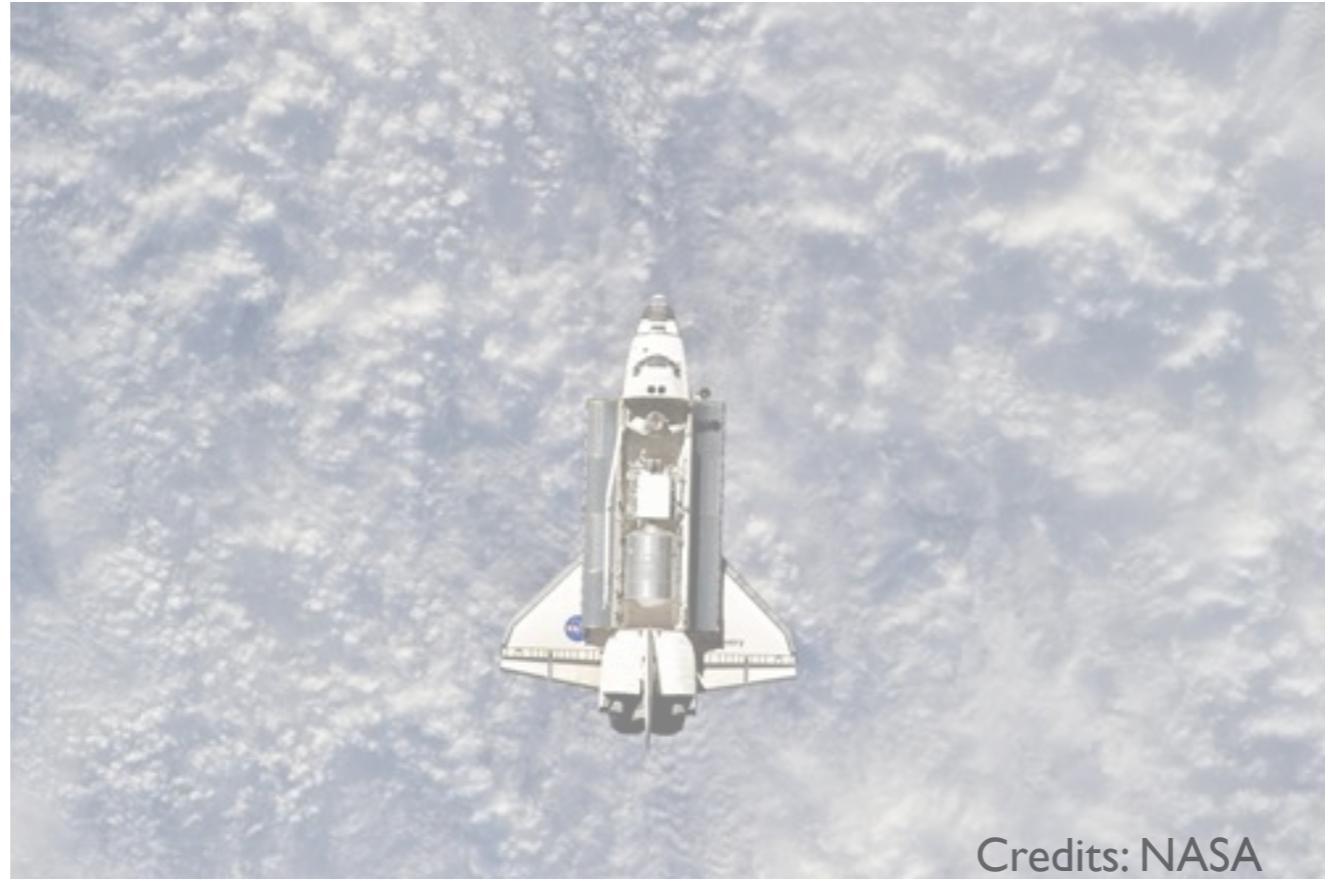


Beyond the Clouds, the DISCOVERY Initiative



Credits: NASA

Localization is a key element to deliver
efficient as well as **sustainable Utility Computing Solutions**



Adrien Lèbre / Ascola Project Team
November, 2013



Context

xxx Computing

- Meta / Cluster / Grid / Desktop / Cloud / Sky / Fog ...
- A common objective: provide computing resources (both hardware and software) in a flexible, transparent, secure, reliable, ... way

⇒ xxx as Utility Computing (UC)

new concept, but rather has quite a long history. Among the earliest references is:

“ If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility... The computer utility could become the basis of a new and important industry.”

—John McCarthy, speaking at the MIT Centennial in 1961 [2]

Utility Computing

- Successive generations

- Mainframes (time-sharing, database - 1980 / 20xx)

- Network of workstations (clusters)) (1990 / 20xx)
 - Grid (clusters federation))

- Cloud Computing (SaaS/PaaS/**IaaS** - 2005 / 20xx)

- Challenges

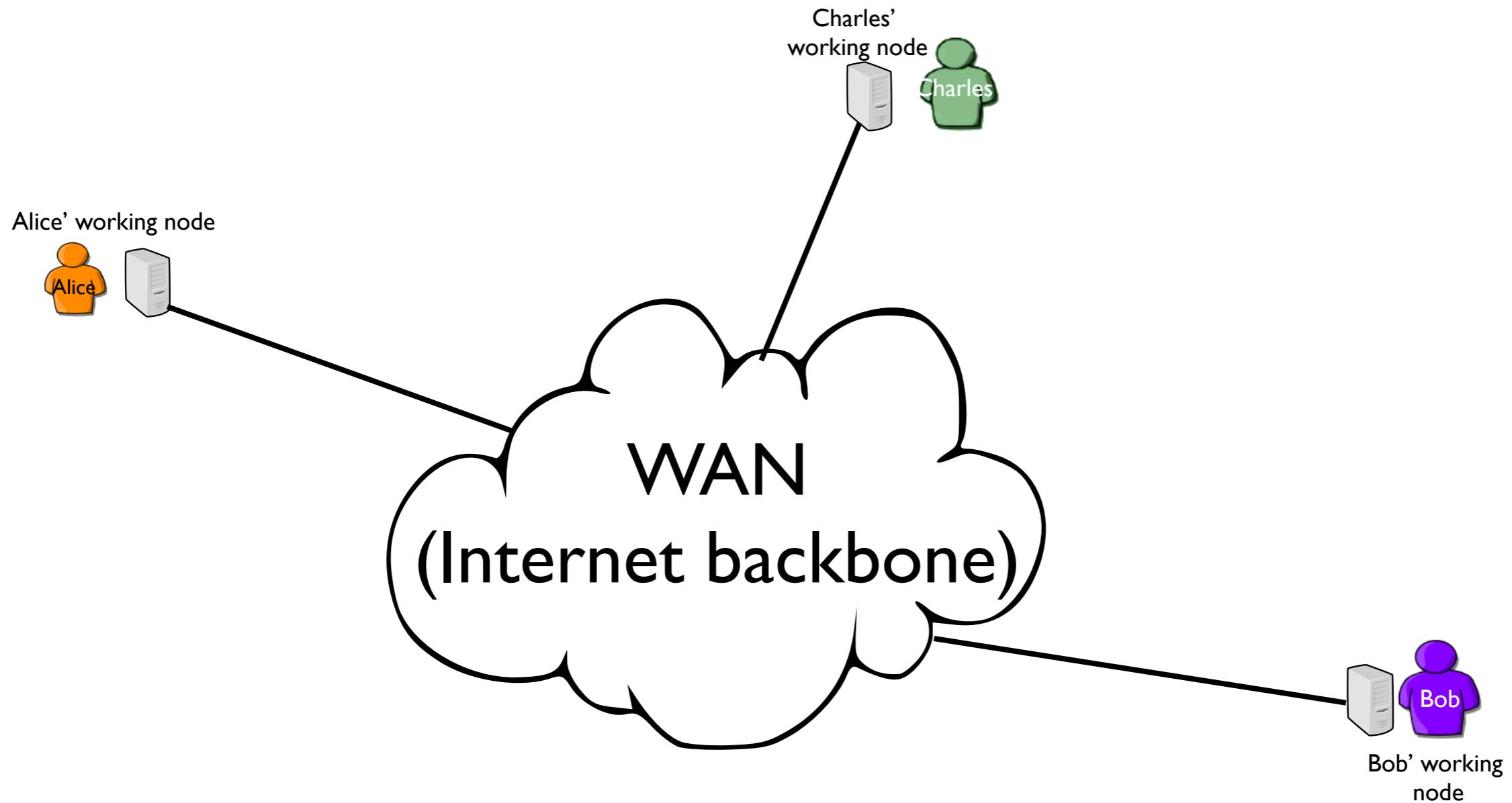
- Data Sharing

- Software/Hardware heterogeneity

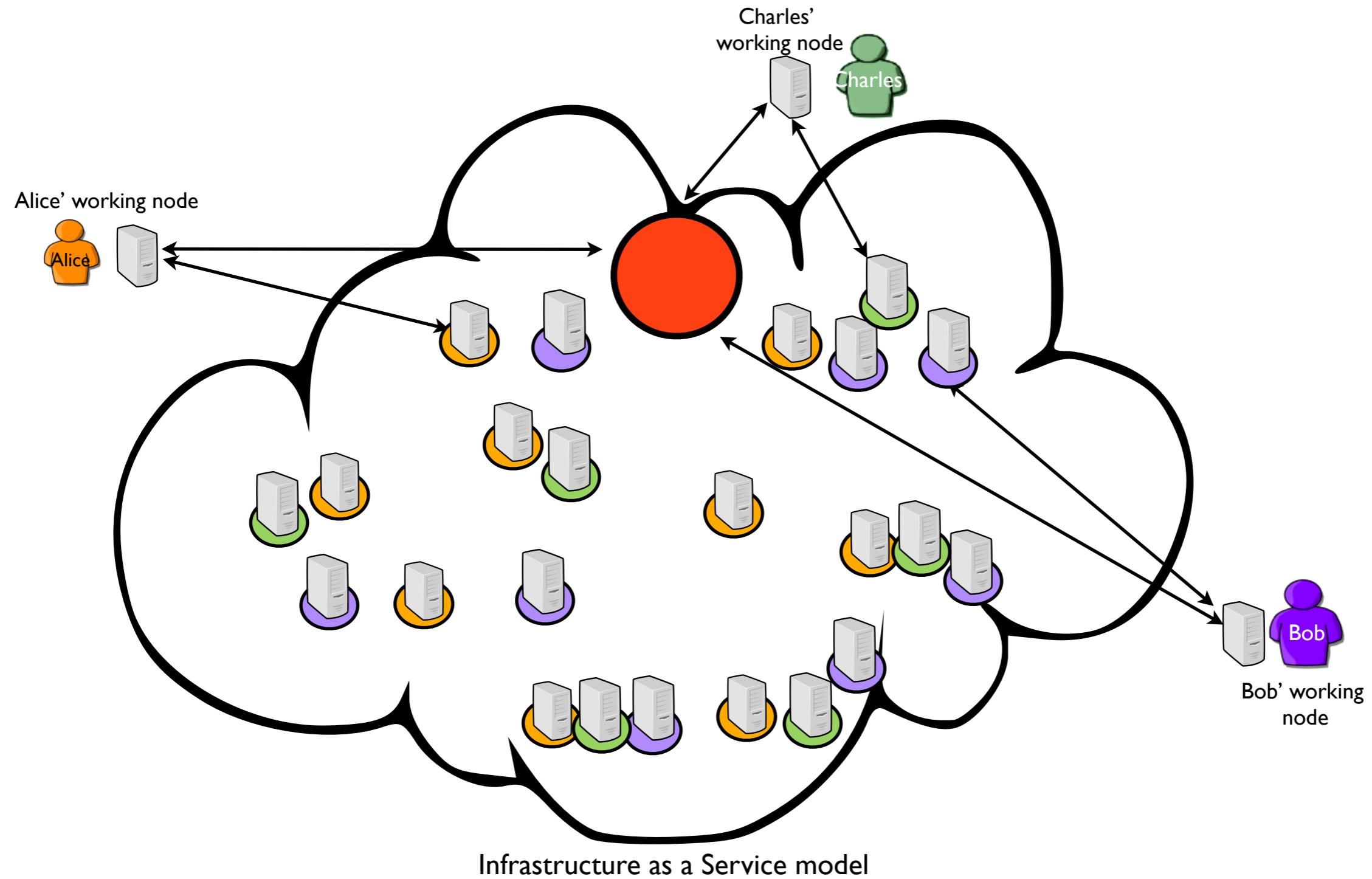
- Security (Isolation between applications, ...)

- Reachability / Reliability / Resiliency ...

