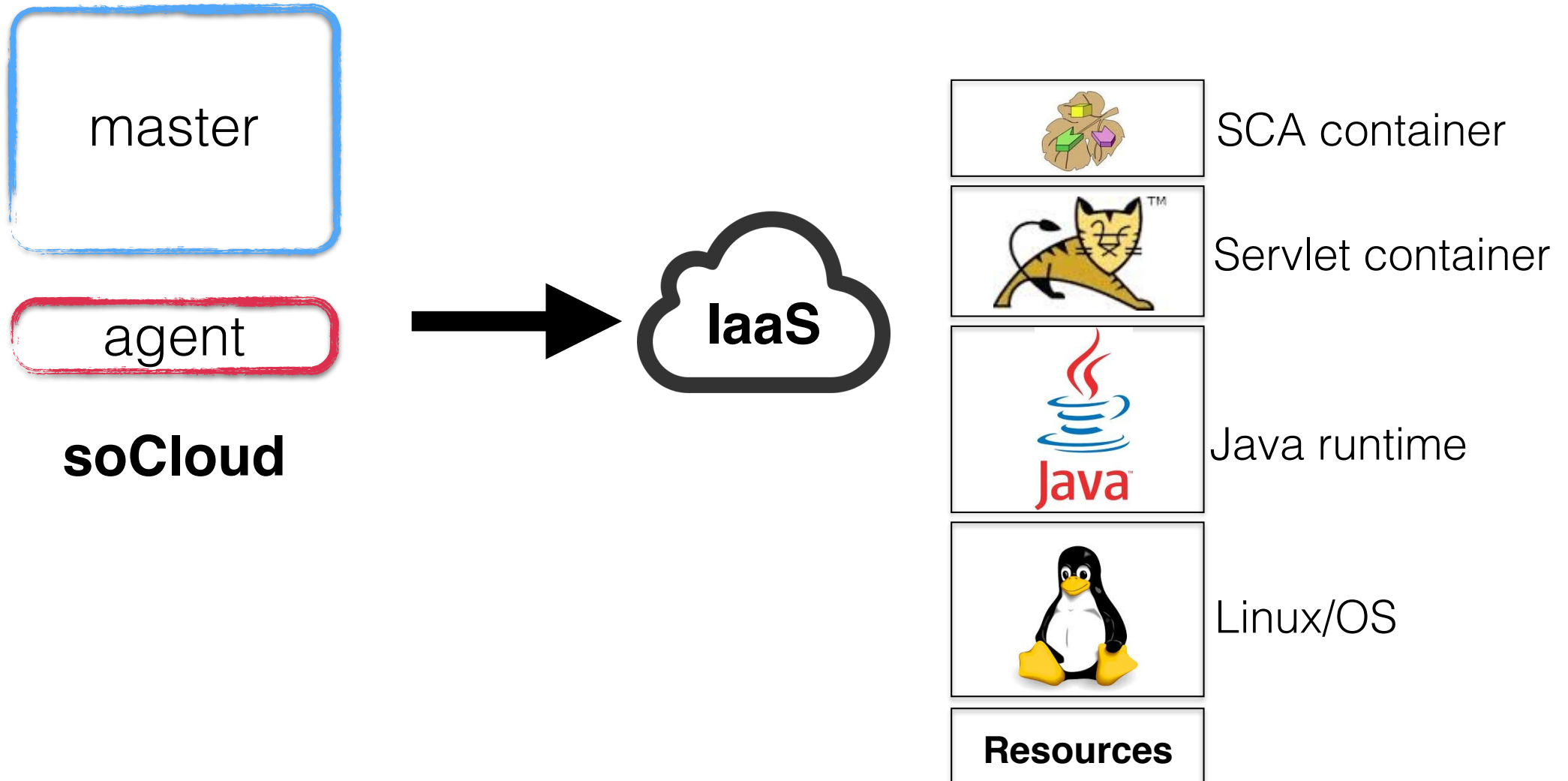
1. **Transparency** is the ultimate goal [Waldo et. al]
2. **Automatic** component and applications **replication** [Waldo et. al]
3. All replications are **equal** and **deterministic** [Waldo et. al]

