Efficient Resource Management for Data Centers: The ACTiCLOUD Approach

Vasileios Karakostas, Georgios Goumas, Ewnetu Bayuh Lakew, Erik Elmroth, Stefanos Gerangelos, Simon Kolberg, Konstantinos Nikas, Stratos Psomadakis, Dimitrios Siakavaras, Petter Svard, Nectarios Koziris

SAMOS XVIII: International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation

Outline

- About ACTiCLOUD
  - General Information
  - Motivation
  - Objectives & Goal
  - Overview
- ACTiManager
  - Overview
  - Design Principles
  - Components
  - Execution lifetime of a VM under ACTiManager

ACTiCLOUD Information

EU H2020 Project

Grant Agreement Nº: 732366

Start date: 1 Jan 2017

Duration: 36 months

Partners:

ICCS ΕΠΙΣΕΥ

NUMASCALE

MANCHESTER 1824
The University of Manchester

KALEAO

monetdb solutions

neo4j

onapp

UMEÅ UNIVERSITET

Coordinator: ICCS

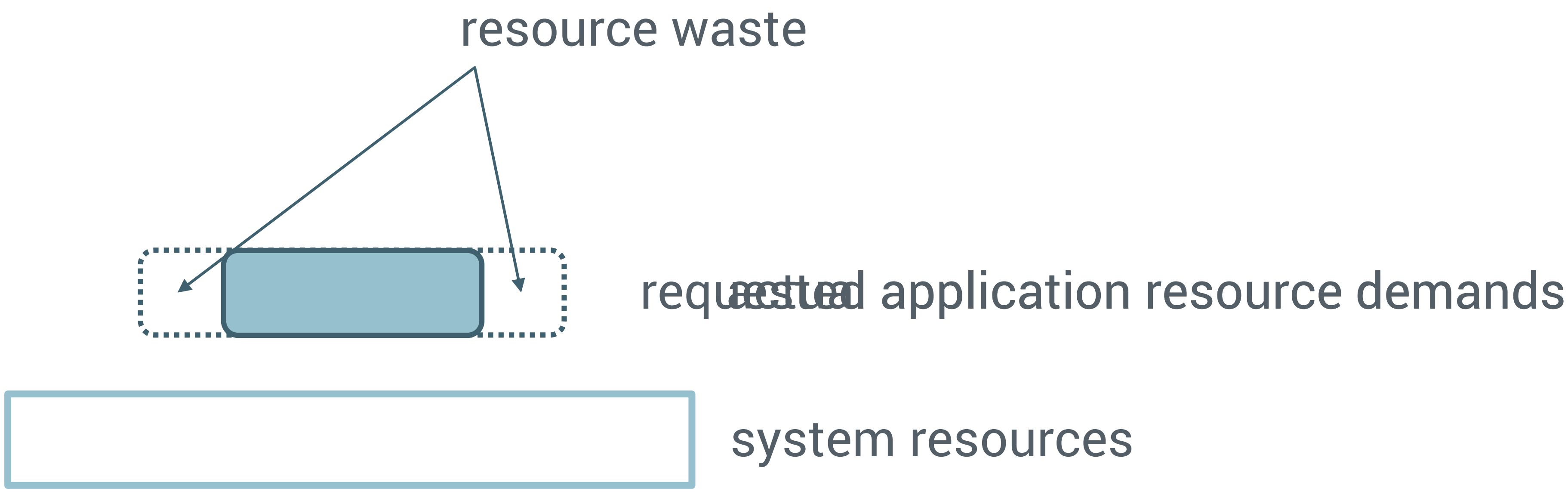
ACTiCLOUD Motivation

- **Key observation:** severe resource *misuse* in cloud offerings

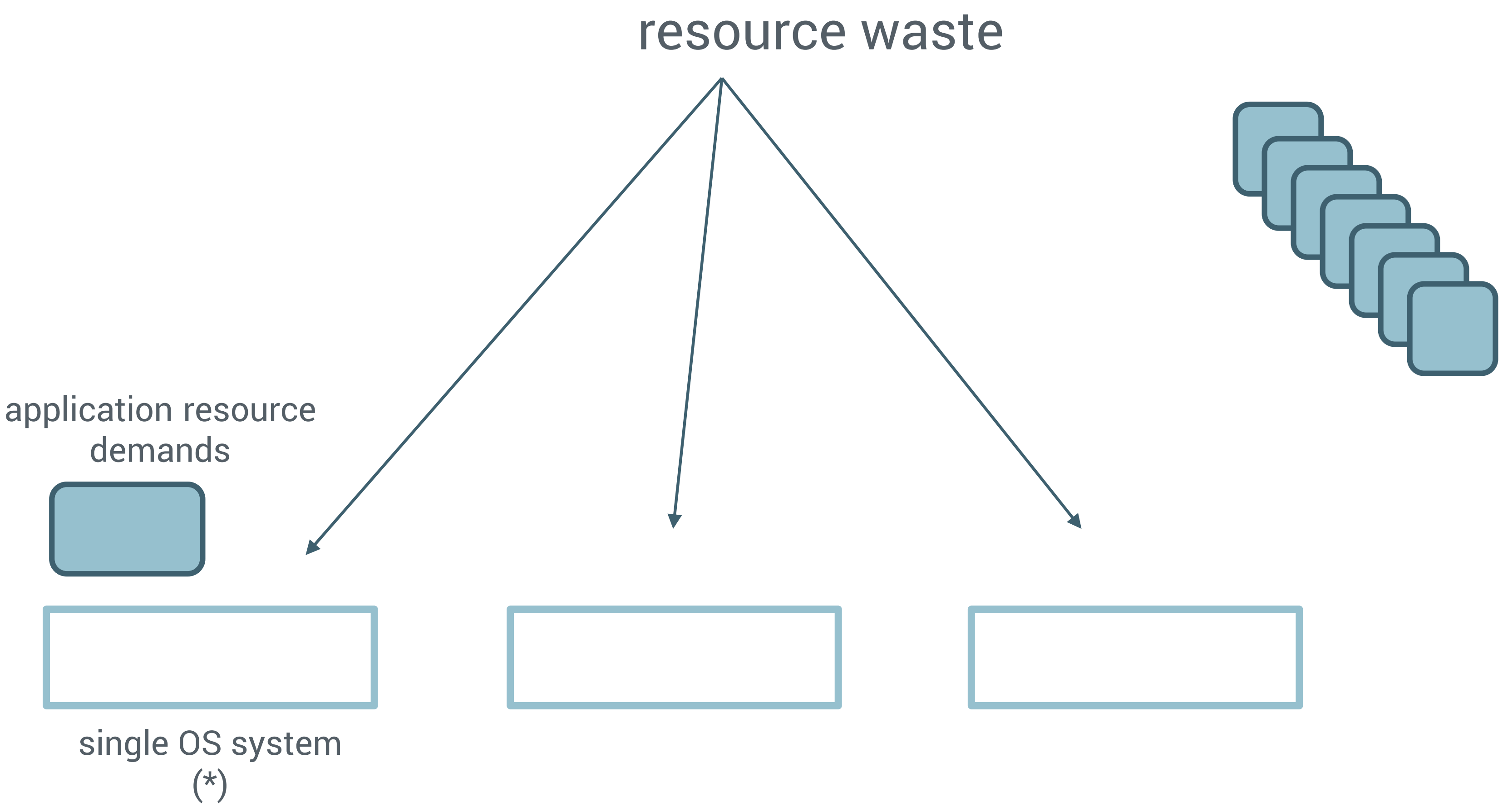
- **Fact 1:** Resource *waste*
  - *Underutilization*
  - *Fragmentation*
  - *Interference*