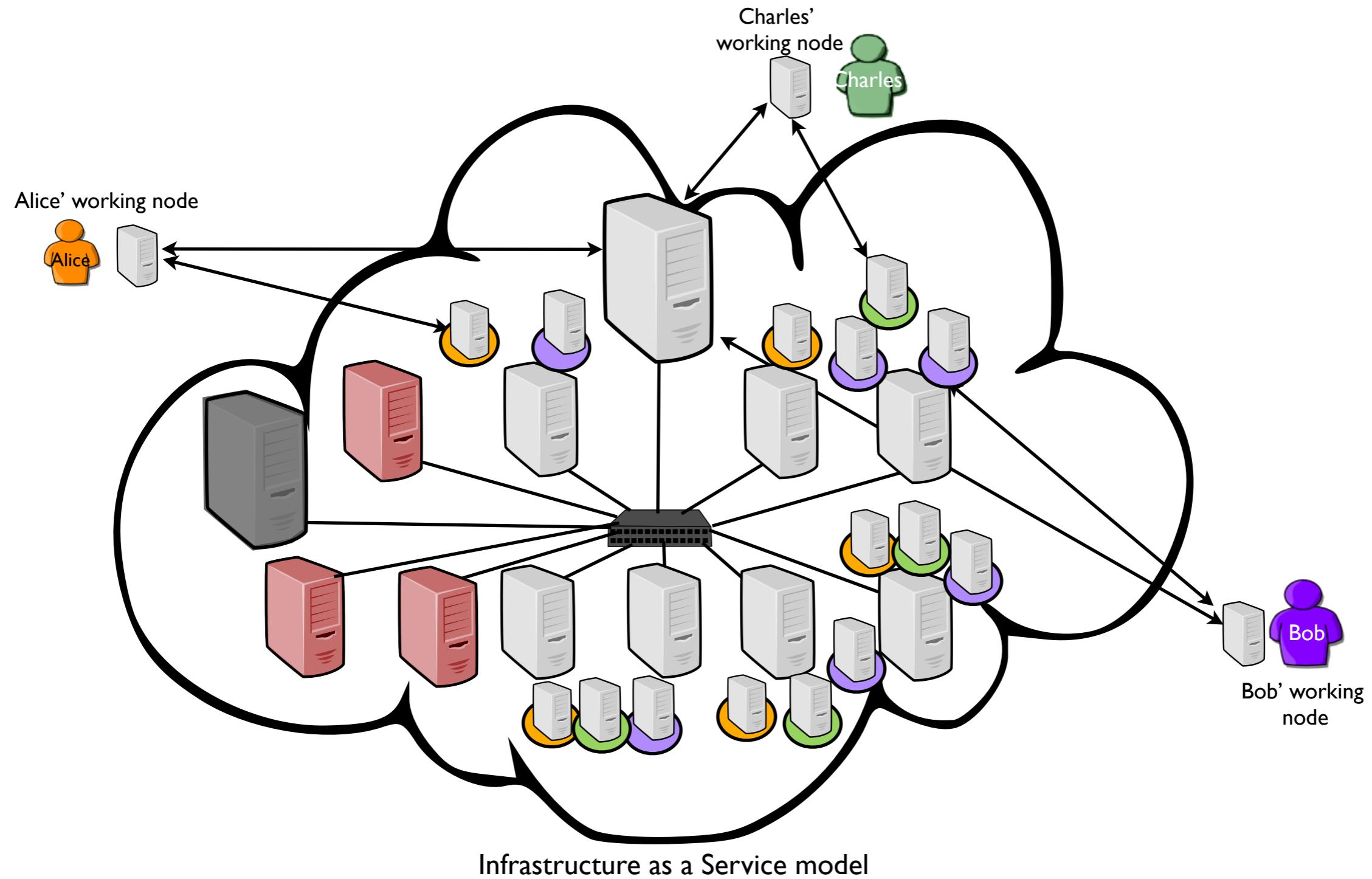
Utility Computing - The Cloud



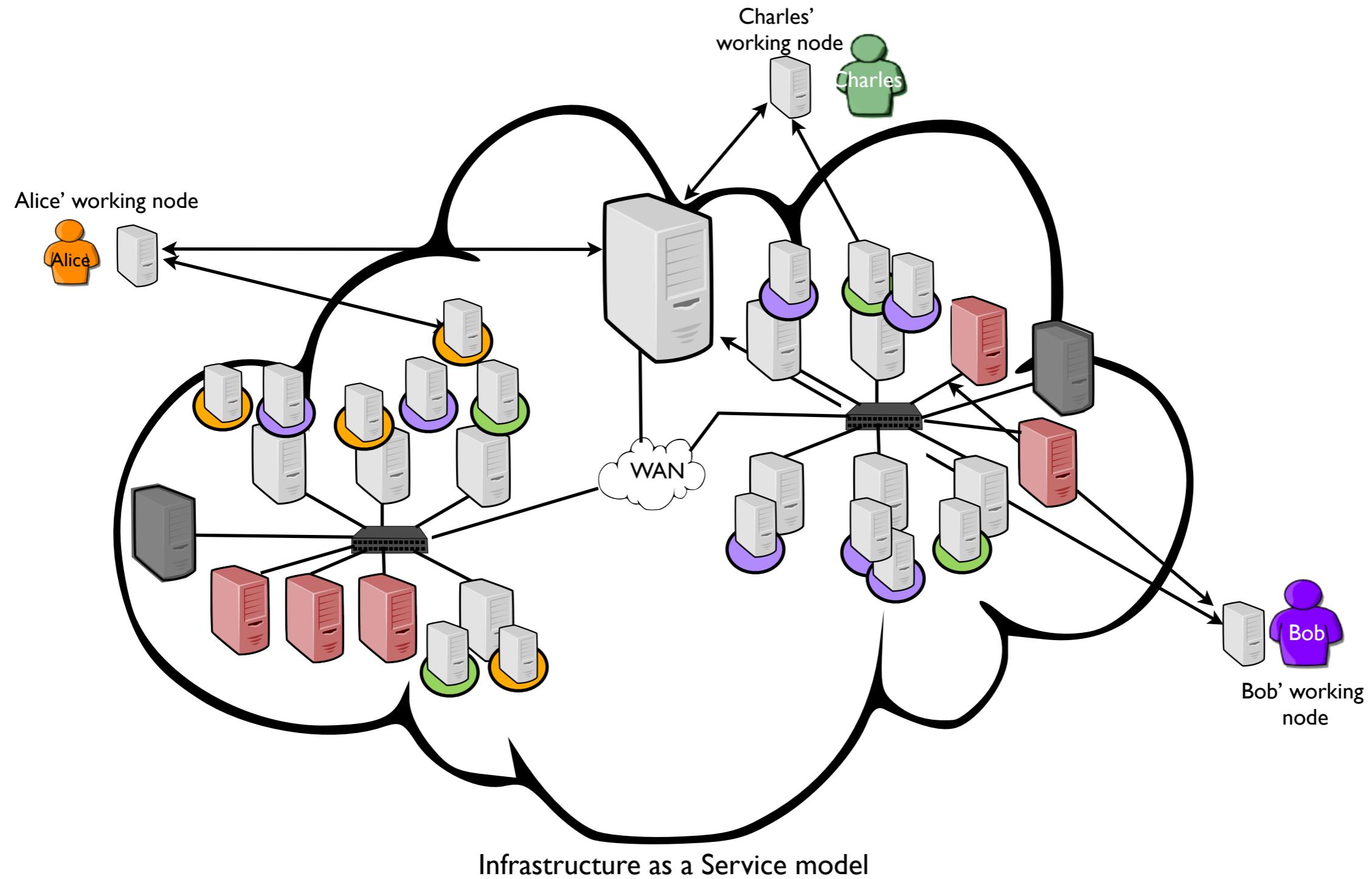
Utility Computing - The Cloud



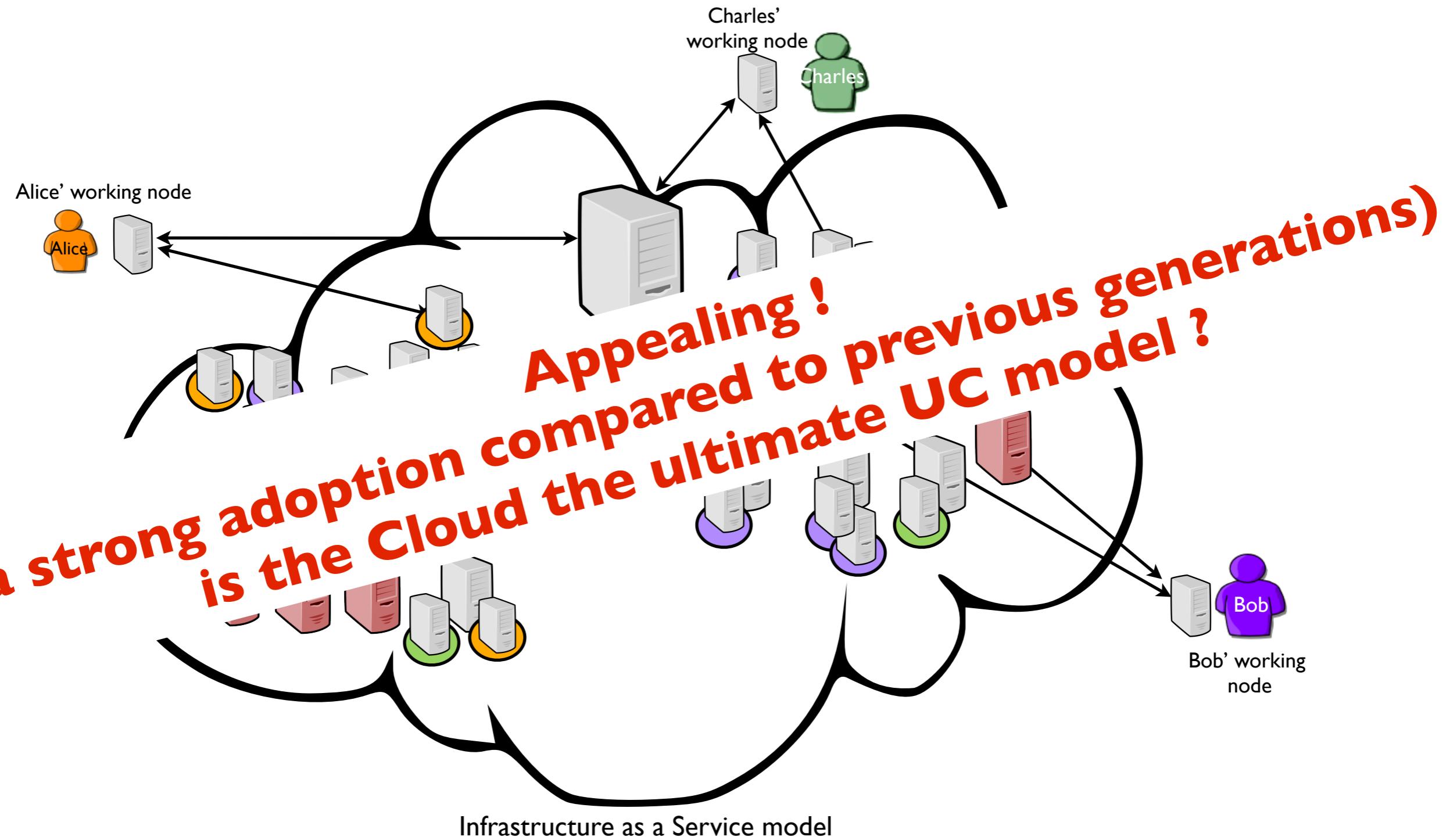
Utility Computing - The Cloud



Utility Computing - The Cloud



Utility Computing - The Cloud



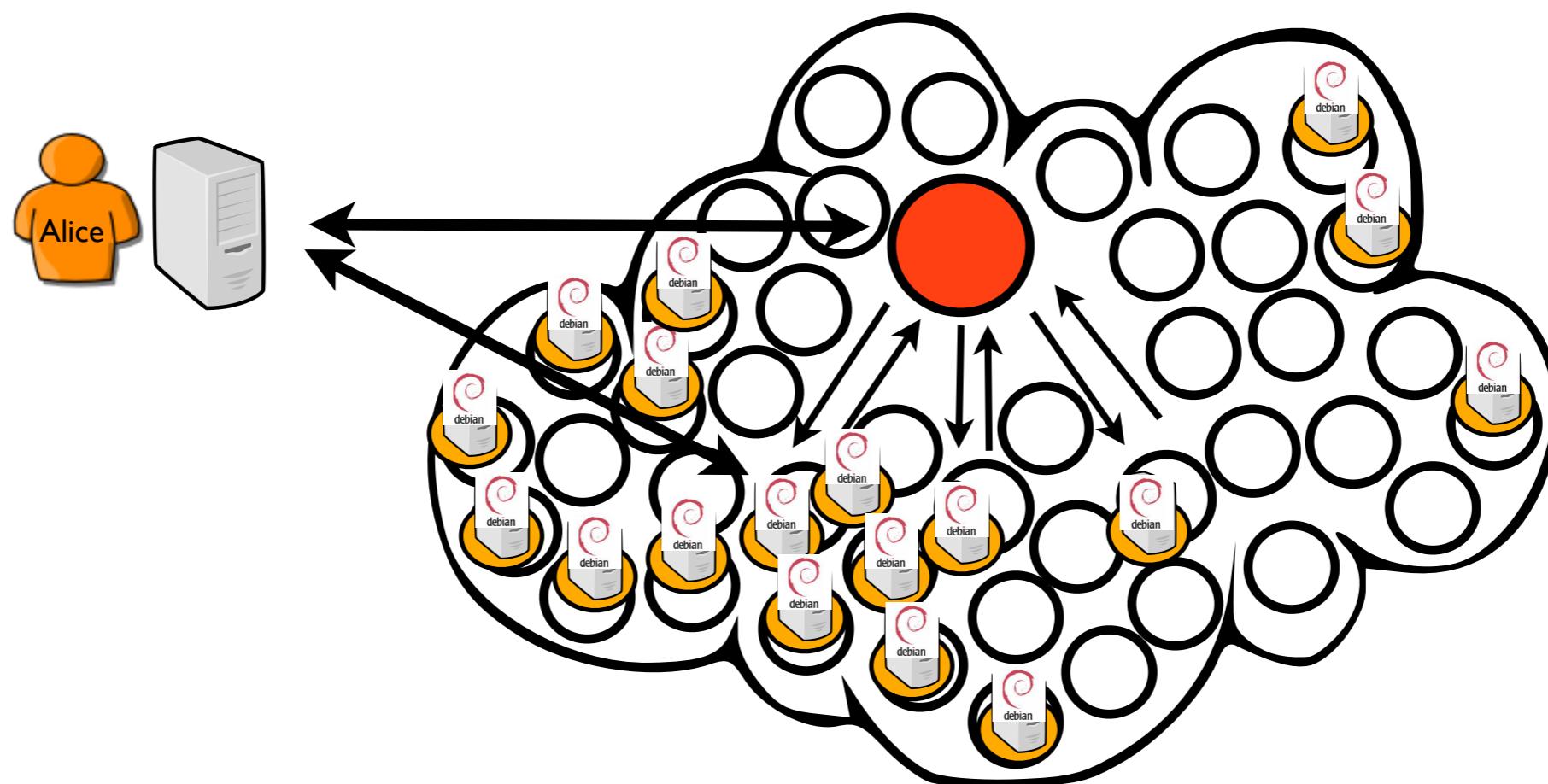
The Cloud the ultimate UC model ...

- Mature for one site/cloud !

Open Nebula, Nimbus... vSphere... CloudStack, OpenStack
More flexibility ! ? Infinite resources ! ?