[Waldo et. al]-Classic paper: A Note On Distributed Computing

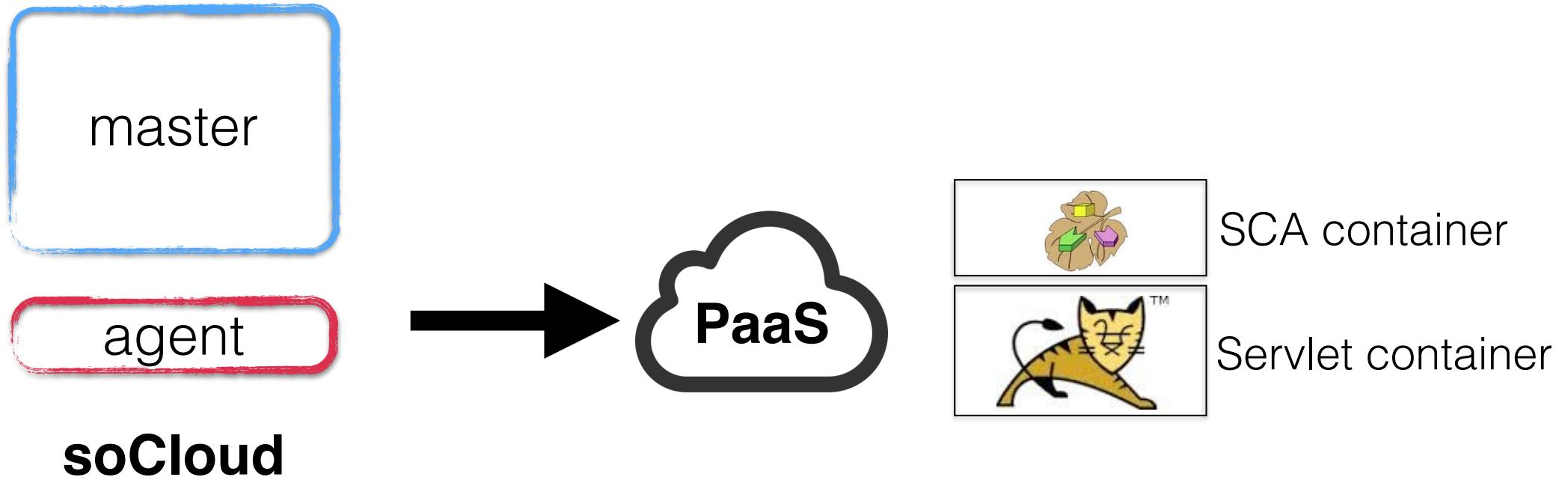
soCloud Platform: Replication features

- A cluster of N servers distributed across several **Clouds**
- Any (exactly one) component can be **leader**
- Active replication by the **leader**
- Consensus **election** of the leader
- Automatic **failover**
- Automatic **recovery**

soCloud Platform: deployment stack



soCloud Platform: deployment stack



Summary

- Runtime support for managing Multi-Cloud **portability**, **provisioning**, **elasticity** and **high-availability**
- **Reactive** Platform

- PARAISO Fawaz et.al : *A federated multi-cloud PaaS infrastructure*. In **IEEE 5th International Conference on Cloud Computing (CLOUD)**, pages 392–399., *Hawaii IEEE*, 2012.
- PARAISO Fawaz et al.: *Managing elasticity across multiple cloud providers*. In **Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds**, pages 53–60. **ACM**, 2013.