- **Fact 2:** Resource *shortage*
  - *Unavailability*

Resource waste due to *overprovisioning*

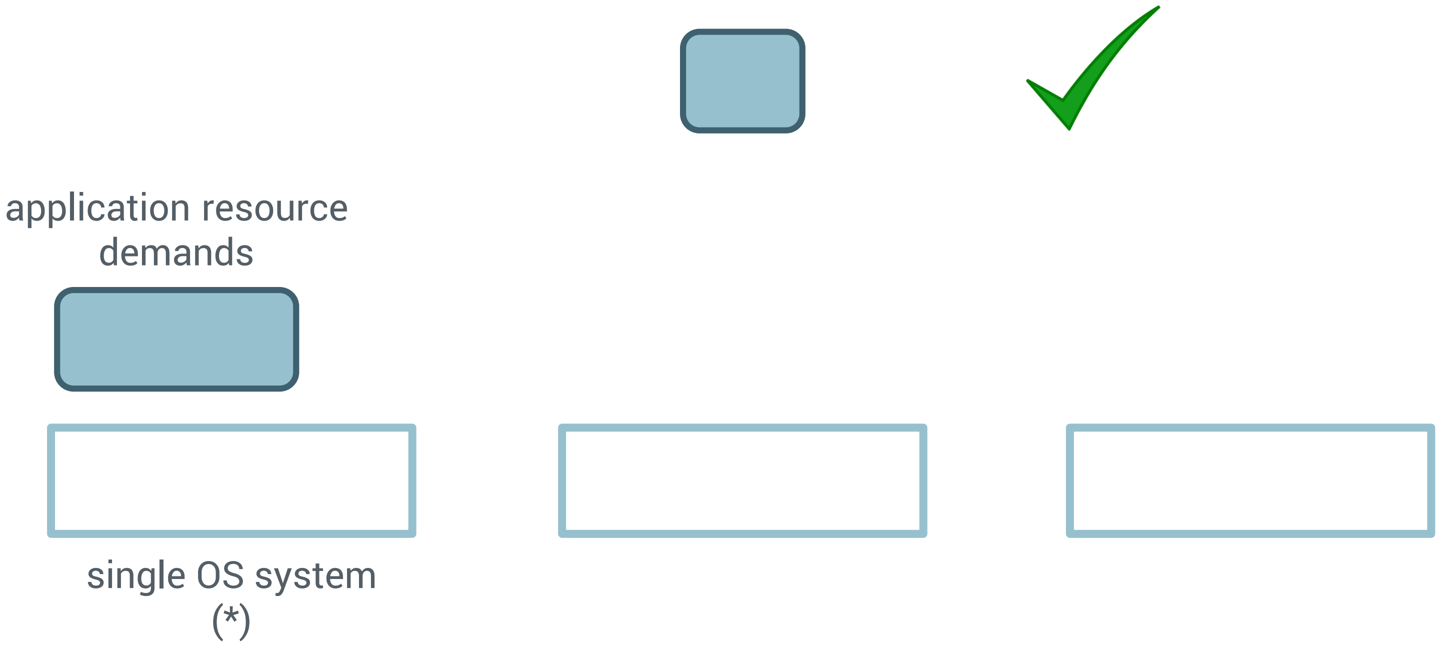


Resource waste due to *stand-by for peak traffic*



(*) typical cloud server

Resource waste due to *fragmentation*

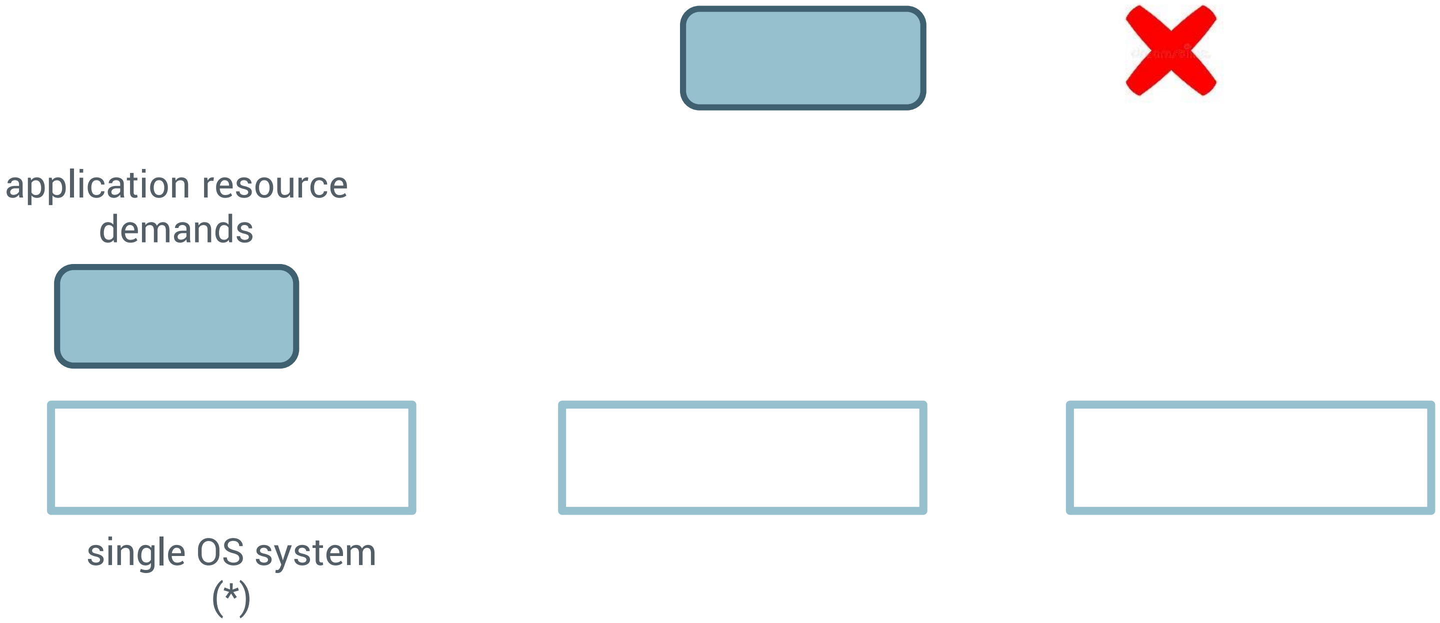


application resource demands

single OS system (*)