- Current concerns

Scalability (VM Sprawl)



6 / 67

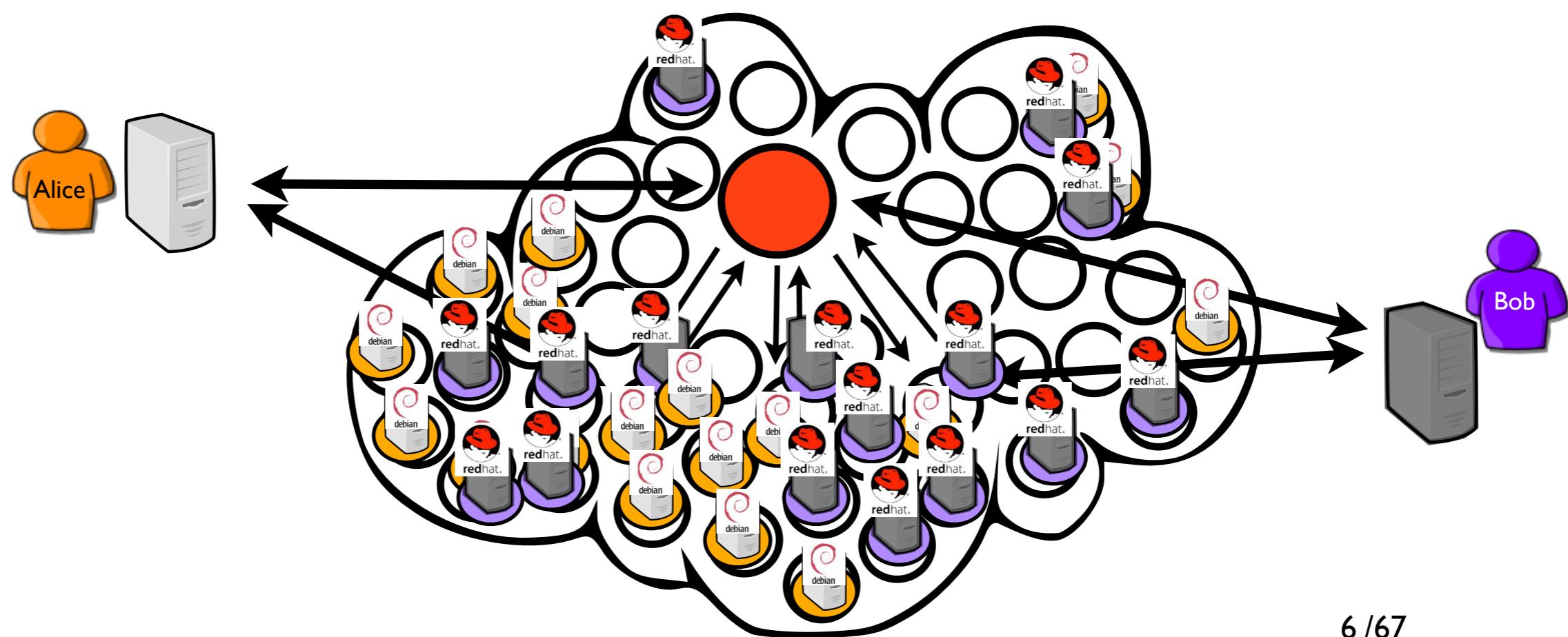
The Cloud the ultimate UC model ...

- Mature for one site/cloud !

Open Nebula, Nimbus... vSphere... CloudStack, OpenStack
More flexibility ! ? Infinite resources ! ?

- Current concerns

Scalability (VM Sprawl)



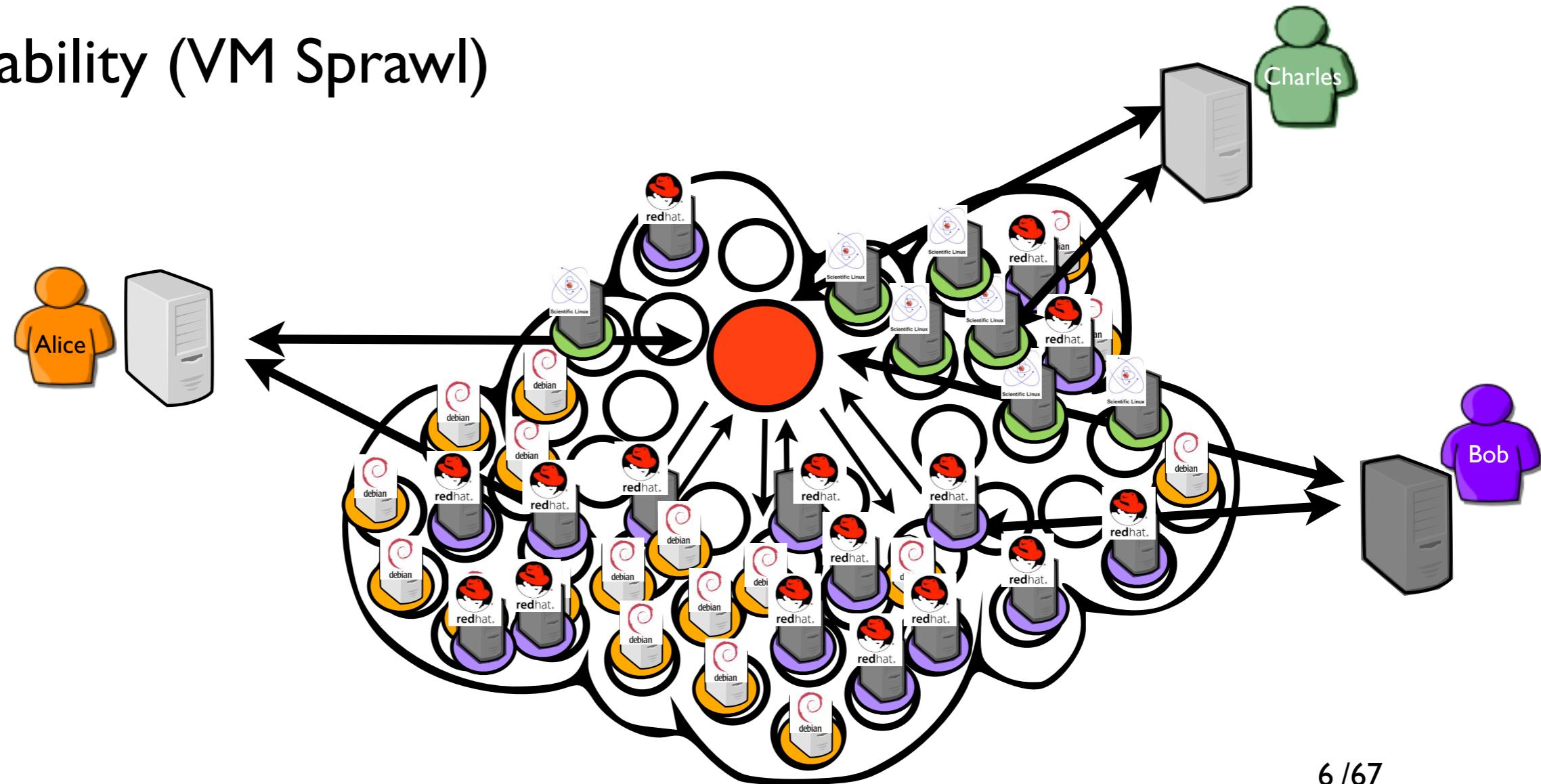
The Cloud the ultimate UC model ...

- Mature for one site/cloud !

Open Nebula, Nimbus... vSphere... CloudStack, OpenStack
More flexibility ! ? Infinite resources ! ?

- Current concerns

Scalability (VM Sprawl)



The Cloud the ultimate UC model ...

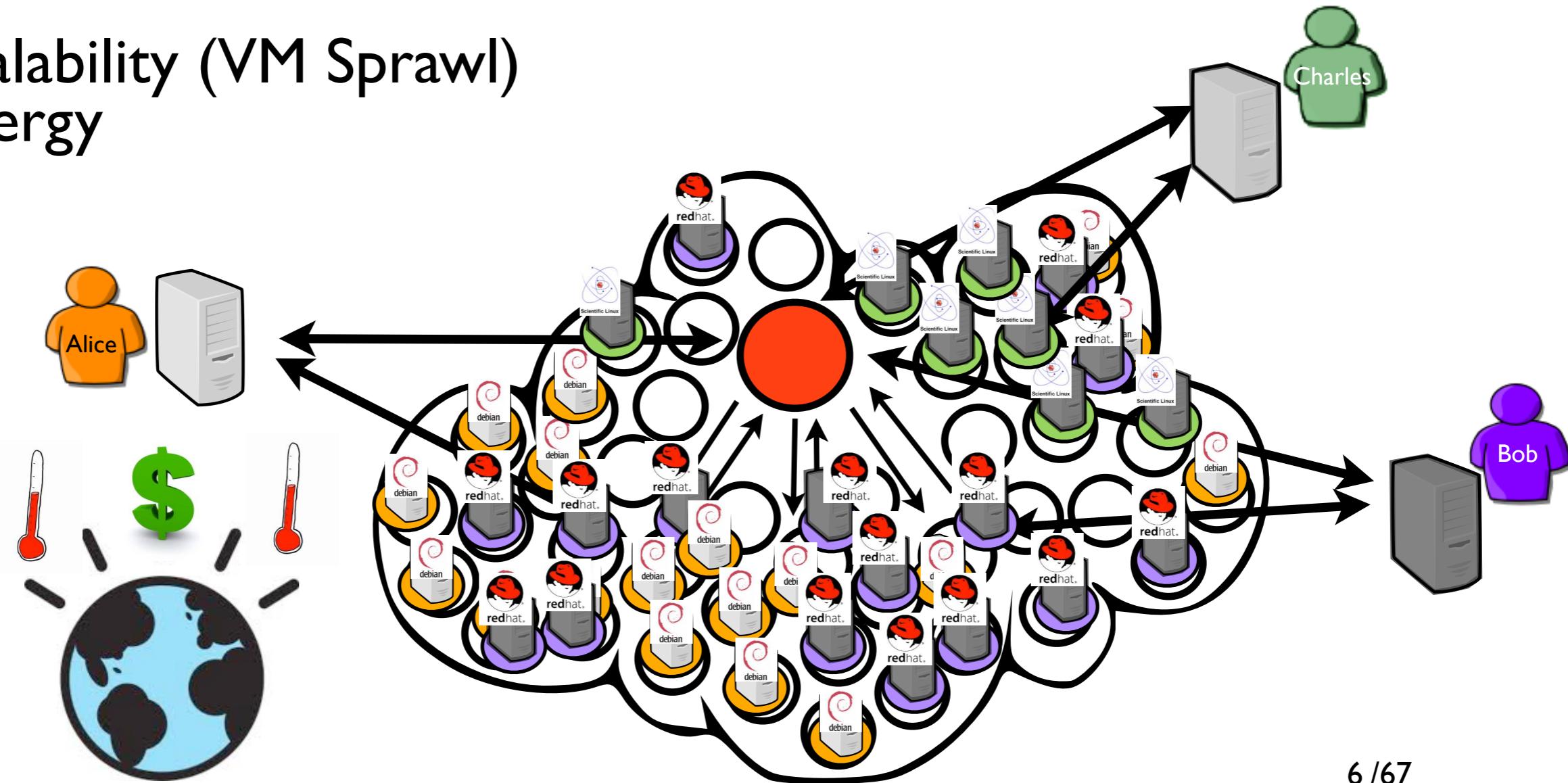
- Mature for one site/cloud !

Open Nebula, Nimbus... vSphere... CloudStack, OpenStack
More flexibility ! ? Infinite resources ! ?

- Current concerns

Scalability (VM Sprawl)

Energy



The Cloud the ultimate UC model ...

- Mature for one site/cloud !

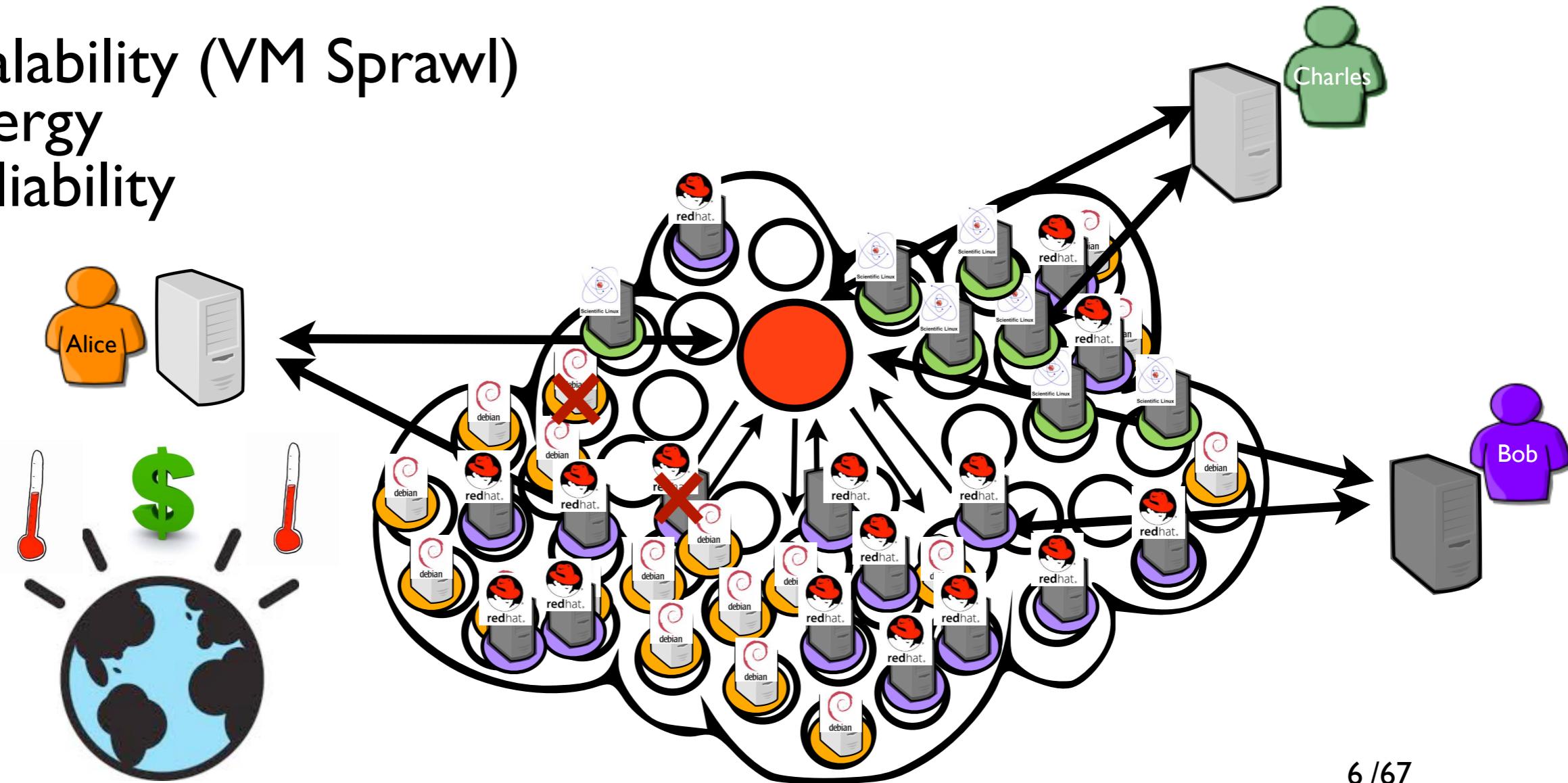
Open Nebula, Nimbus... vSphere... CloudStack, OpenStack
More flexibility ! ? Infinite resources ! ?

- Current concerns

Scalability (VM Sprawl)

Energy

Reliability



The Cloud the ultimate UC model ...

- Mature for one site/cloud !