Outline

1. Context and motivation

2. Challenges

3. State of the art

4. Contributions

4.1. soCloud Model

4.2. soCloud Platform

5. Validation

6. Conclusion & Perspectives

soCloud Model

soCloud Platform

Validation: soCloud Model

Modeling of three concrete applications using the
soCloud Model

1. APISENSE application



2. DiCEPE application

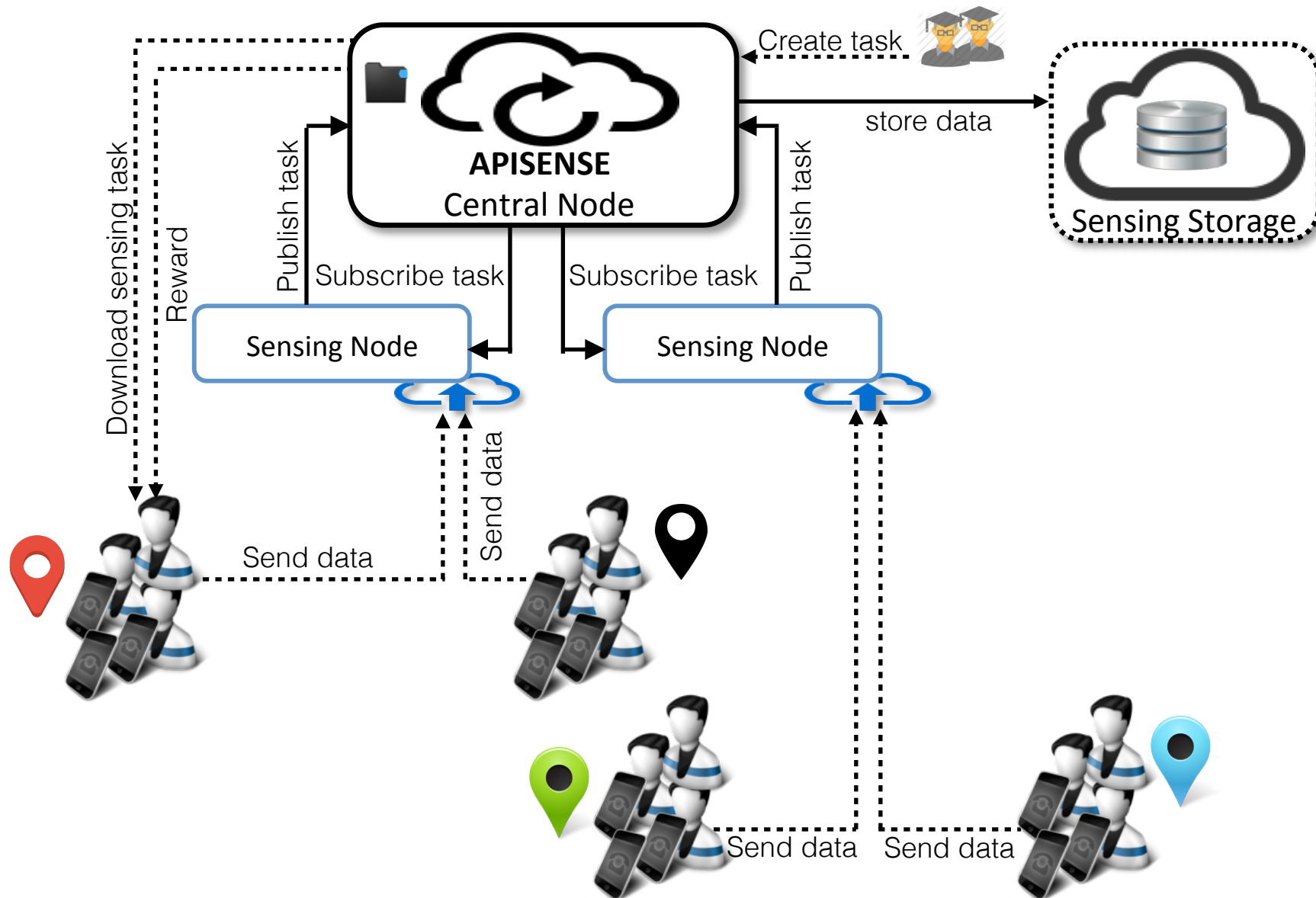


3. P2P Monitoring application



Validation: soCloud Model

1. APISENSE application [Nicolas Haderer]



Validation: soCloud Model

- Geo-location
 - ✦ Paris
- Unpredictable growth of smartphones
- Availability despite failures
- Cost control

Validation: soCloud Model

```
<composite name="Application-APISENSE">
```

```
  <component name="SensingNode">
```

```
    <implementation.contribution contribution="sensingnode.zip"/>
```

```
    <reference name="compute" target="CentralNode/compute"/>
```

```
    <reference name="storage" target="SensingStorage/storage"/>
```

```
    <annotation name="location">Paris</annotation>
```

```
    <annotation name="replication">2</annotation>
```

```
    <annotation name="elasticity">
```

```
      scaling in when (totalCost(computeCost, 24h) > 1000)
```

```
    </annotation>
```

```
  </component>
```

```
</composite>
```

Summary

- The soCloud Model has enabled us to build an App for collecting **data from smartphones**, an App to **integrate heterogenous CEP Engines** and make **Big Data**, and finally a **P2P distributed App**

- PARAISO Fawaz et.al : *A federated multi-cloud PaaS infrastructure*. In **IEEE 5th International Conference on Cloud Computing (CLOUD)**, pages 392–399., *Hawaii IEEE*, 2012.
- PARAISO Fawaz et. al.: *A Middleware Platform to Federate Complex Event Processing*. In **Sixteenth IEEE International EDOC Conference**, pages 113–122, *Beijing, China*, septembre 2012. Springer.