(*) typical cloud server

Resource waste due to *fragmentation*

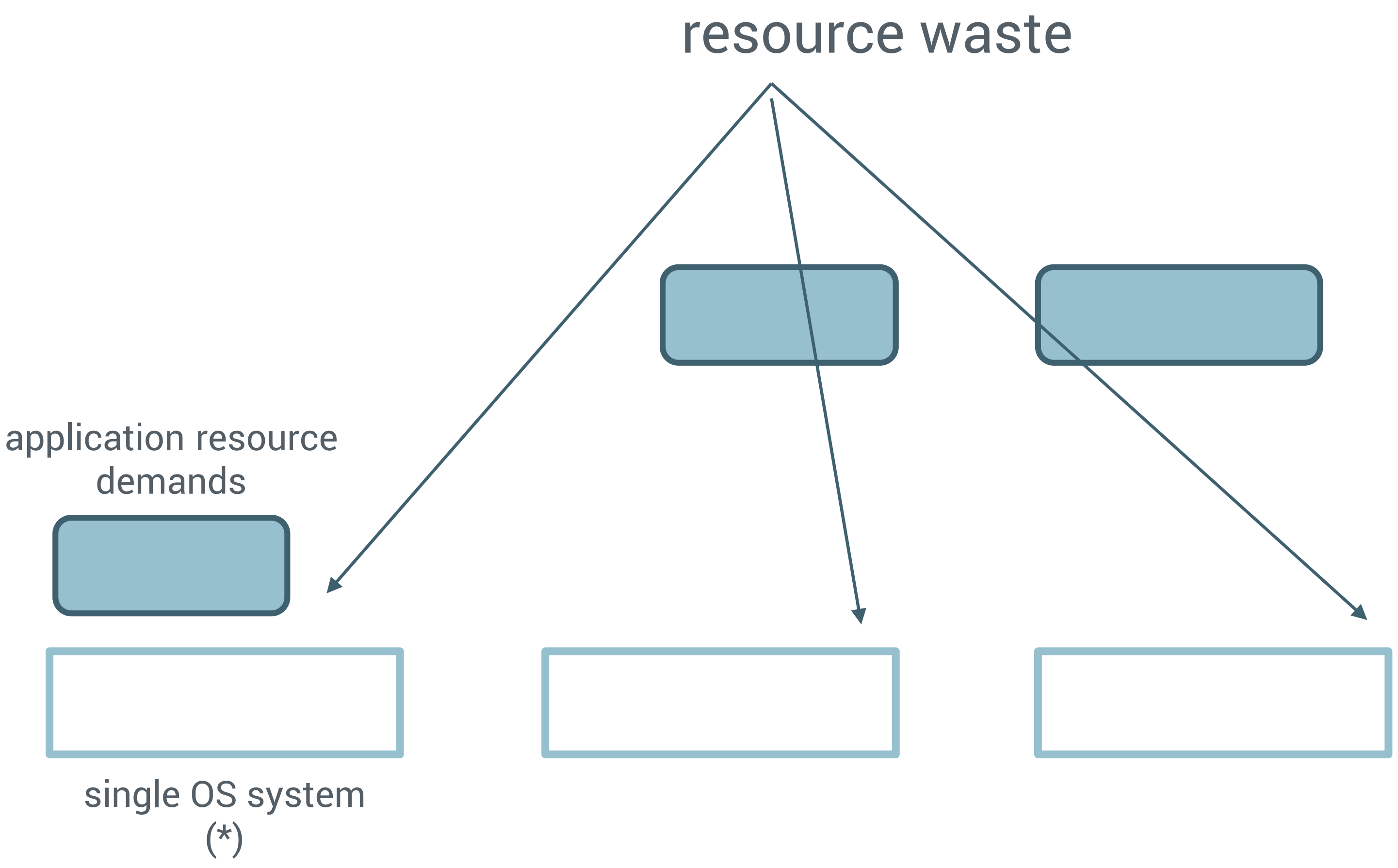


application resource demands

single OS system (*)

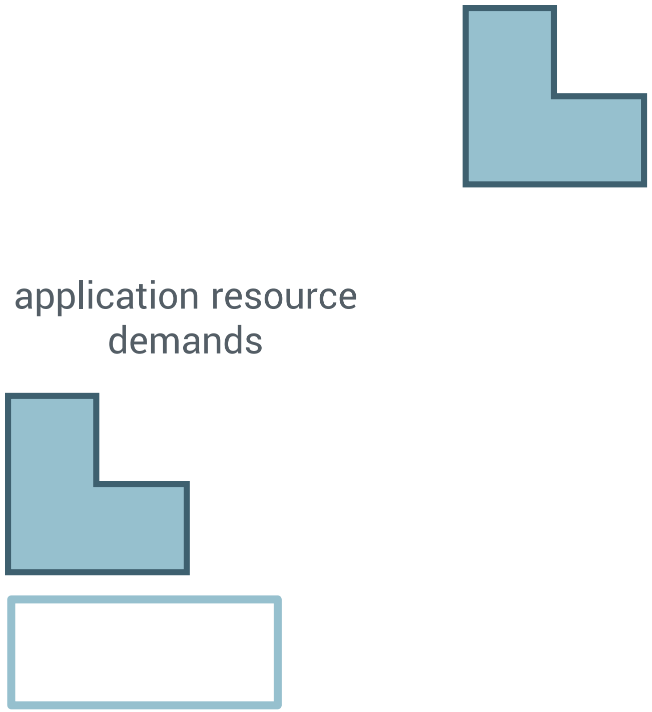
(*) typical cloud server

Resource waste due to *fragmentation*



(*) typical cloud server

Resource waste due to *contention* and *interference*



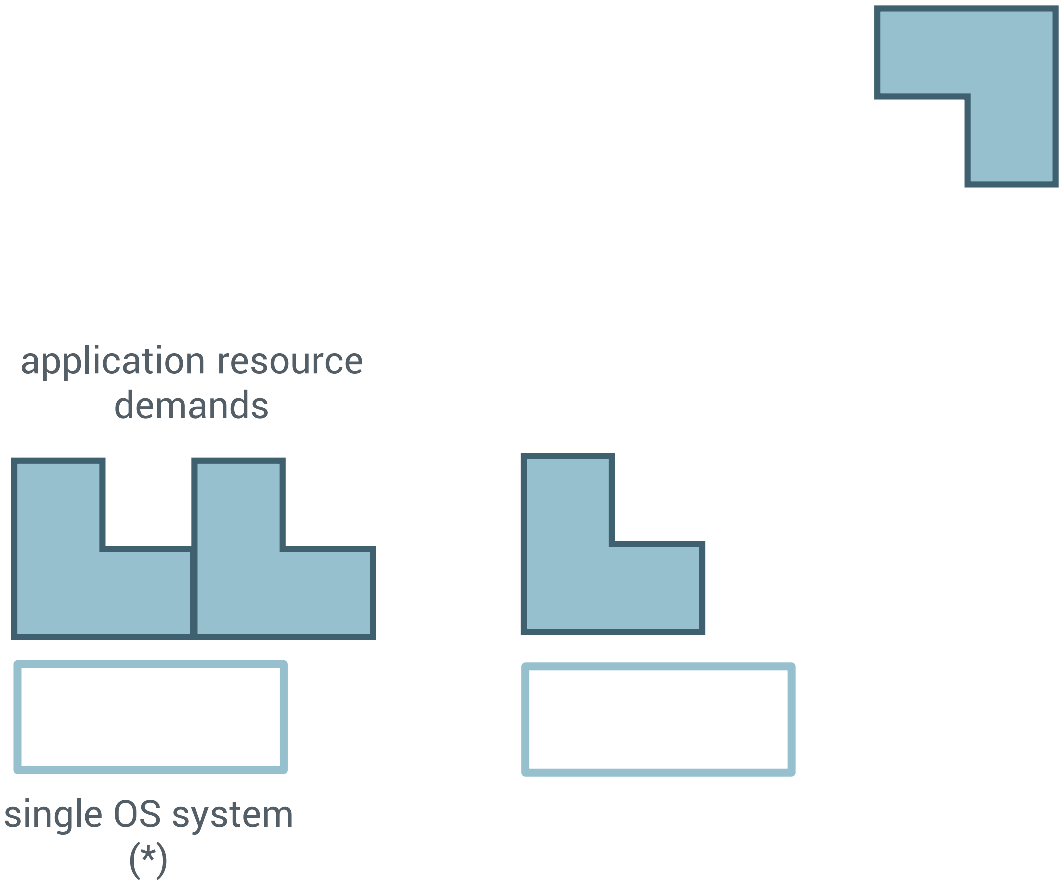
application resource demands

single OS system (*)

Collocated VMs with intensive and similar resource footprints may cause contention and performance **degradation** and **instability**.