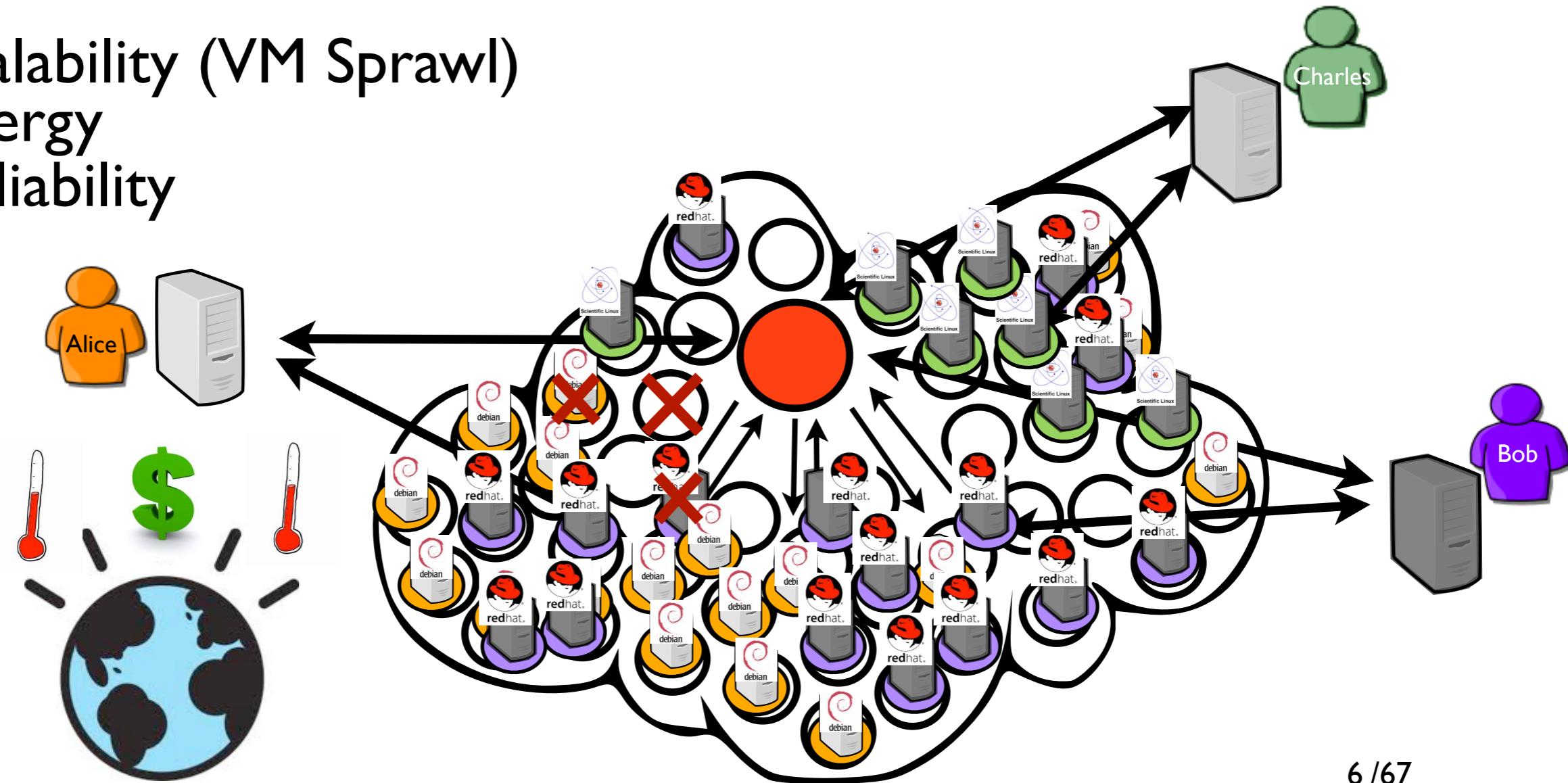
Open Nebula, Nimbus... vSphere... CloudStack, OpenStack
More flexibility ! ? Infinite resources ! ?

- Current concerns

Scalability (VM Sprawl)

Energy

Reliability



The Cloud the ultimate UC model ...

- Mature for one site/cloud !

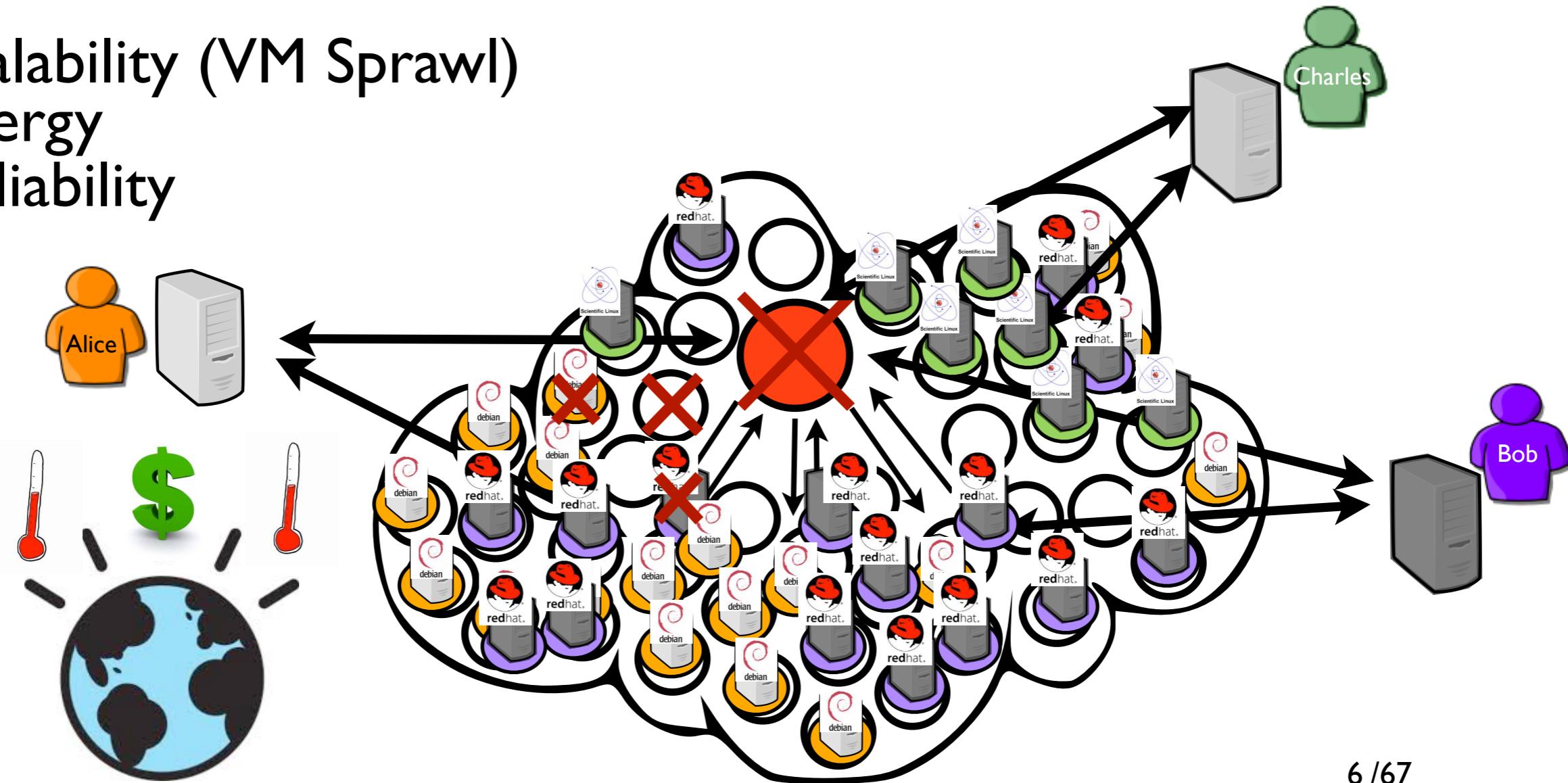
Open Nebula, Nimbus... vSphere... CloudStack, OpenStack
More flexibility ! ? Infinite resources ! ?

- Current concerns

Scalability (VM Sprawl)

Energy

Reliability



The Cloud the ultimate UC model ...

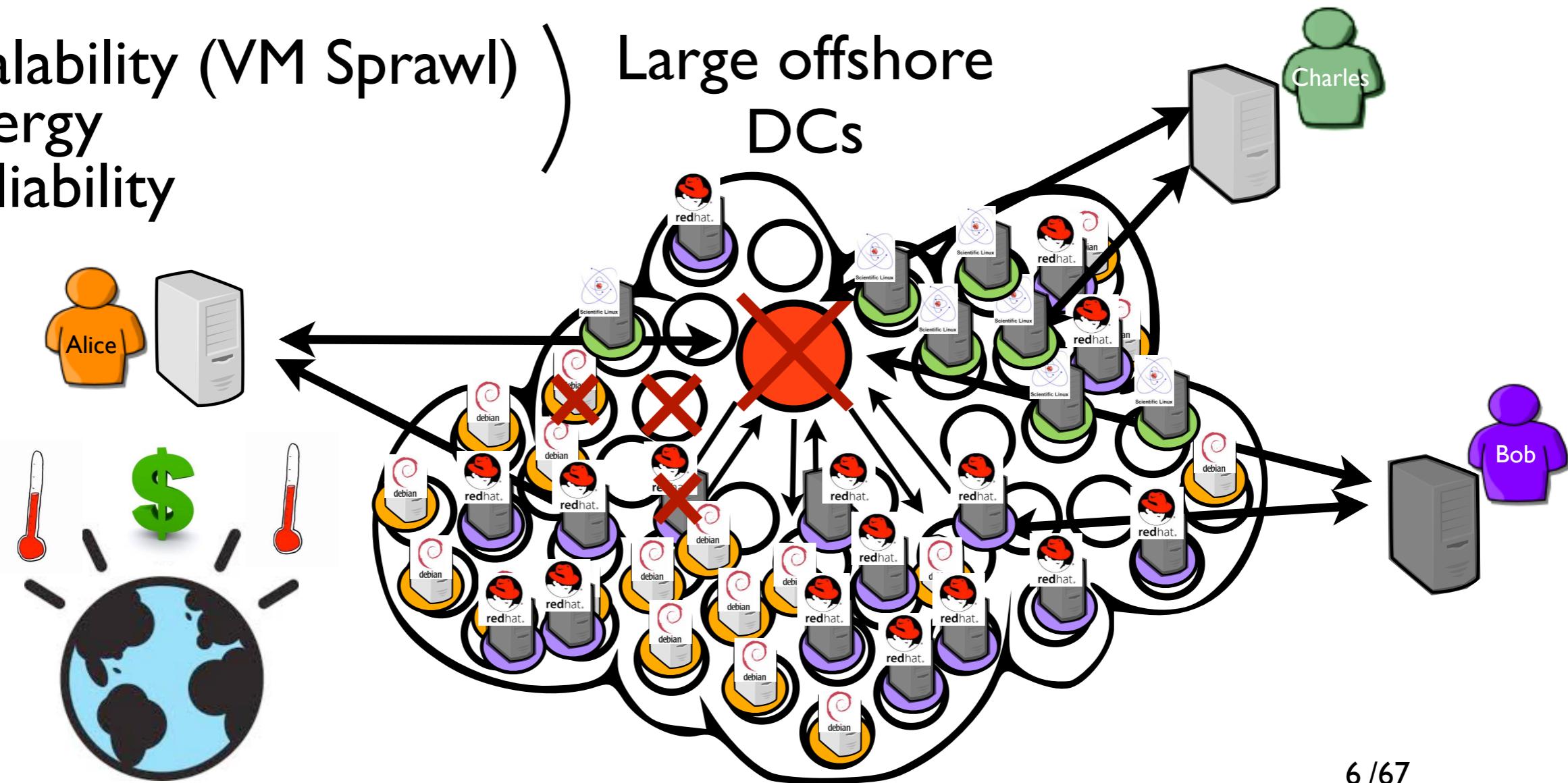
- Mature for one site/cloud !

Open Nebula, Nimbus... vSphere... CloudStack, OpenStack
More flexibility ! ? Infinite resources ! ?

- Current concerns

Scalability (VM Sprawl)
Energy
Reliability

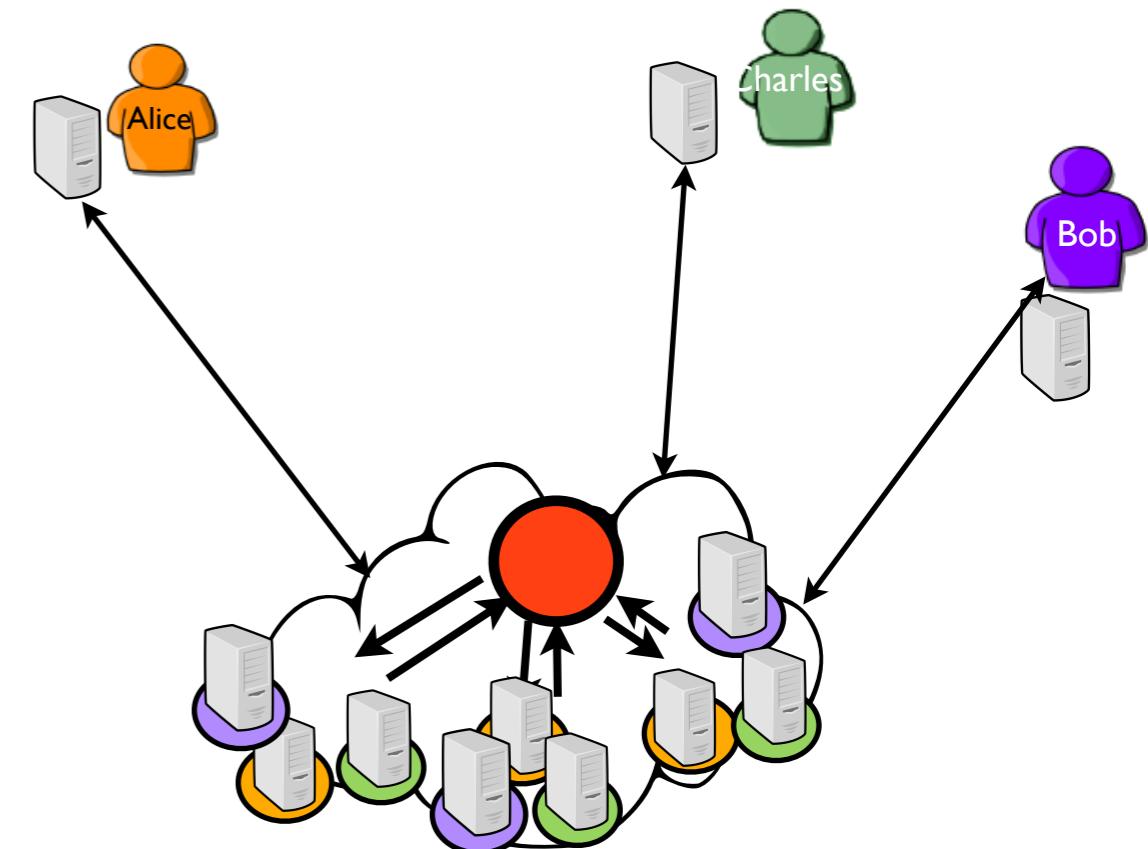
) Large offshore
DCs



... Not so sure !