Validation: soCloud Platform

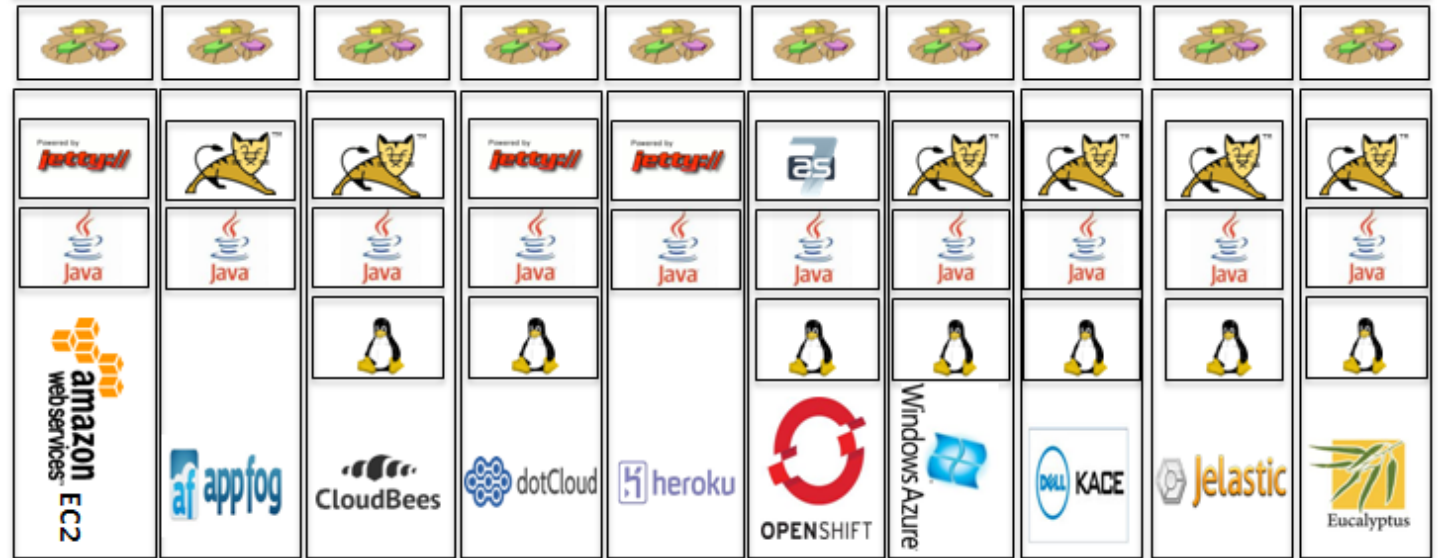
1. Portability
2. High-availability
3. Elasticity
4. Overhead introduced by soCloud

Portability

Validation: soCloud Platform

Deployed on

10 Clouds



IaaS and PaaS

High-availability

Validation: soCloud Platform

$$\text{Availability} = \frac{\text{MTBF}^*}{\text{MTBF} + \text{MTTR}^{**}} \quad [\text{Marcus et. al.}]$$

MTBF* = Mean **T**ime **B**etween **F**ailure

MTTR** = Mean **T**ime **T**o **R**ecover

[Marcus et. al.] : Blueprints for High availability

Validation: soCloud Platform

	MTTR* (Hour)	MTTR (Minute)	Ratio
soCloud	0.06 Hour	3.6 Minutes	-
Public clouds [IWGCR]	7.5 Hours	450 Minutes	125

MTTR* = Mean Time To Recover

[IWGCR] = International Working Group on Cloud Computing Resiliency. <http://iwgcr.org>