(*) typical cloud server

Resource waste due to *contention* and *interference*

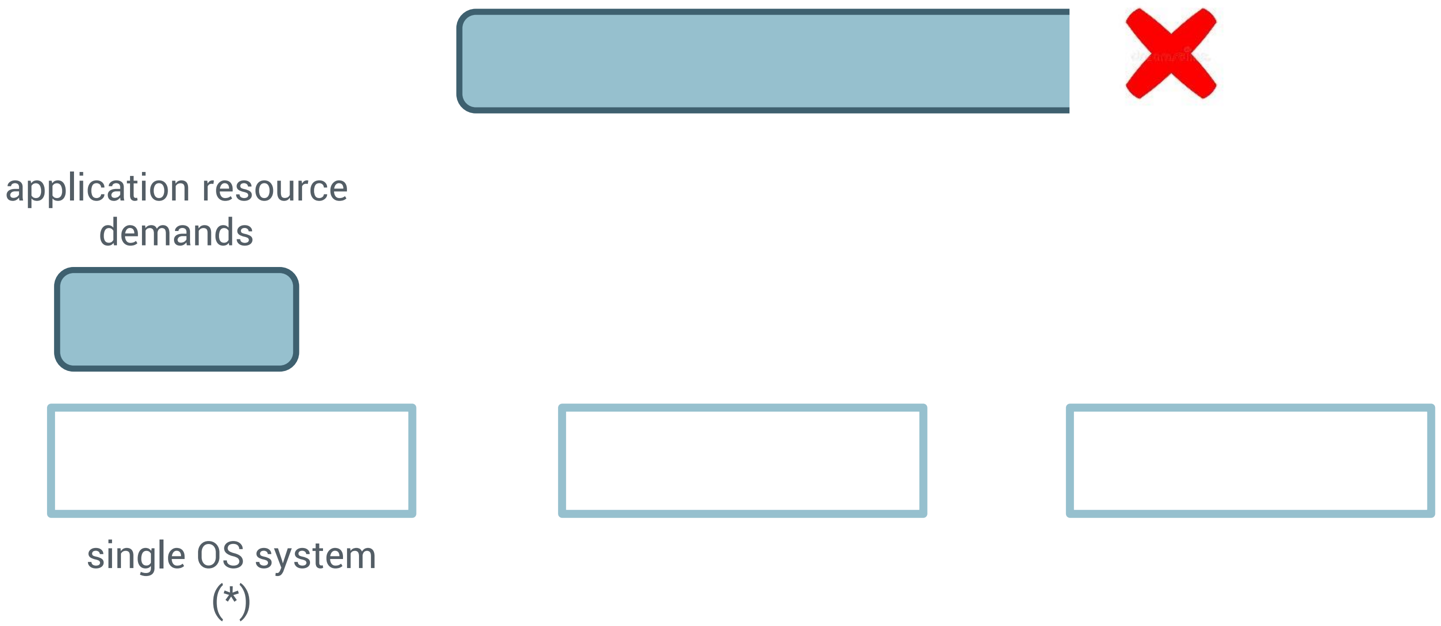


Collocated VMs with intensive and similar resource footprints may cause contention and performance **degradation** and **instability**.

Judicious collocation of VMs can mitigate the effect

(*) typical cloud server

Resource *unavailability*



application resource demands

single OS system (*)

(*) typical cloud server

ACTiCLOUD Objectives

- Effective utilization of cloud resources
  - Resource **efficiency** & Performance **stability**
- Deployment of resource demanding applications in the cloud
  - **Scalability** & **elasticity** in resource provisioning
- Special focus on database applications

ACTiCLOUD Goals

- Significant cost and performance improvements for Cloud Service Providers
- Higher performance, stability, and lower pricing for cloud applications
- Enhanced flexibility and scalability for database applications

ACTiCLOUD Approach

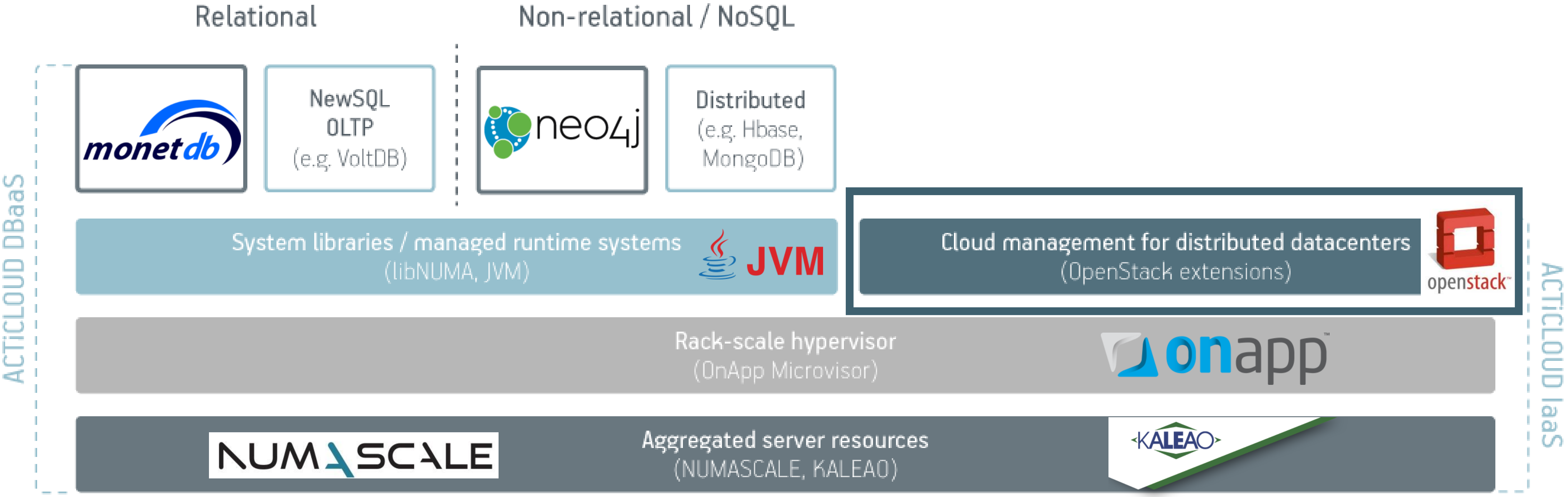
1. Resource underutilization
   - ACTiCLOUD: **effective prioritization and consolidation**

2. Resource fragmentation
   - ACTiCLOUD : **effective remapping, migration, and co-scheduling actions**

3. Resource contention
   - ACTiCLOUD : **accurate modeling and effective actions**

4. Resource unavailability
   - ACTiCLOUD : **Leverage platforms that break typical server architecture**

ACTiCLOUD Architecture at a Glance



Relational
Non-relational / NoSQL
monetdb
NewSQL OLTP (e.g. VoltDB)
neo4j
Distributed (e.g. Hbase, MongoDB)
System libraries / managed runtime systems (libNUMA, JVM)
JVM
Cloud management for distributed datacenters (OpenStack extensions)
openstack
Rack-scale hypervisor (OnApp Microvisor)
onapp
Aggregated server resources (NUMASCALE, KALEAO)
NUMASCALE
KALEAO
ACTiCLOUD DBaaS
ACTiCLOUD IaaS