- Inherent limitations of the cloud computing model w.r.t public offers (or why building large offshore DCs is not appropriated).

I. Externalization of private applications/
data (jurisdiction concerns)

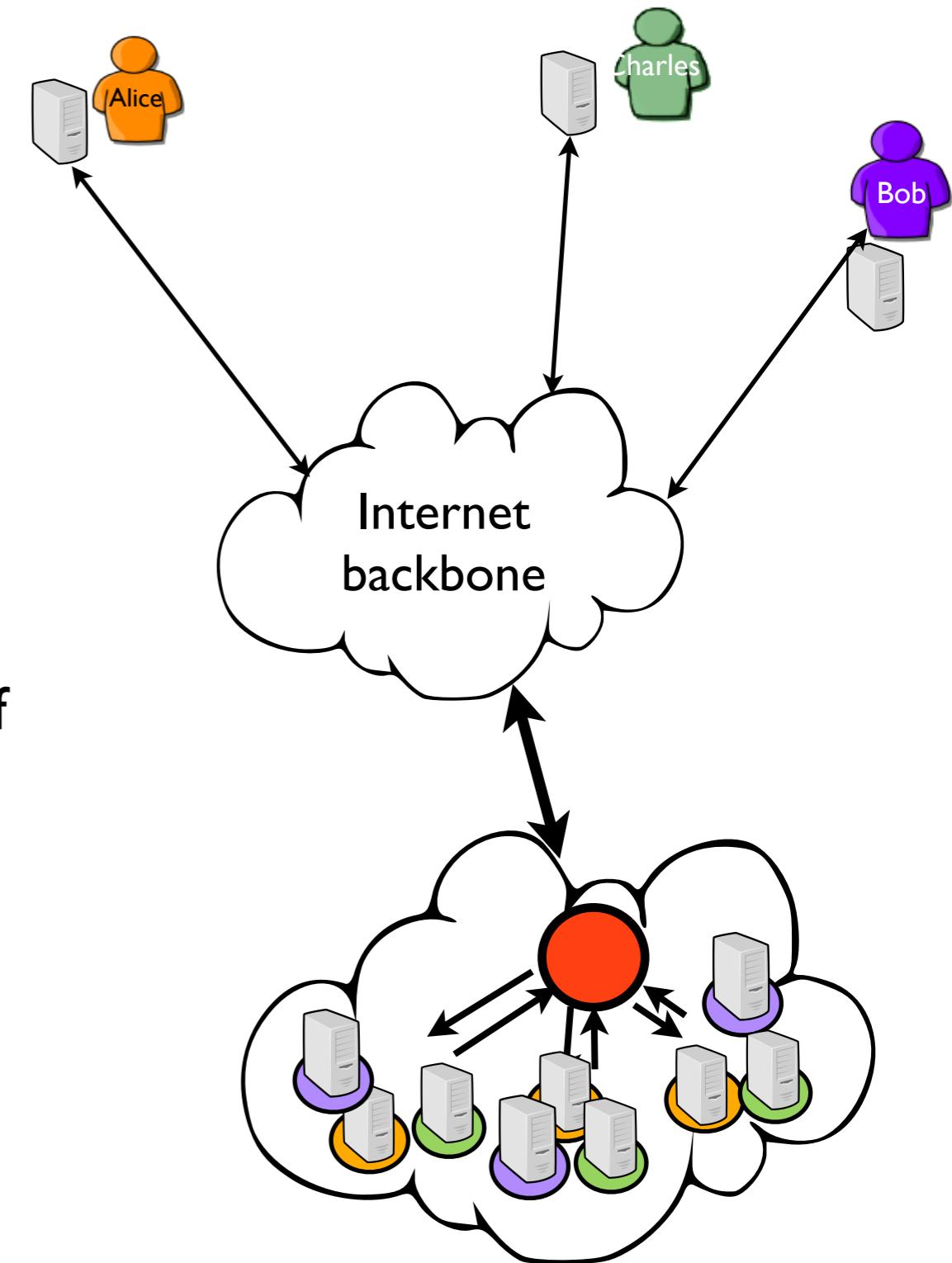


... Not so sure !

- Inherent limitations of the cloud computing model w.r.t public offers (or why building large offshore DCs is not appropriated).

1. Externalization of private applications/ data (jurisdiction concerns)

2. Overhead implied by the unavoidable use of the Internet to reach distant platforms



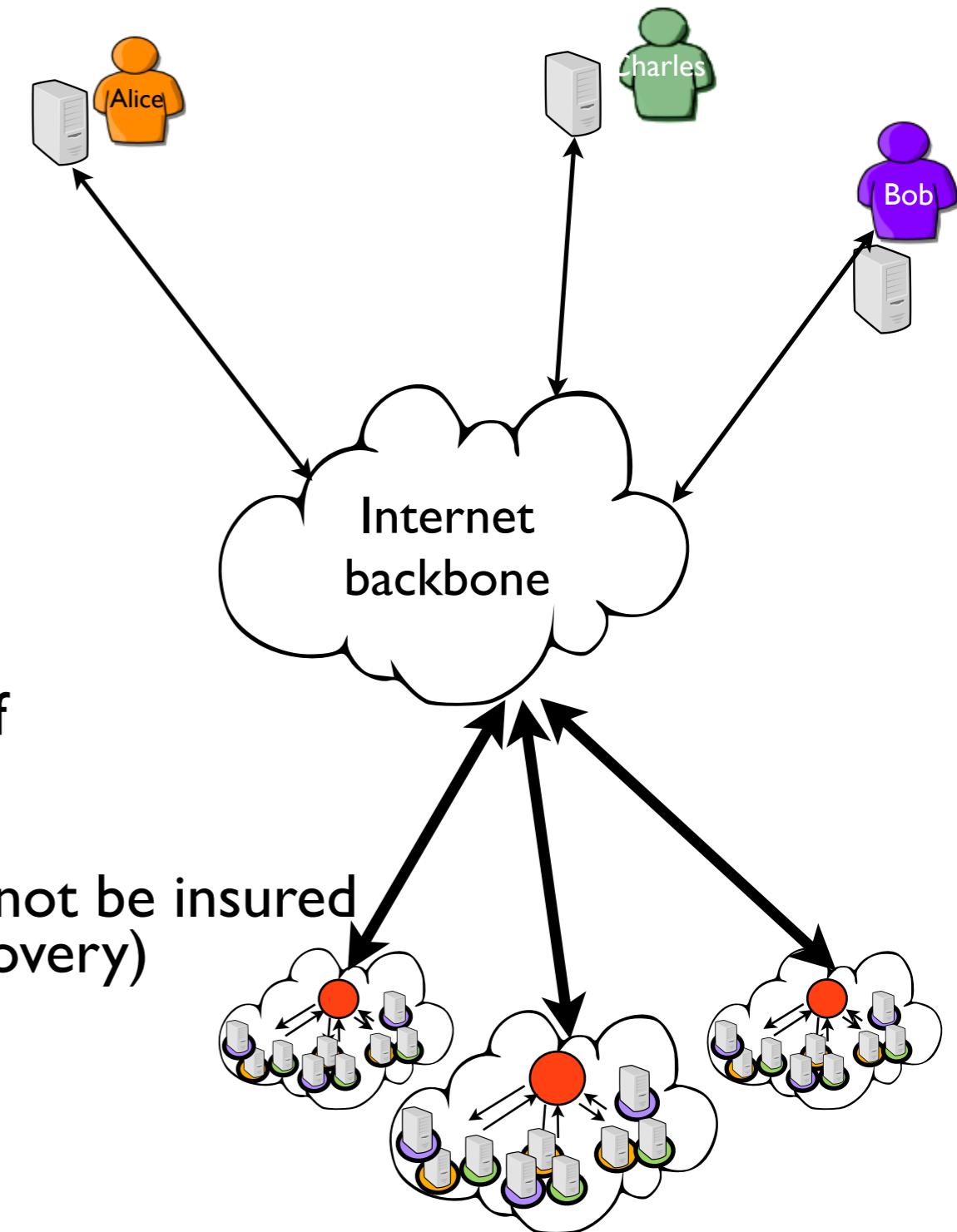
... Not so sure !

- Inherent limitations of the cloud computing model w.r.t public offers (or why building large offshore DCs is not appropriated).

- I. Externalization of private applications/ data (jurisdiction concerns)

2. Overhead implied by the unavoidable use of the Internet to reach distant platforms

3. The connectivity to the application/data cannot be insured by centralized dedicated centers (disaster recovery)



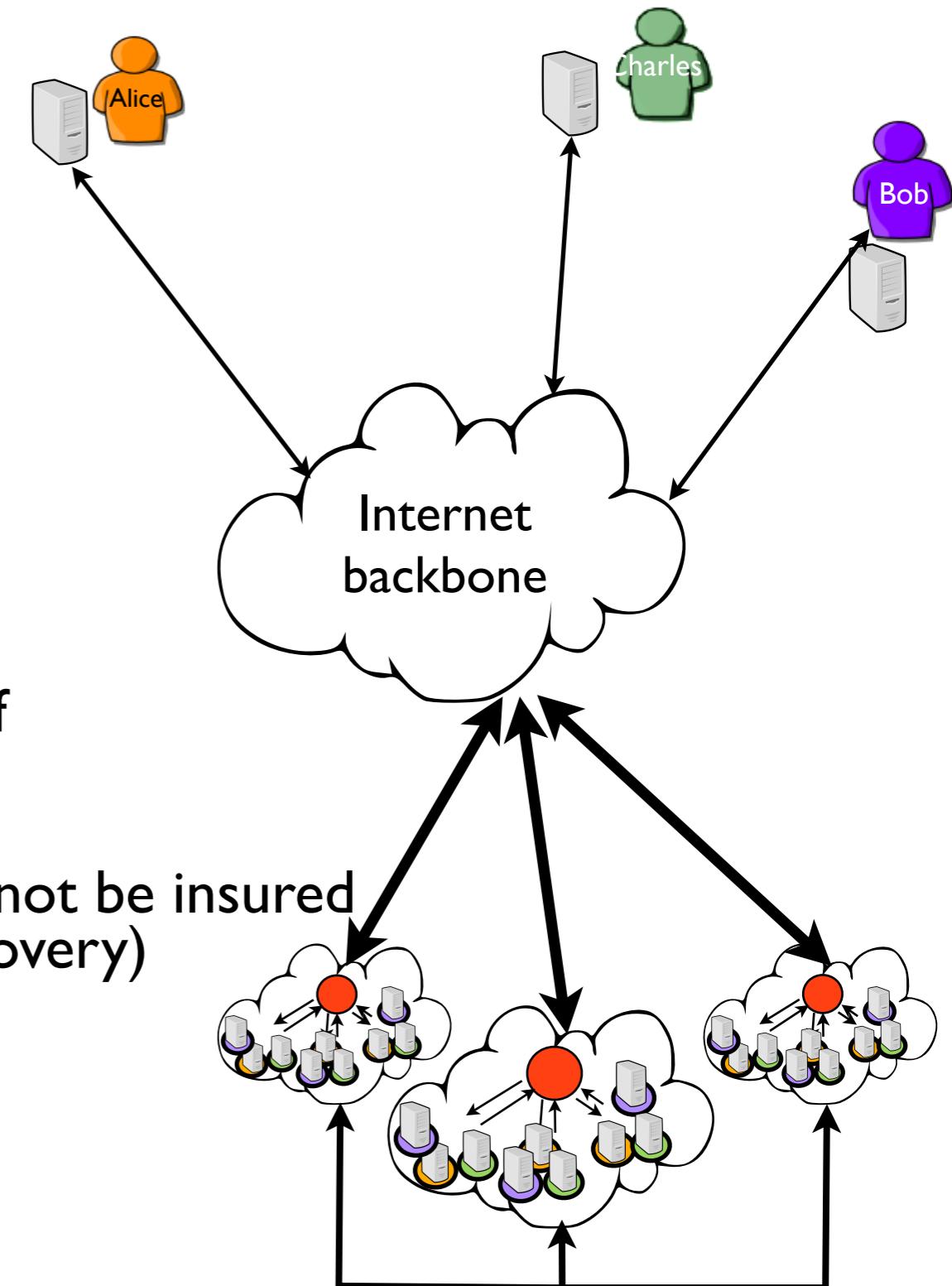
... Not so sure !

- Inherent limitations of the cloud computing model w.r.t public offers (or why building large offshore DCs is not appropriated).

1. Externalization of private applications/ data (jurisdiction concerns)

2. Overhead implied by the unavoidable use of the Internet to reach distant platforms

3. The connectivity to the application/data cannot be insured by centralized dedicated centers (disaster recovery)



... Not so sure !

- Inherent limitations of the cloud computing model w.r.t public offers (or why building large offshore DCs is not appropriated).

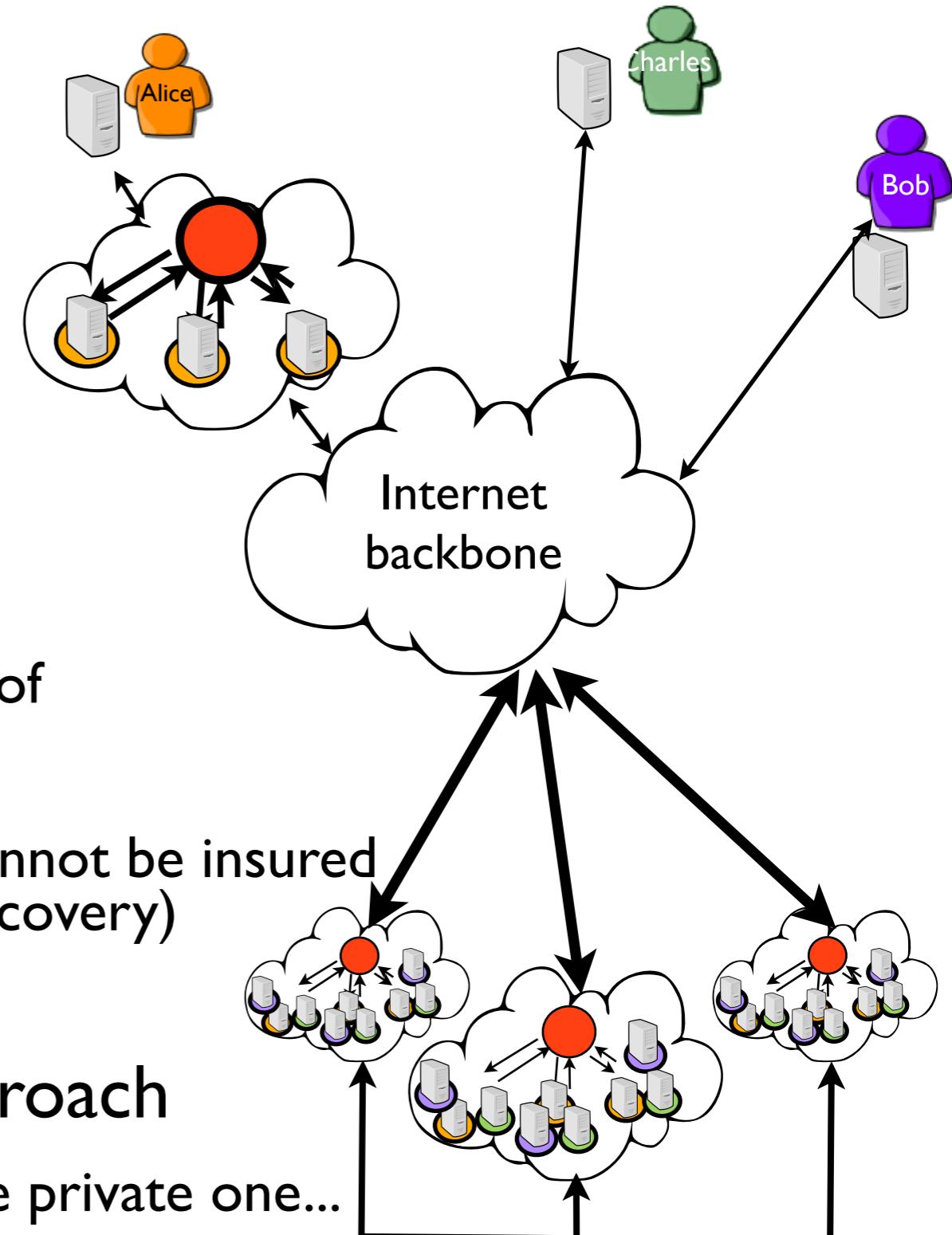
- I. Externalization of private applications/ data (jurisdiction concerns)

2. Overhead implied by the unavoidable use of the Internet to reach distant platforms

3. The connectivity to the application/data cannot be insured by centralized dedicated centers (disaster recovery)

- Hybrid platforms: a promising approach

It depends how you are going to extend the private one...