Validation: soCloud Platform

If it is assumed that a failure occurs once per year

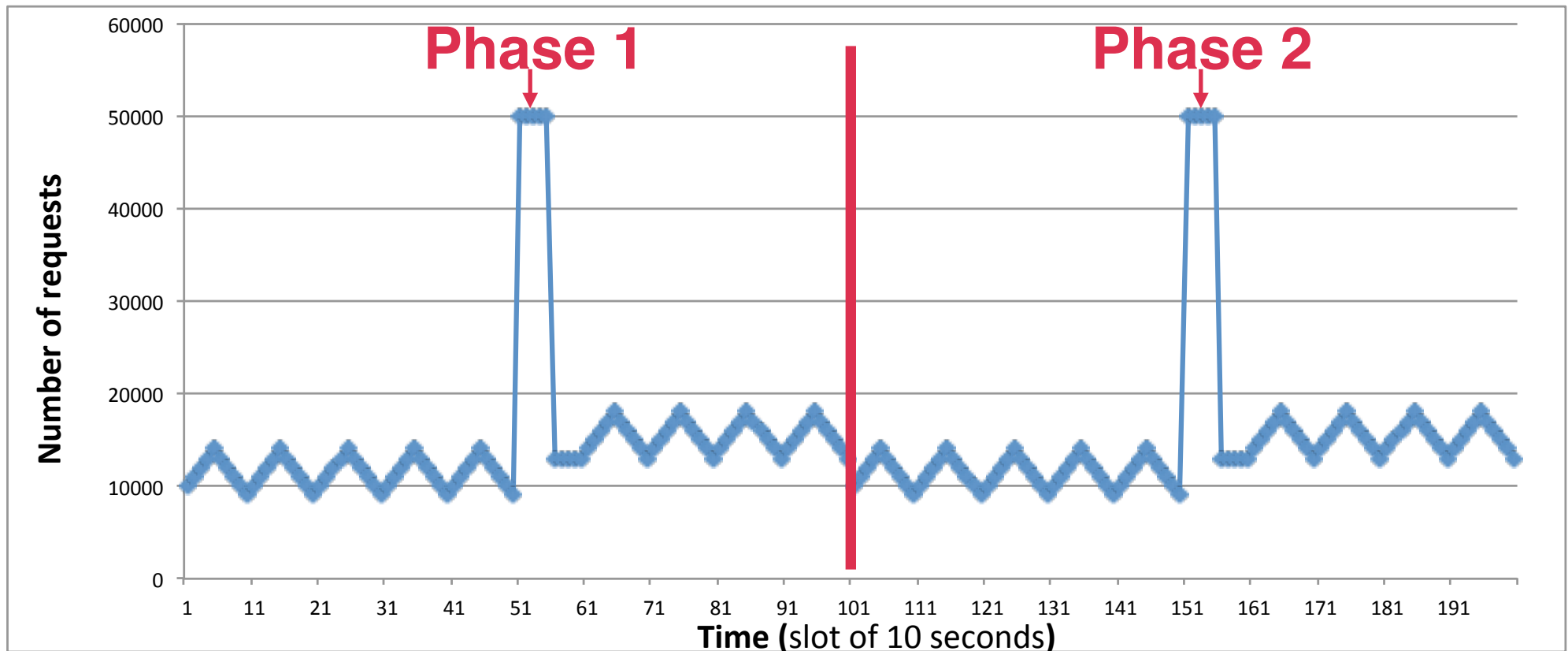
$$\text{MTBF} = 8760 \text{ Hours}$$

	Availability
soCloud	$\frac{8760}{8760 + 0.06} = 99.999\%$
Public clouds	$\frac{8760}{8760 + 7.5} = 99.914\%$