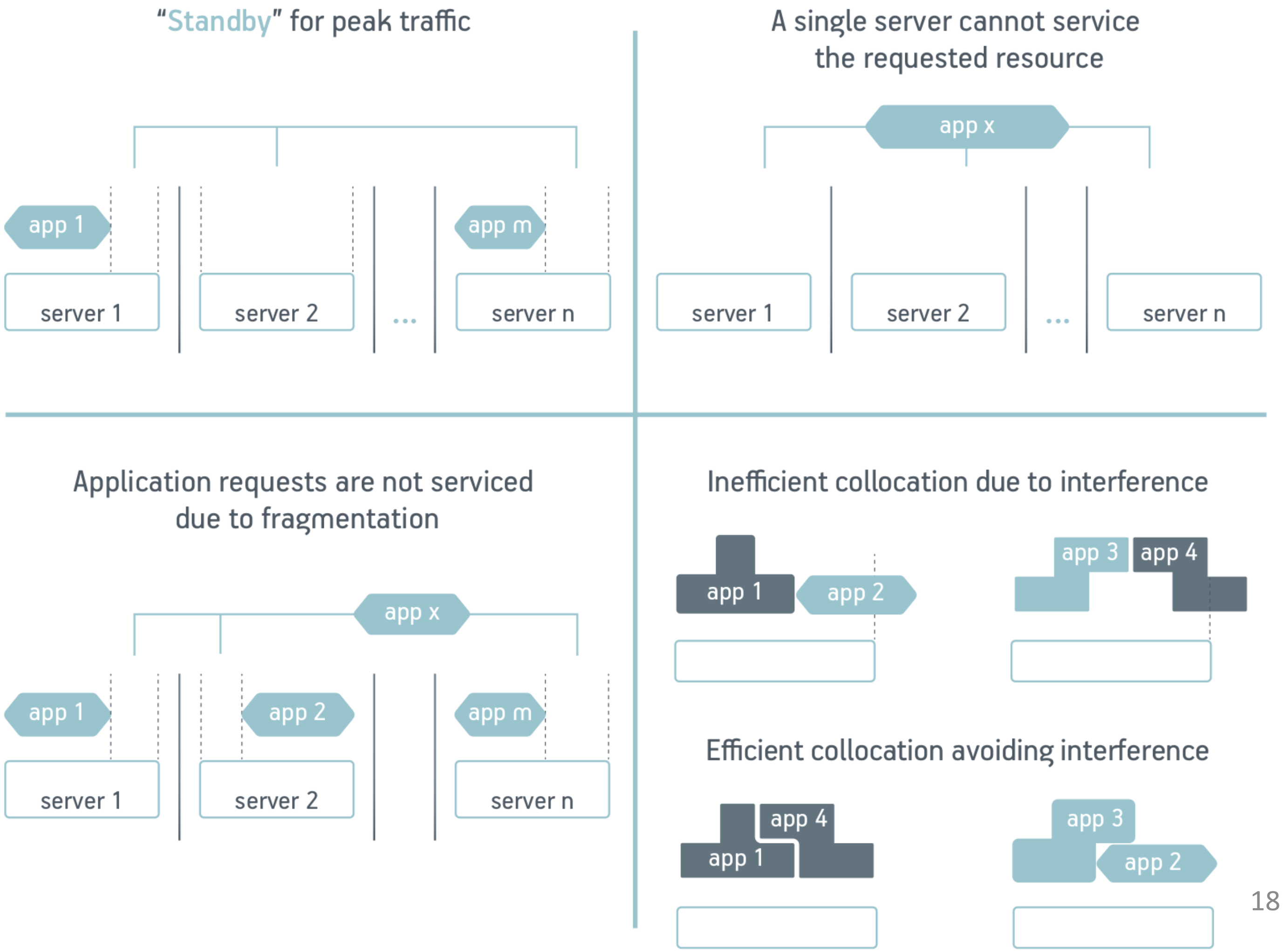
Outline

- About ACTiCLOUD
- ACTiManager
  - Overview
  - Design Principles
  - Components
  - Execution lifetime of a VM under ACTiManager

ACTiManager

- Plays a key role in realizing the project’s objectives and goals

- Resource management
- Prioritization
- Consolidation
- Resource utilization
- Interference mitigation


“Standby” for peak traffic
app 1
app m
server 1
server 2
...
server n
A single server cannot service the requested resource
app x
server 1
server 2
...
server n
Application requests are not serviced due to fragmentation
app x
app 1
app 2
app m
server 1
server 2
server n
Inefficient collocation due to interference
app 1
app 2
app 3
app 4
Efficient collocation avoiding interference
app 4
app 1
app 3
app 2

ACTiManager – Design Principles

1. To be scalable in large-scale cloud installations
   - Hierarchical design
   - Within the node & across nodes

2. To operate in the typical closed-loop control fashion
   - Monitoring and information aggregation
   - Modeling
   - Decision making

3. To minimize modifications to an existing cloud management system
   - Operate as "out-of-the-box" add-on component
   - We focus on OpenStack

ACTiManager – Design Principles (cont.)

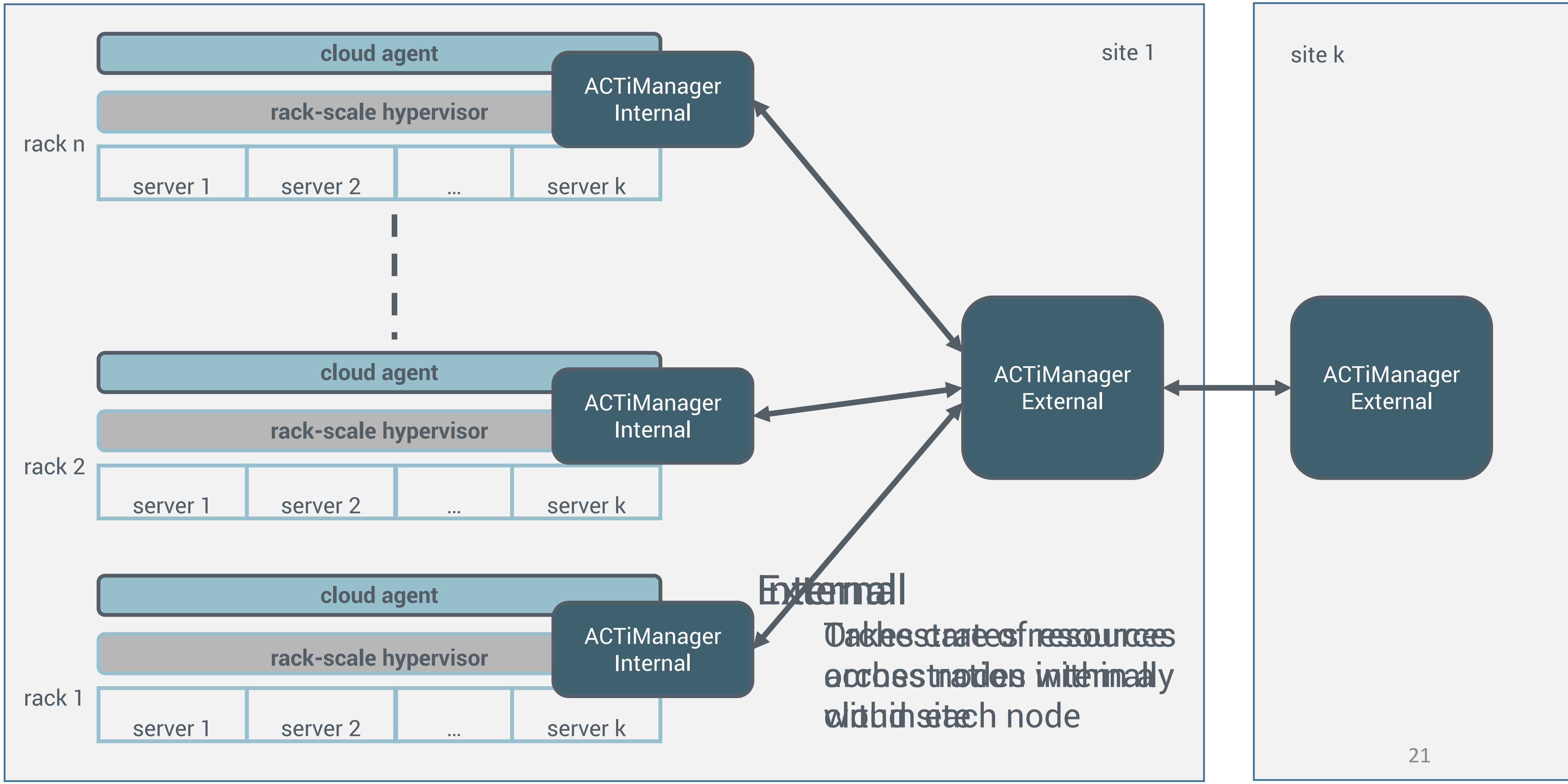
4. Cloud site administration may distinguish VMs *[ASPLOS'14, ISCA'15]*
   - High-priority, latency-critical VMs (gold instances)
   - Low-priority, batch VMs (silver instances)