Can we address these concerns “all in one” ? ?

7/67

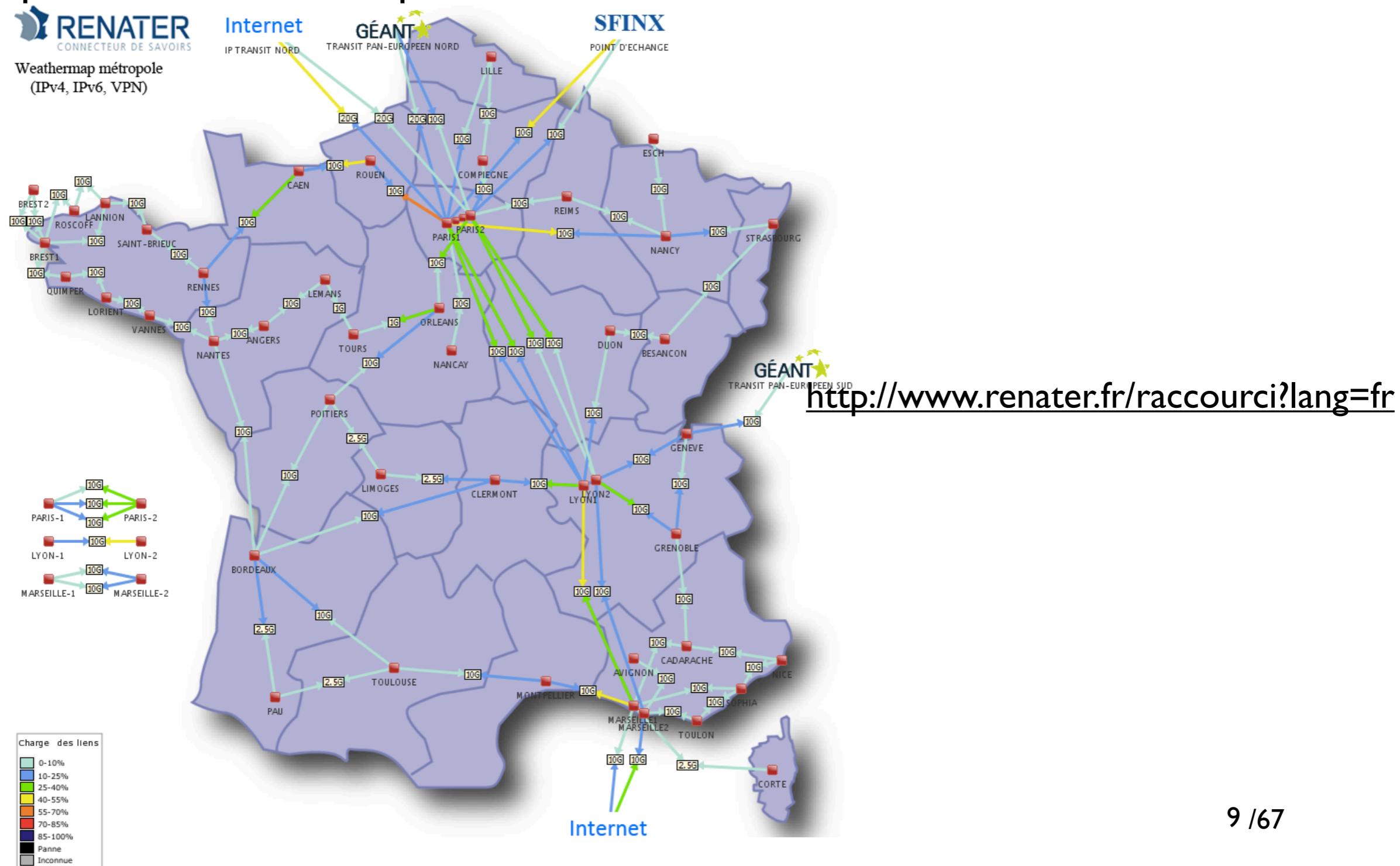
How can the μ DC concept be deployed ?

Locality Based Utility Computing Toward LUC Infrastructures

Beyond the Cloud, the DISCOVERY Initiative

• Locality-based UC infrastructures

The only way to deliver highly efficient and sustainable UC services is to provide UC platforms as close as possible to the end-users.



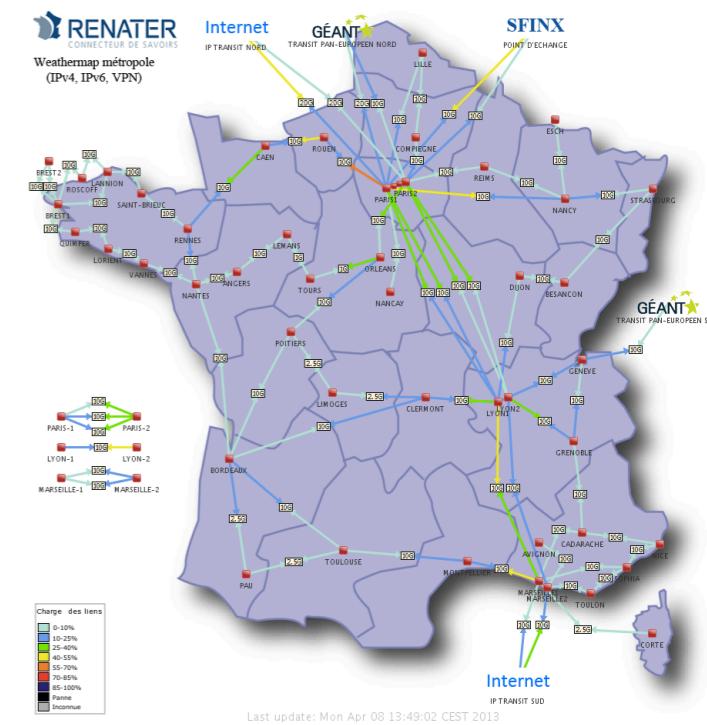
Beyond the Cloud, the DISCOVERY Initiative

- Locality-based UC infrastructures

The only way to deliver highly efficient and sustainable UC services is to provide UC platforms as close as possible to the end-users.

- Leveraging network backbones

Extend any point of presence of a network backbone with UC servers (from network hubs up to major DSLAMs that are operated by telecom companies and network institutions).



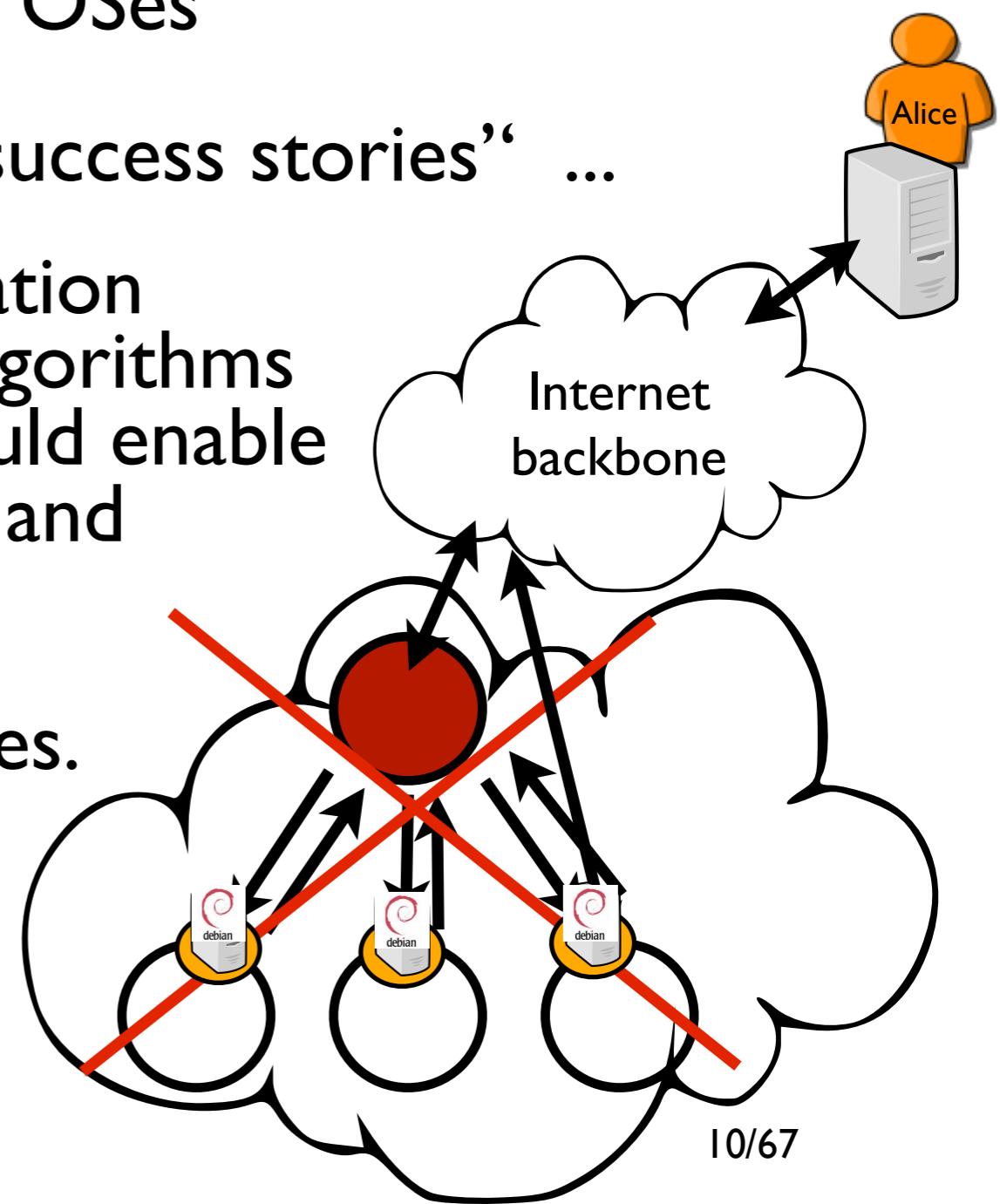
⇒ **Operating such widely distributed resources requires the definition of a fully distributed system**

The DISCOVERY Proposal

- DIStributed and COoperative framework to manage Virtual EnviRonments autonomicallY (the LUC OS)
- Designing/implementing Distributed OSes

Deeply investigated with no “real success stories” ...

... But maturity of system virtualization capabilities as well as large scale algorithms and autonomous mechanisms should enable to design and implement a unified and autonomic system manipulating virtual environments (VEs) like traditional OS manipulate processes.

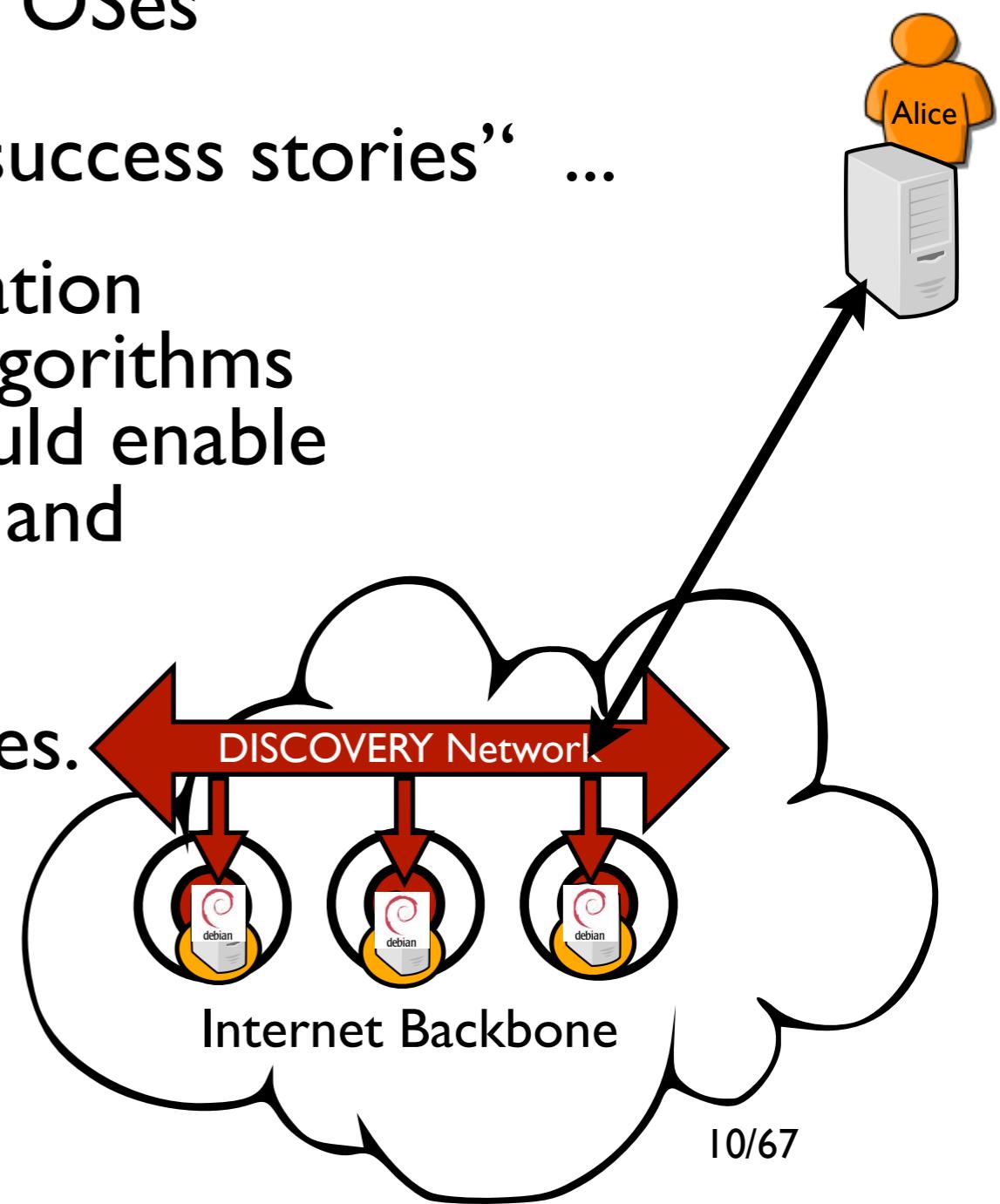


The DISCOVERY Proposal

- DIStributed and COoperative framework to manage Virtual EnviRonments autonomicallY (the LUC OS)
- Designing/implementing Distributed OSes