Elasticity

Flash crowd effect

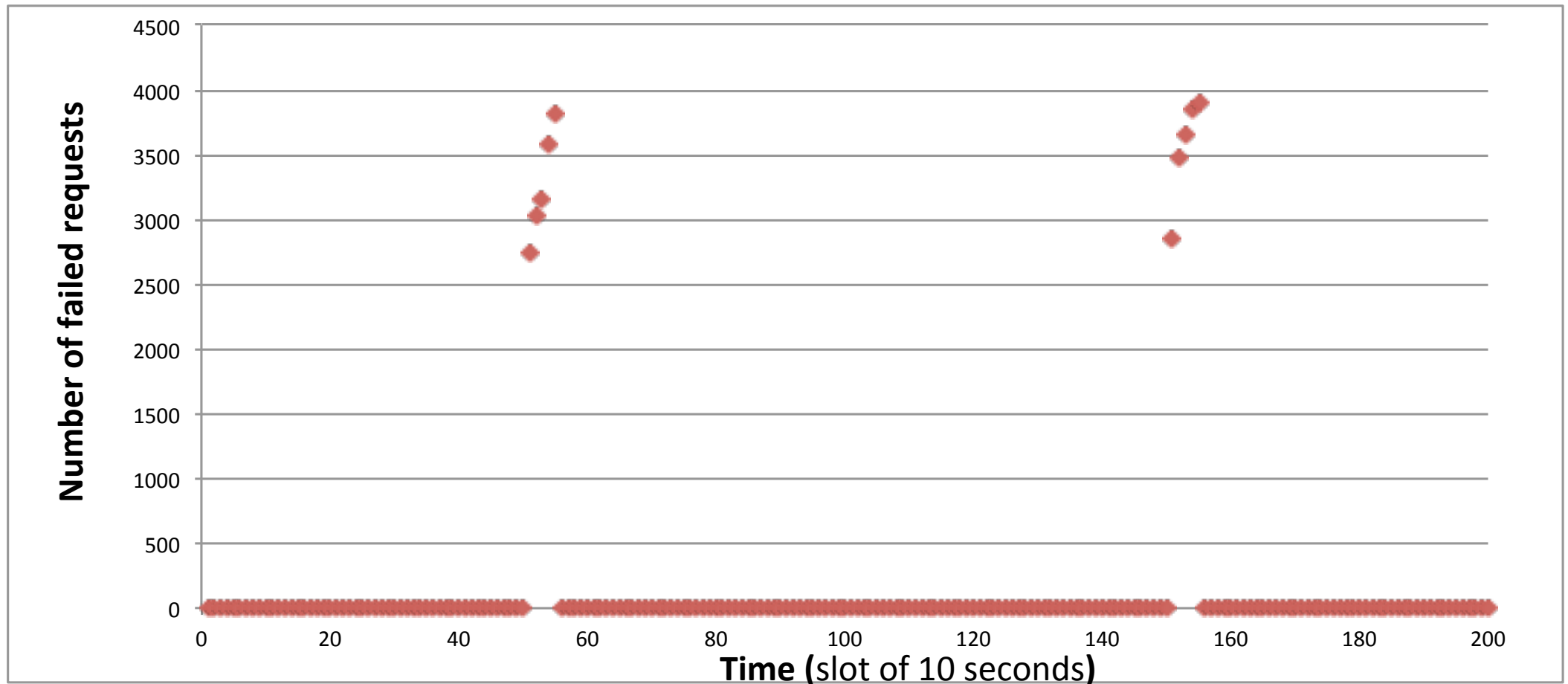
3-Tiers application was deployed on ten cloud providers



(a)

Total Number of Request = **3020000**

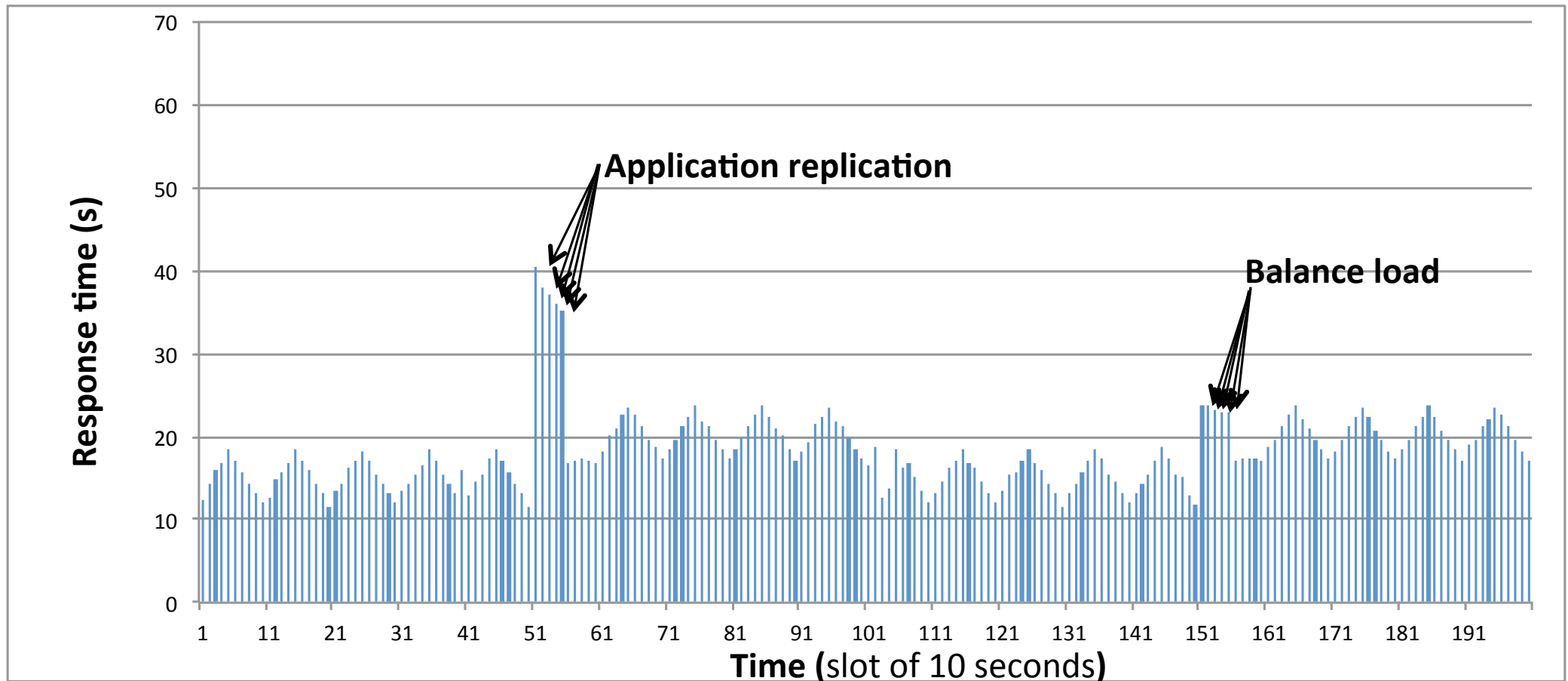
Flash crowd effect without soCloud elasticity