5. On-line characterization of the VMs *[MICRO'11, ASPLOS'13, ATC'13]*
   - Potential to suffer from or create interference (noisy/sensitive)

6. Logical split of infrastructure *[ASPLOS'12, ATC'13]*
   - Laboratory node(s): to characterize VMs when no interference occurs
   - Production nodes

7. Applications may expose the desired metric of interest *[ASPLOS'14]*
   - To maintain Quality of Service (QoS)
   - But this is an optional feature

ACTiManager – Modules



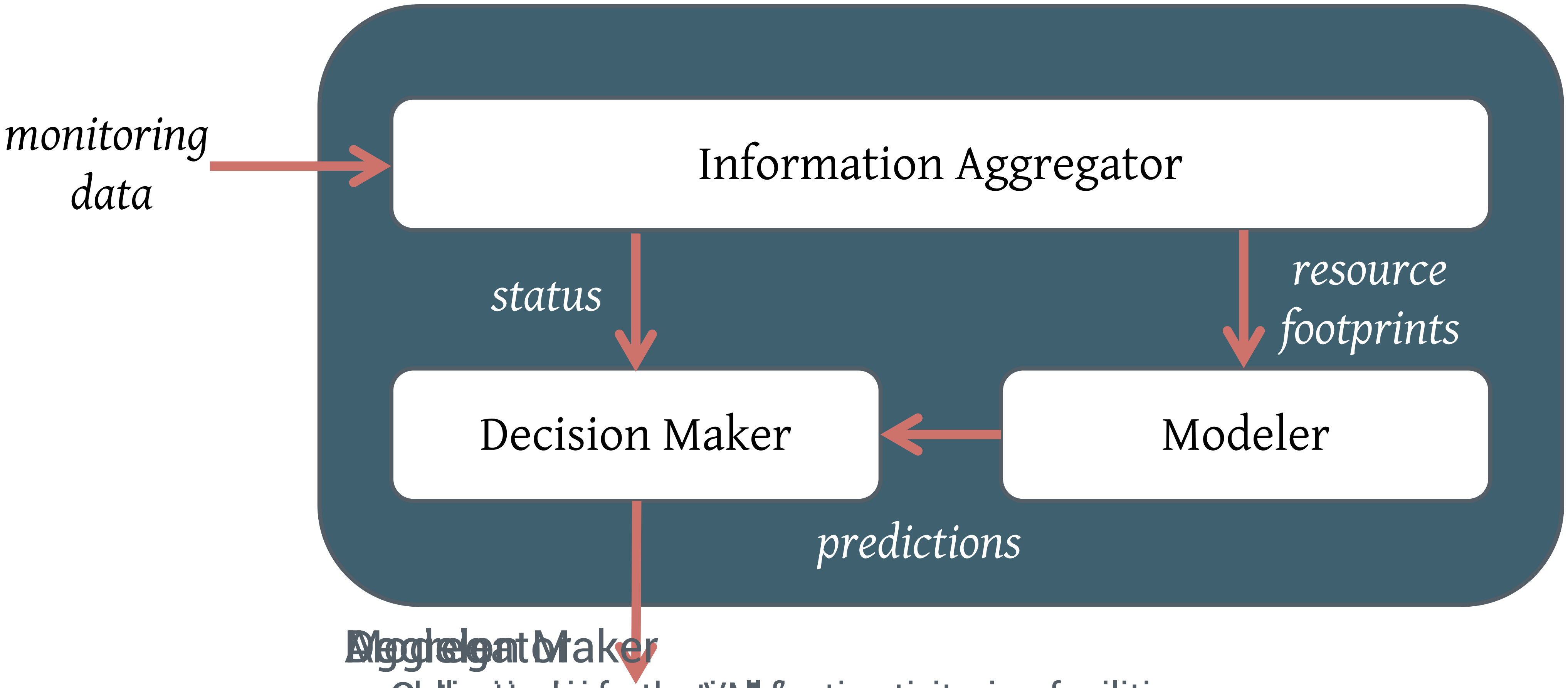
External

Takes care of resources across nodes within cloud site

Internal

Orchestrates resources internally within each node

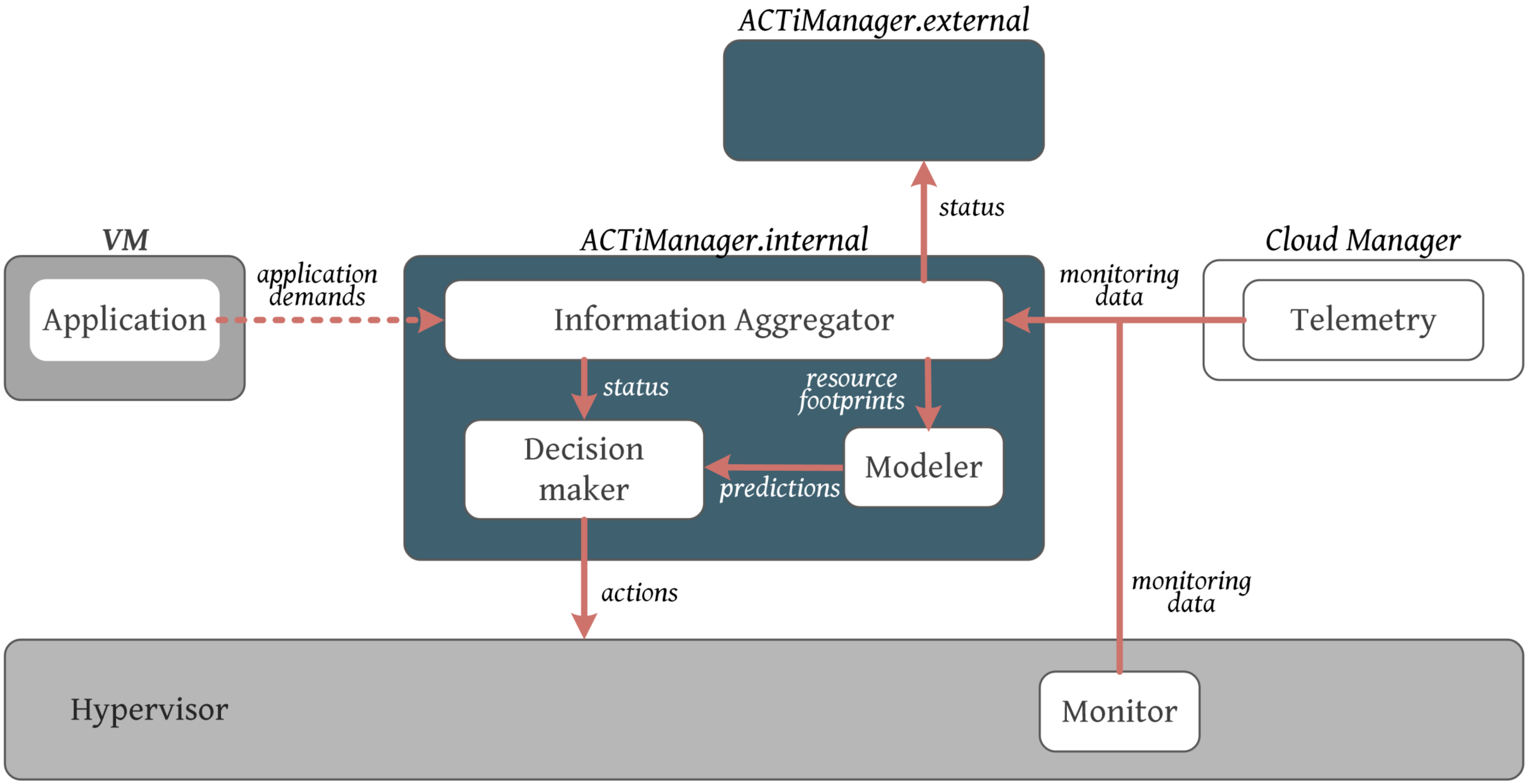
ACTiManager – Components