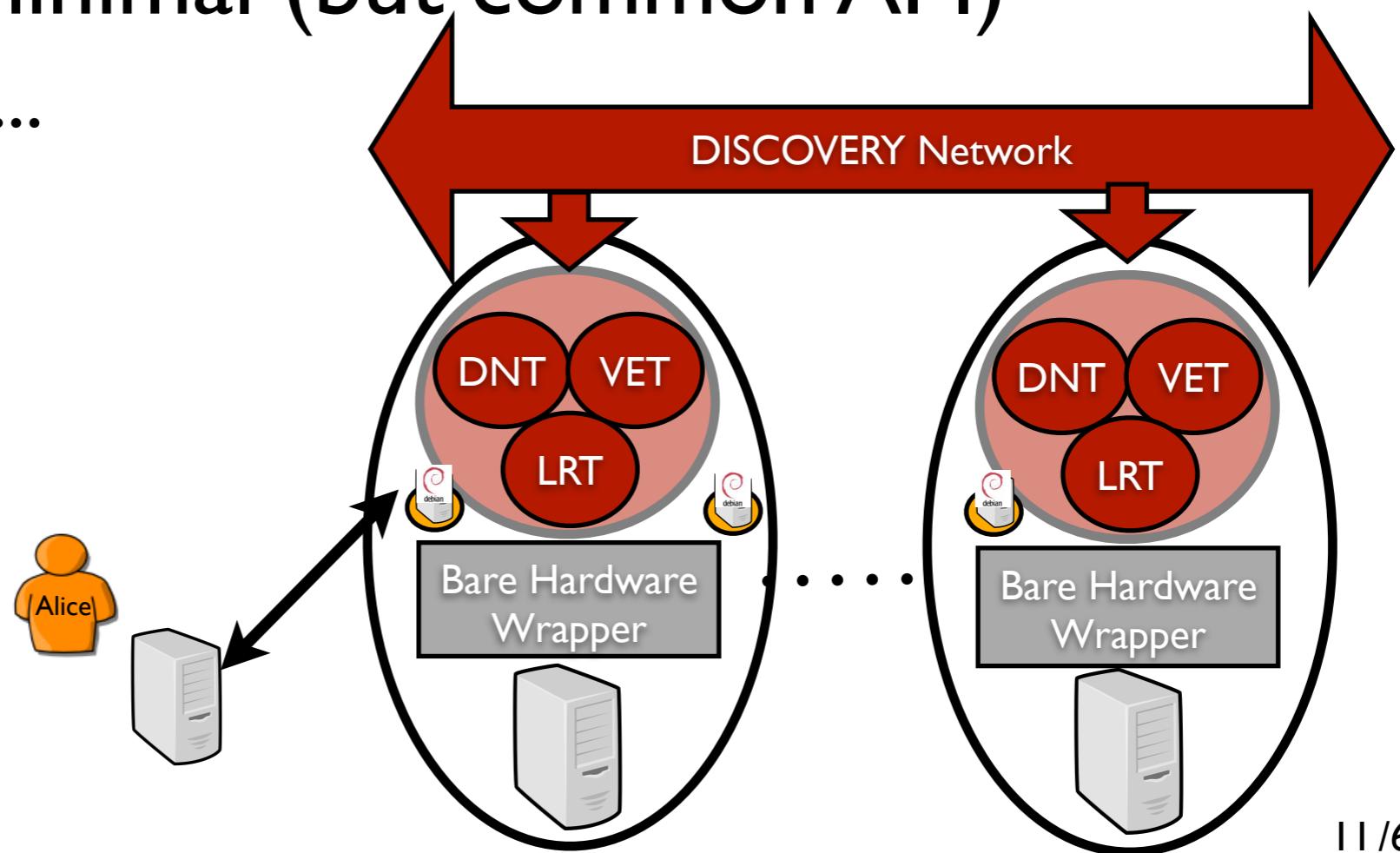
Deeply investigated with no “real success stories” ...

... But maturity of system virtualization capabilities as well as large scale algorithms and autonomous mechanisms should enable to design and implement a unified and autonomic system manipulating virtual environments (VEs) like traditional OS manipulate processes.



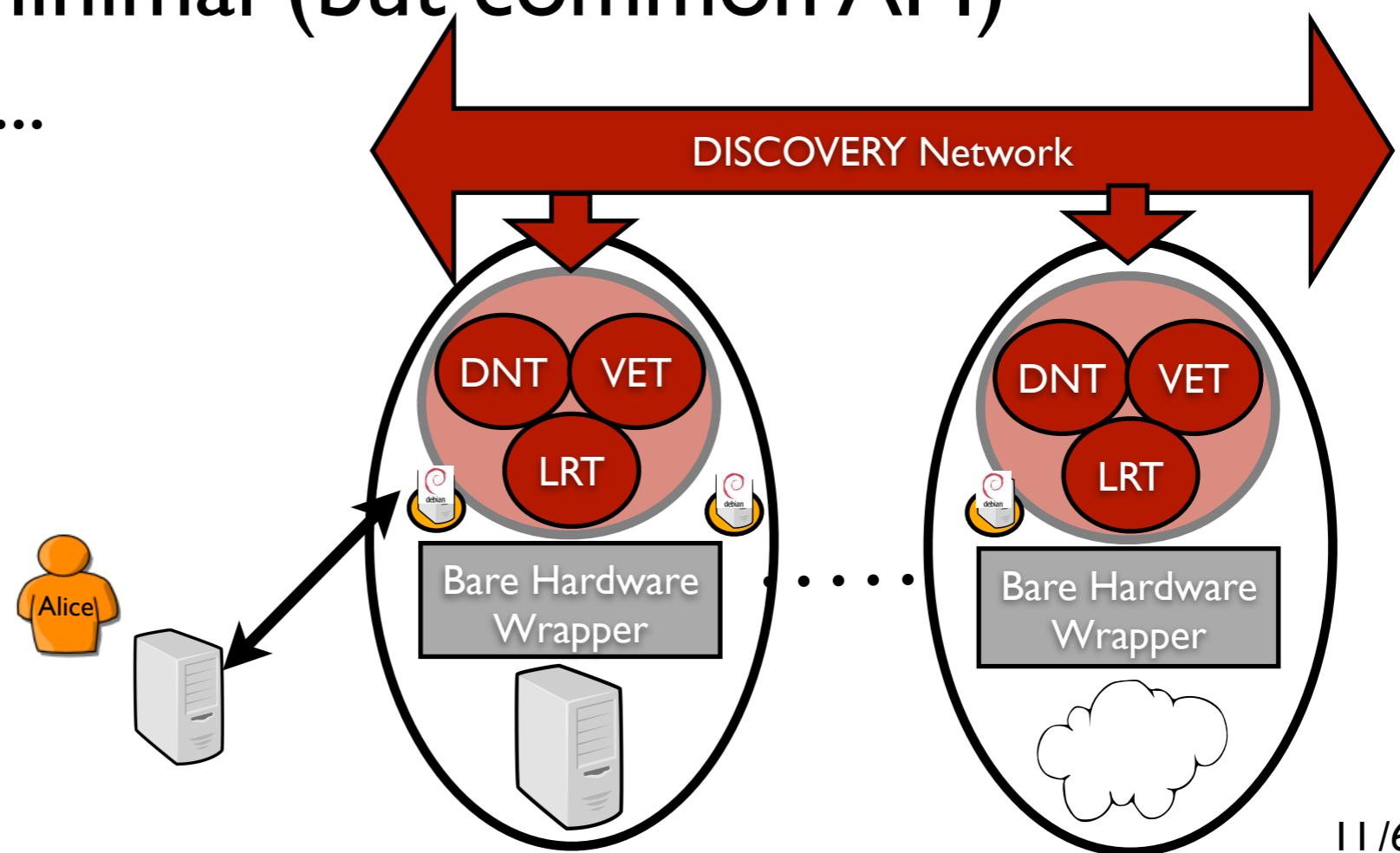
The LUC OS Agent - Overview

- 3 services
Discovery Network Tracker (DNT)
Virtual Environments Tracker (VET)
Local Resources Tracker (LRT)
- Relying on a minimal (but common API)
libvirt / OCCI / ...



The LUC OS Agent - Overview

- 3 services
Discovery Network Tracker (DNT)
Virtual Environments Tracker (VET)
Local Resources Tracker (LRT)
- Relying on a minimal (but common API)
libvirt / OCCI / ...



The DISCOVERY Initiative

- Focusing on the design and the implementation of a complete OS for IaaS platforms (i.e. the LUC OS)

The LUC OS

Based on VMs and VEs (group of VMs) as the fundamental granularity

Scalability, targeting the management of hundred thousands of VMs upon thousands of physical machines (PMs)

Reliability, considering “hardware failures as the norm rather the exception”

Reactivity, handling each reconfiguration event as swiftly as possible to maintain VEs' QoS.

- May look simple but lots of scientific/technical challenges

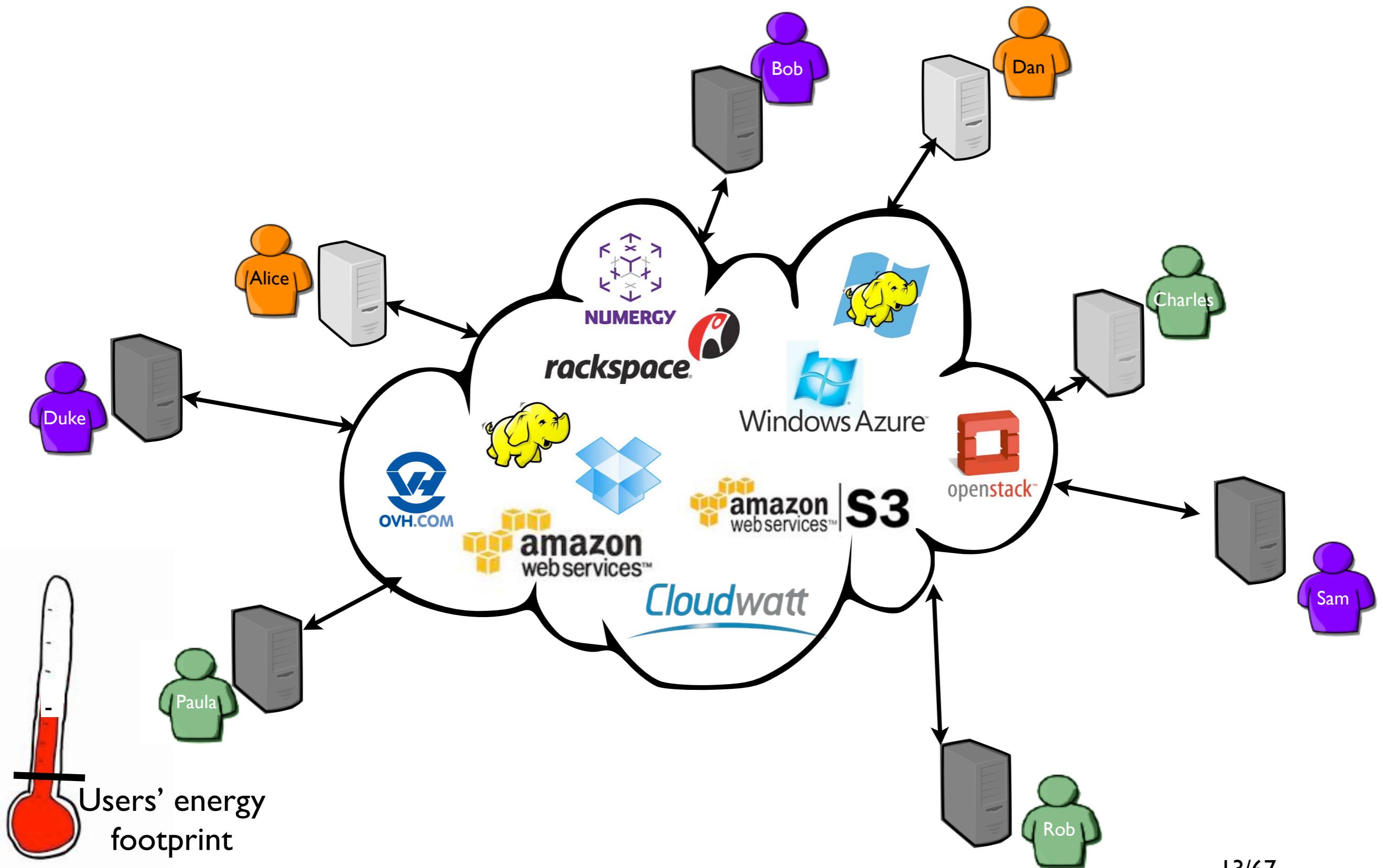
Cost of the DISCOVERY network !? partial view of the system !?

Impact on the others VMs !?, management of VM images !?