1.3% of requests are failed that correspond to **34039**

Response Time = **65.90 s**

Flash crowd effect with soCloud elasticity



No request has failed

Without soCloud elasticity, the Response Time = **65.90 s**

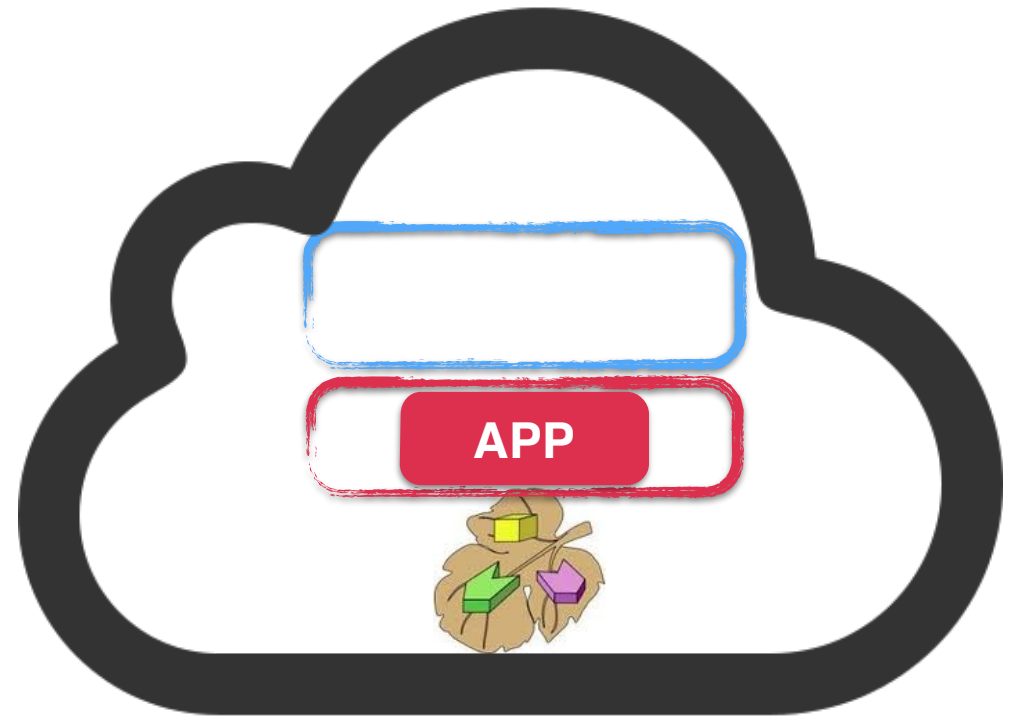
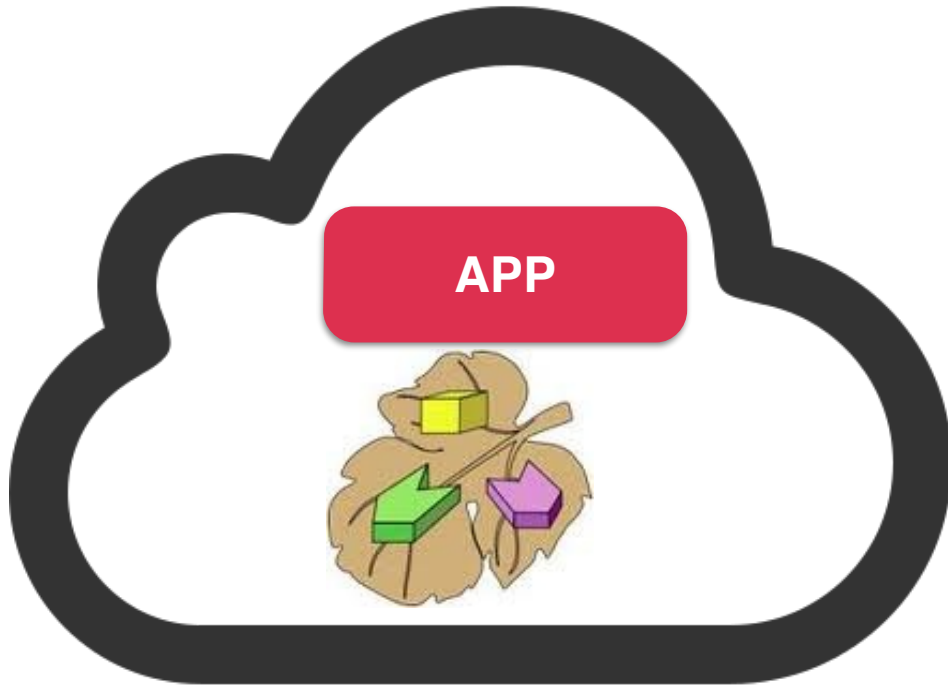
Response Time = **37.3 s**