Aggregator / Modeler / Decision Maker

- [illegible] monitoring facilities
- Detect anomalies (interference, imbalance, overload, underload, etc.)
- [illegible]
    - Placement, prioritization, consolidation, migration, interference mitigation, VM resizing

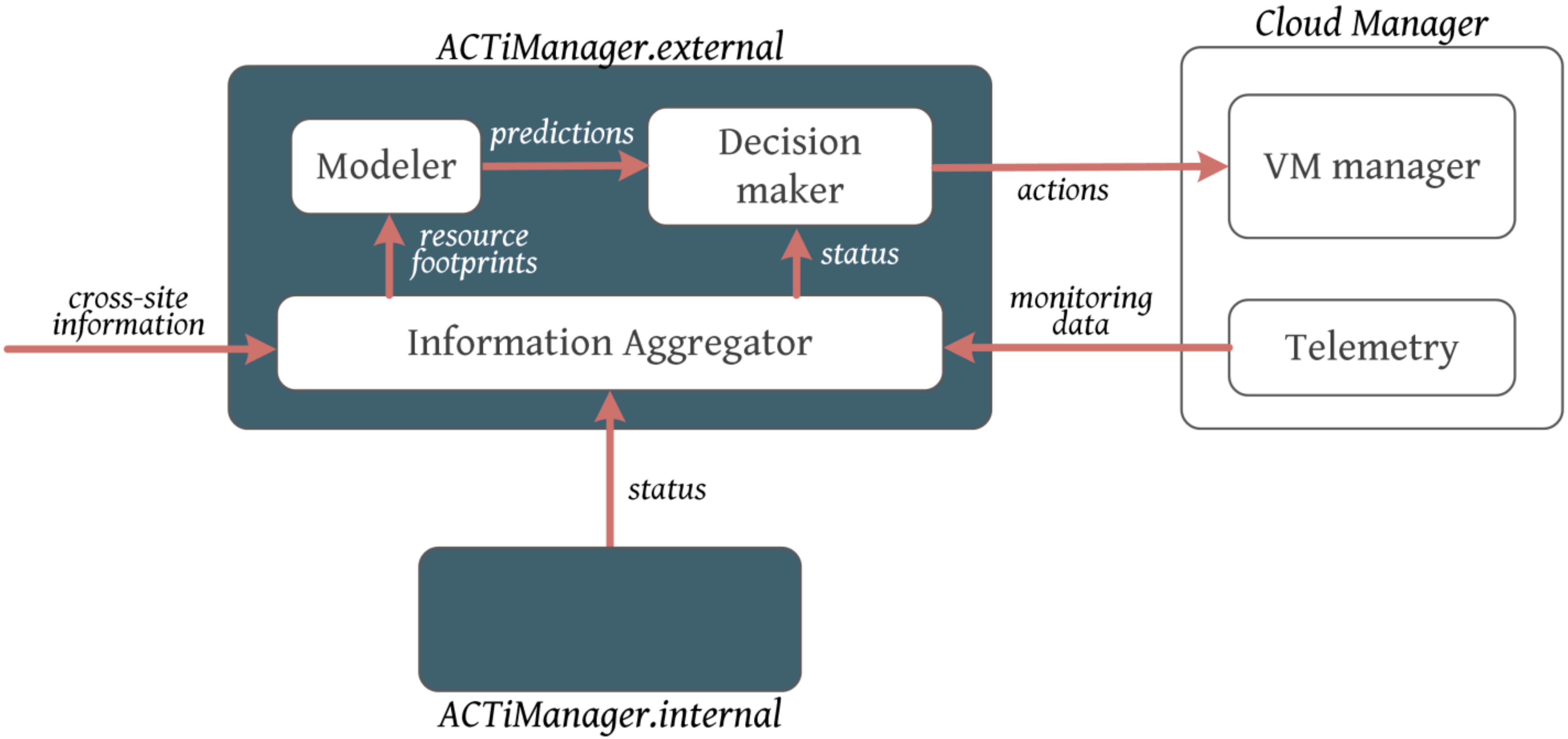
ACTiManager.Internal @ node level



ACTiManager.Internal @ node level

- Events
  - Change in total execution workload
  - Node over/under-utilization
  - Performance interference
- Actions
  - VM re-mapping (CPUs, memory)
  - Capping
  - Report interference
  - Report under/over-utilization
- Models
  - Characterization model
  - Interference model
  - Re-mapping cost model
  - Node over/under-utilization model