Phase 1

Response Time = **23.38 s**

Phase 2

Overhead introduced by soCloud



Implementation	Execution time	Overhead introduced by soCloud
(Application + FraSCAti)		
(Application + FraSCAti + soCloud)		

Overhead introduced by soCloud

Implementation	Execution time	Overhead introduced by soCloud
(Application + FraSCati)	10.85 sec	-
(Application + FraSCati + soCloud)	11,10 sec	2.3%

The **benefit** provided by the soCloud Platform **outweighs** the difference in the execution time

Summary

- **Reactivity** face:
 - ❖ Failures (**High-availability**)
 - ❖ Flash crowd effect (**Elasticity**)
- Negligible **Overhead** introduced.

Outline

1. Context and motivation

2. Challenges

3. State of the art

4. Contributions

4.1. soCloud Model

4.2. soCloud Platform

5. Validation

6. Conclusion & Perspectives

Conclusion

soCloud Model

- We use **annotations** to express **non-functional** requirements.
- New **language** is proposed to effectively express the elasticity.
- The soCloud Model is illustrated on three **distributed applications** deployed in Multi-Cloud environments.

soCloud Platform

- **Multi-Cloud PaaS** for deploying, executing and managing distributed application.
- It was deployed on **ten** IaaS/PaaS clouds providers.
- soCloud Platform is capable of providing Multi-Cloud **high-availability** and **elasticity** to applications deployed on it.

Perspectives

Short-term further work

- The high-availability management despite software bugs.
- The elasticity management using reinforcement learning.

Further Research Directions

- Security for Multi-Cloud.
- Sharing state between replicates.
- Take into account changes of the underlying platforms.

Thank you !!!

- PARAISO Fawaz, HADERER Nicolas, MERLE Philippe, ROUVOY Romain and SEINTURIER Lionel : *A federated multi-cloud PaaS infrastructure*. In **IEEE 5th International Conference on Cloud Computing (CLOUD)**, pages 392–399. IEEE, 2012.
- PARAISO Fawaz, HERMOSILLO Gabriel, ROUVOY Romain, MERLE Philippe, SEINTURIER Lionel : *A Middleware Platform to Federate Complex Event Processing*. In **Sixteenth IEEE International EDOC Conference**, pages 113–122, Beijing, China, septembre 2012. Springer.
- PARAISO Fawaz, MERLE Philippe and SEINTURIER Lionel : *Managing elasticity across multiple cloud providers*. In **Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds**, pages 53–60. ACM, 2013.
- PARAISO Fawaz, MERLE Philippe and SEINTURIER Lionel : *soCloud : A service- oriented component-based PaaS for managing portability, provisioning, elasticity et high availability across multiple clouds*. **Springer Computing Journal** (To appear)
- HADERER Nicolas, PARAISO Fawaz, RIBEIRO Christophe, MERLE Philippe, ROUVOY Romain and SEINTURIER Lionel : *A Cloud-based Infrastructure for Crowd-sourcing Data from Mobile Devices*. **Springer Review** (To appear)

NEW

NEW