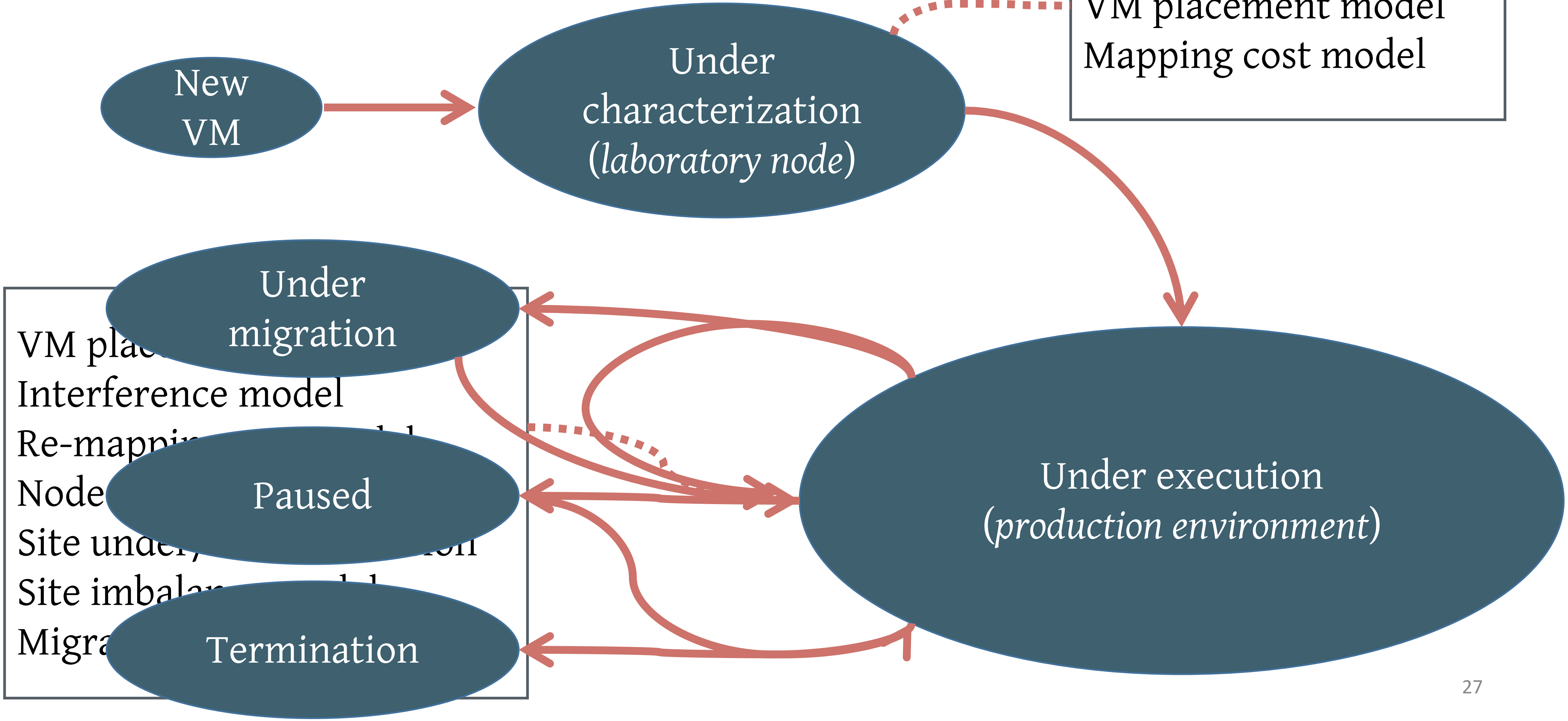
ACTiManager.External @ site level



ACTiManager.External @ site level

- Events
    - Creation of a new VM
    - Placement of a VM
    - Site under/over-utilization & imbalance
    - Notification from an Internal instance about interference
    - Notification from remote sites
    - Application under/over-performance
- Actions
    - Migration
    - VM control (start/stop) & scaling
    - Notification to remote sites
- Models
    - Placement model
    - Under/over-utilization & imbalance models
    - Migration cost model

ACTiManager – Lifecycle of a VM

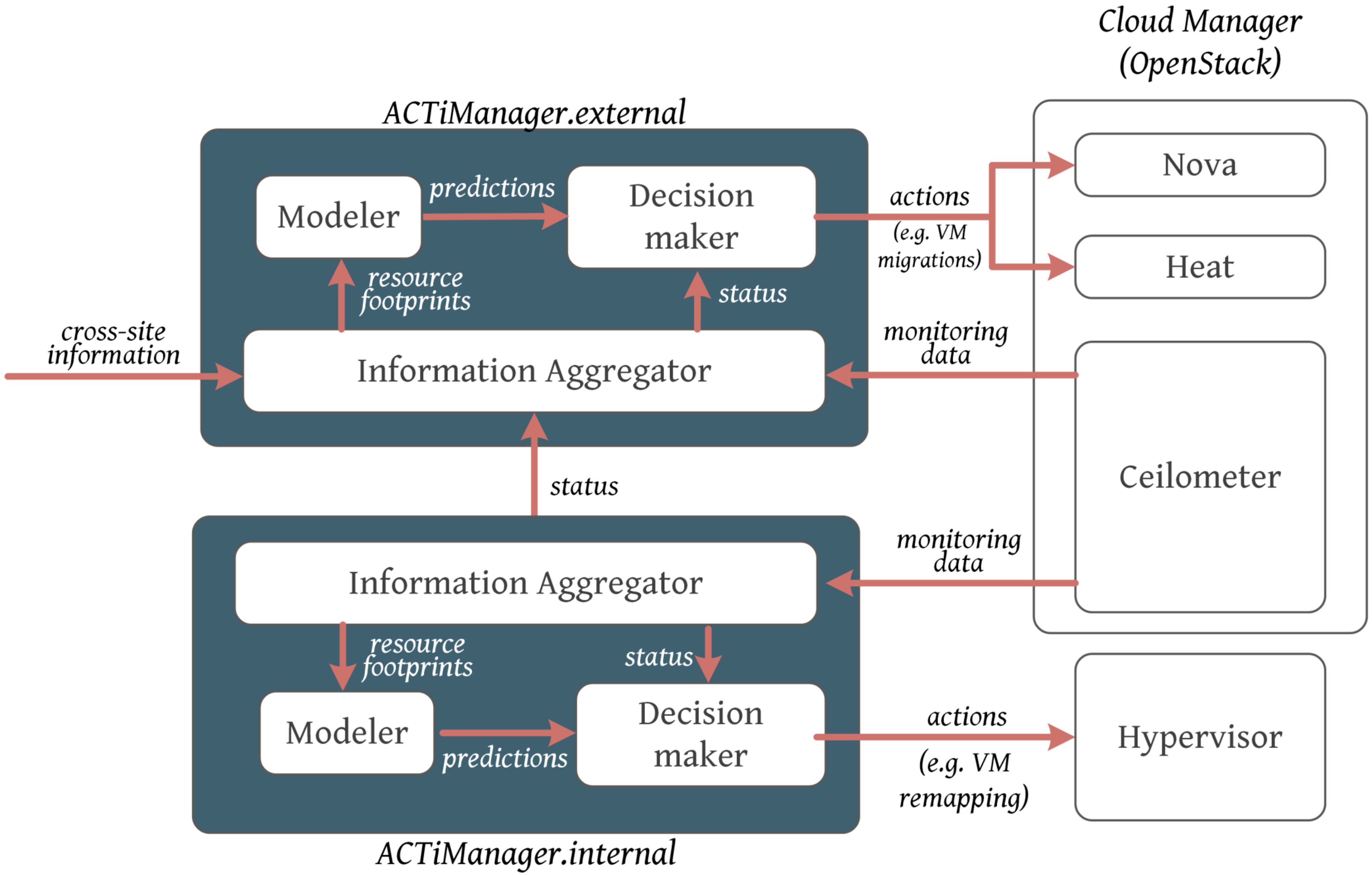


Characterization model
VM placement model
Mapping cost model

New VM
Under characterization (*laboratory node*)
Under migration
Paused
Termination
Under execution (*production environment*)

VM pla
Interference model
Re-mappi
Node
Site under ion
Site imbala
Migra

ACTiManager and OpenStack



Summary

- ACTiManager plays a key role in realizing the project’s objectives and goals
- Current implementation puts together all functional blocks
- On-going work focuses on the modeling and prediction parts
- Stay tuned

Questions?

ACTiCLOUD

More info

- Web: acticloud.eu
- News: twitter.com/acticloud
- Code: github.com/acticloud

References

- Christina Delimitrou and Christos Kozyrakis. Paragon: QoS-aware Scheduling for Heterogeneous Datacenters. *ASPLOS'13*.
- Christina Delimitrou and Christos Kozyrakis. Quasar: Resource-efficient and QoS-aware Cluster Management. *ASPLOS'14*.
- Georgios Goumas *et al.* ACTiCLOUD: Enabling the Next Generation of Cloud Applications. *ICDCS'17*.
- David Lo, Liqun Cheng, Rama Govindaraju, Parthasarathy Ranganathan, and Christos Kozyrakis. Heracles: Improving Resource Efficiency at Scale. *ISCA'15*.
- Jason Mars, Lingjia Tang, Robert Hundt, Kevin Skadron, and Mary Lou Soffa. Bubble-Up: Increasing Utilization in Modern Warehouse Scale Computers via Sensible Co-locations. *MICRO'11*.
- Dejan Novaković, Nedeljko Vasić, Stanko Novaković, Dejan Kostić, and Ricardo Bianchini. DeepDive: Transparently Identifying and Managing Performance Interference in Virtualized Environments. *USENIX ATC'13*.
- Nedeljko Vasić, Dejan Novaković, Svetozar Miučin, Dejan Kostić, and Ricardo Bianchini. 2012. DejaVu: Accelerating Resource Allocation in Virtualized Environments. *ASPLOS'12*.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement no 732366 (ACTiCLOUD)