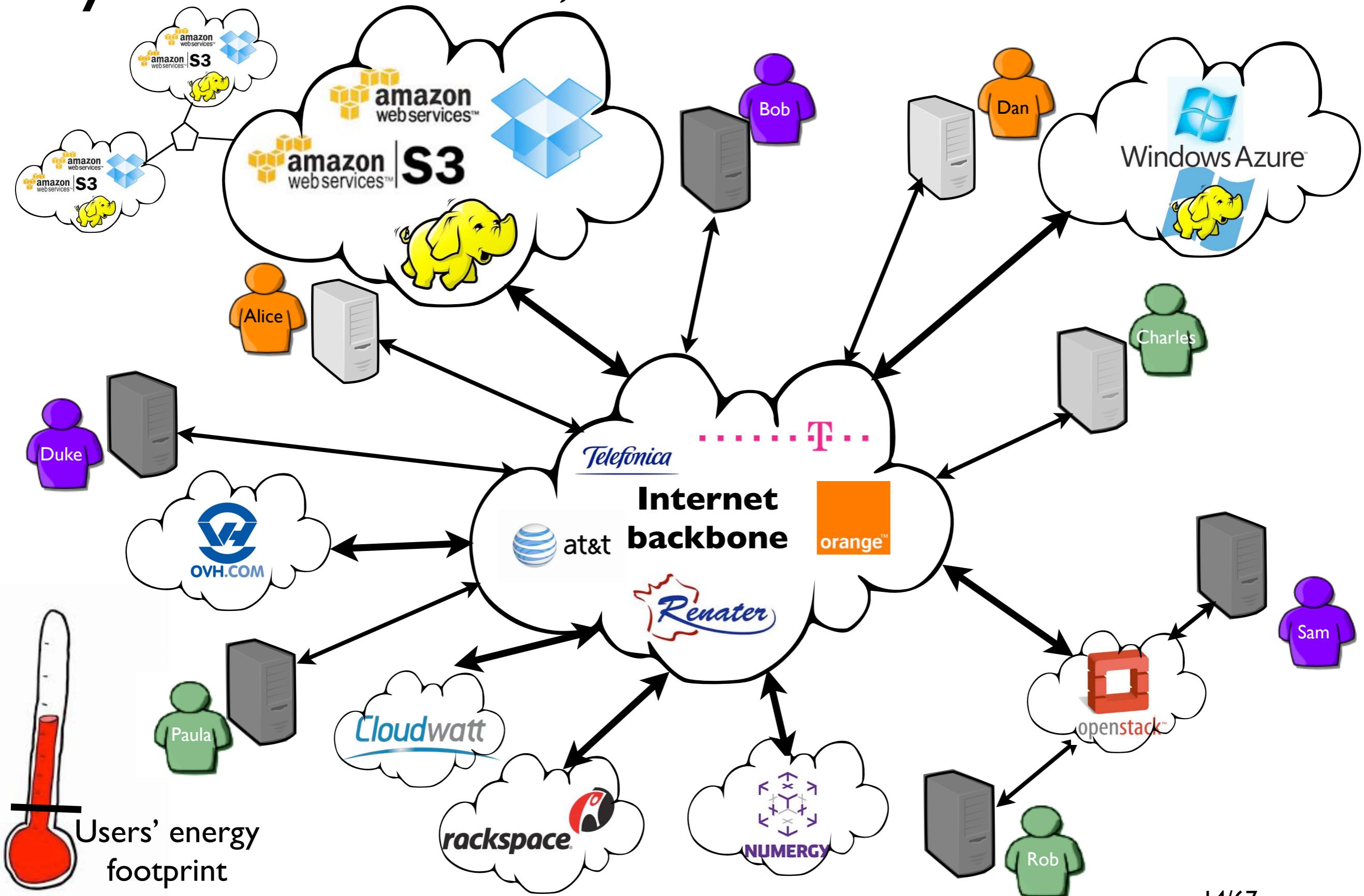
Which software abstractions to make the development easier and more reliable (distributed event programming) ? How to take into account locality aspects ?

- A BitTorrent like system ... but with stronger assumptions

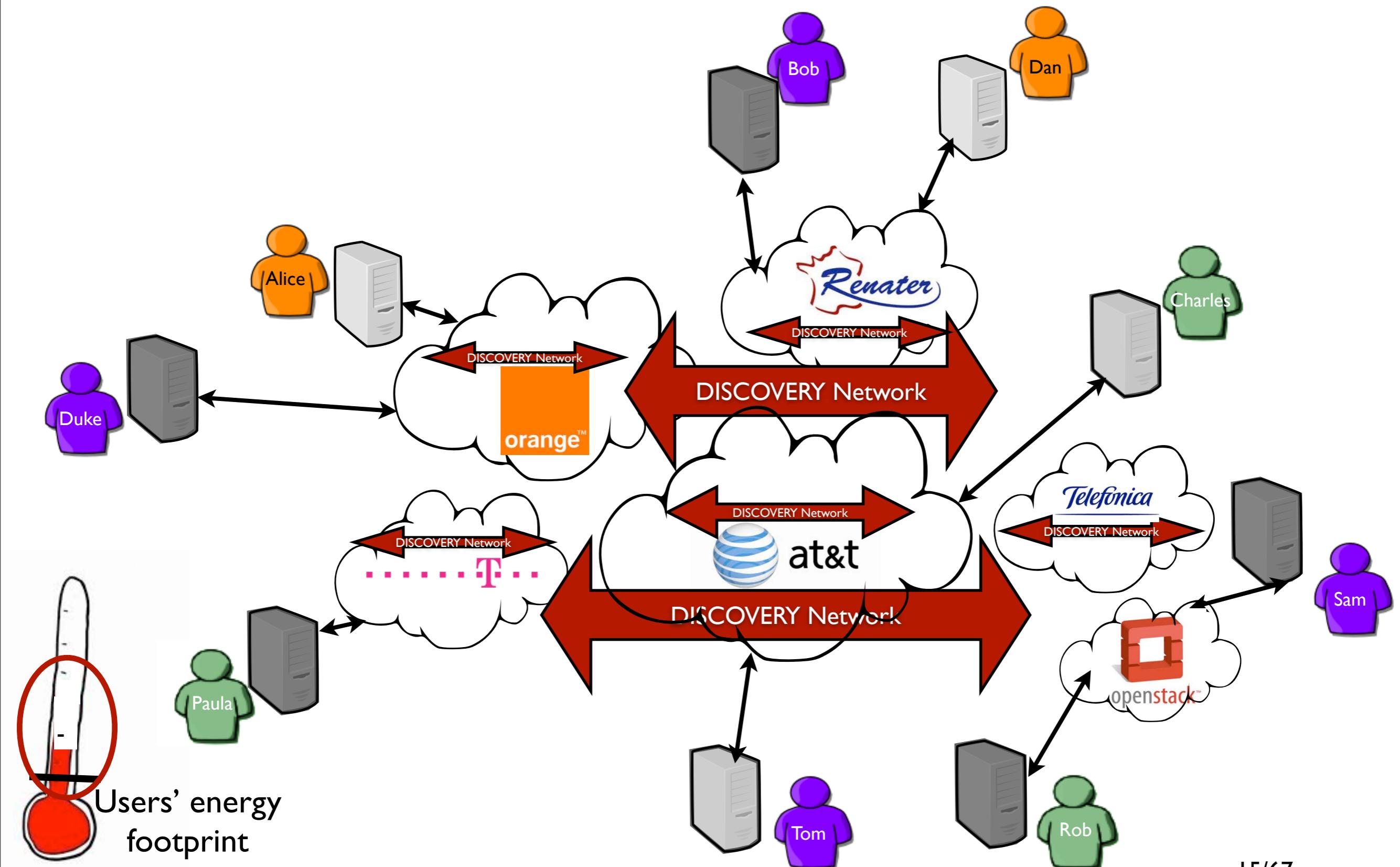
Beyond the Cloud, the DISCOVERY Initiative



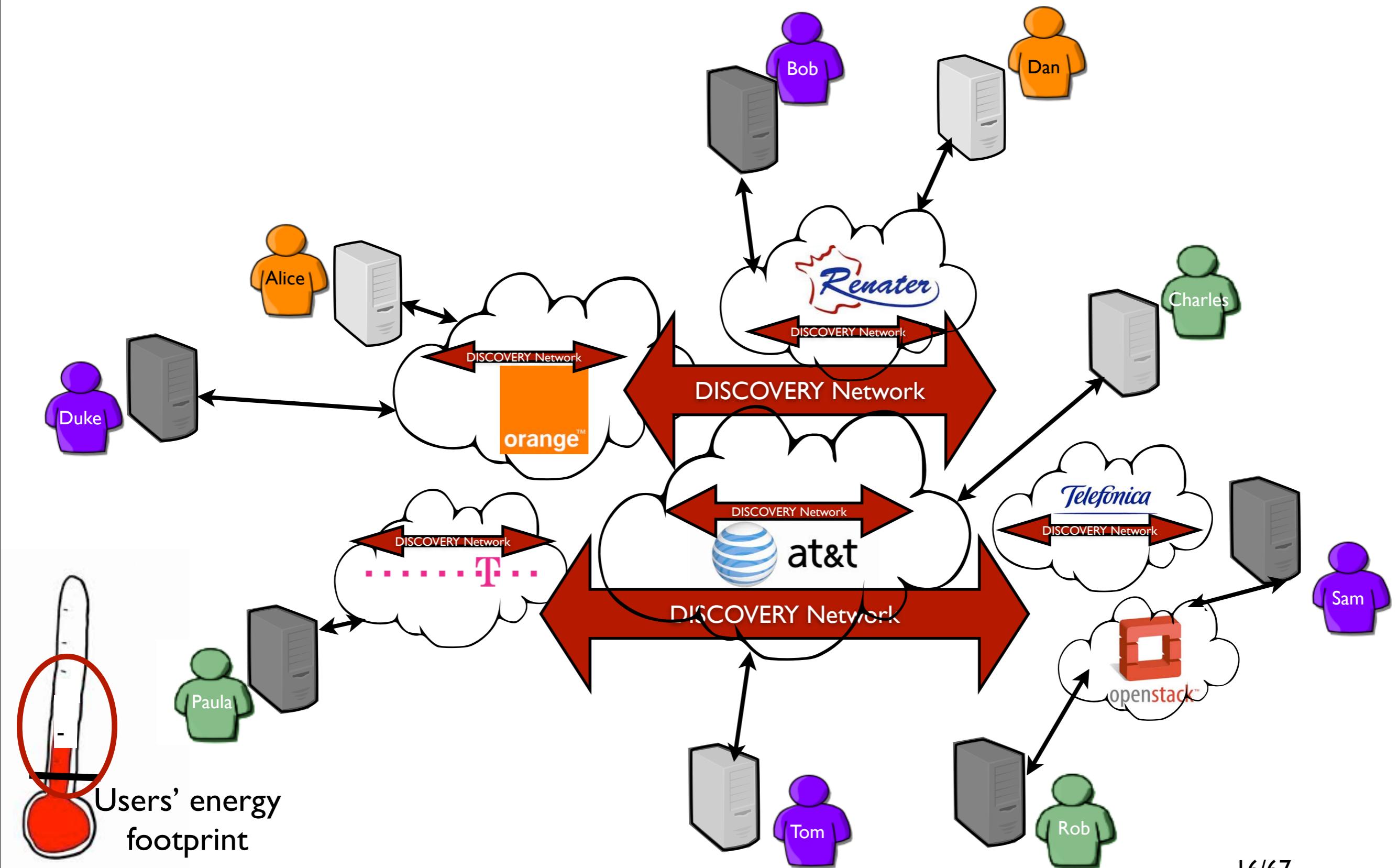
Beyond the Cloud, the DISCOVERY Initiative



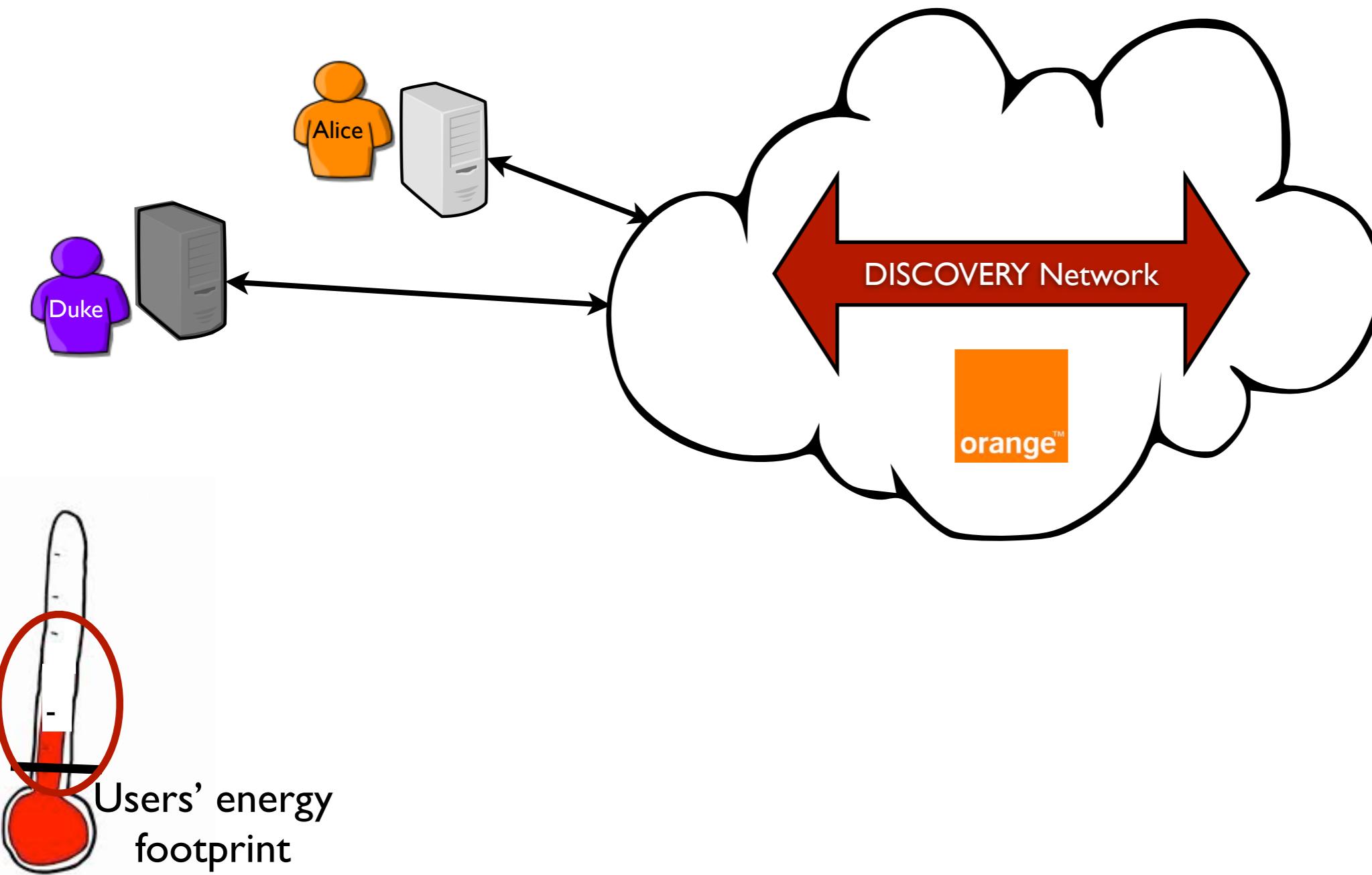
Beyond the Cloud, the DISCOVERY Initiative



Beyond the Cloud, the DISCOVERY Initiative

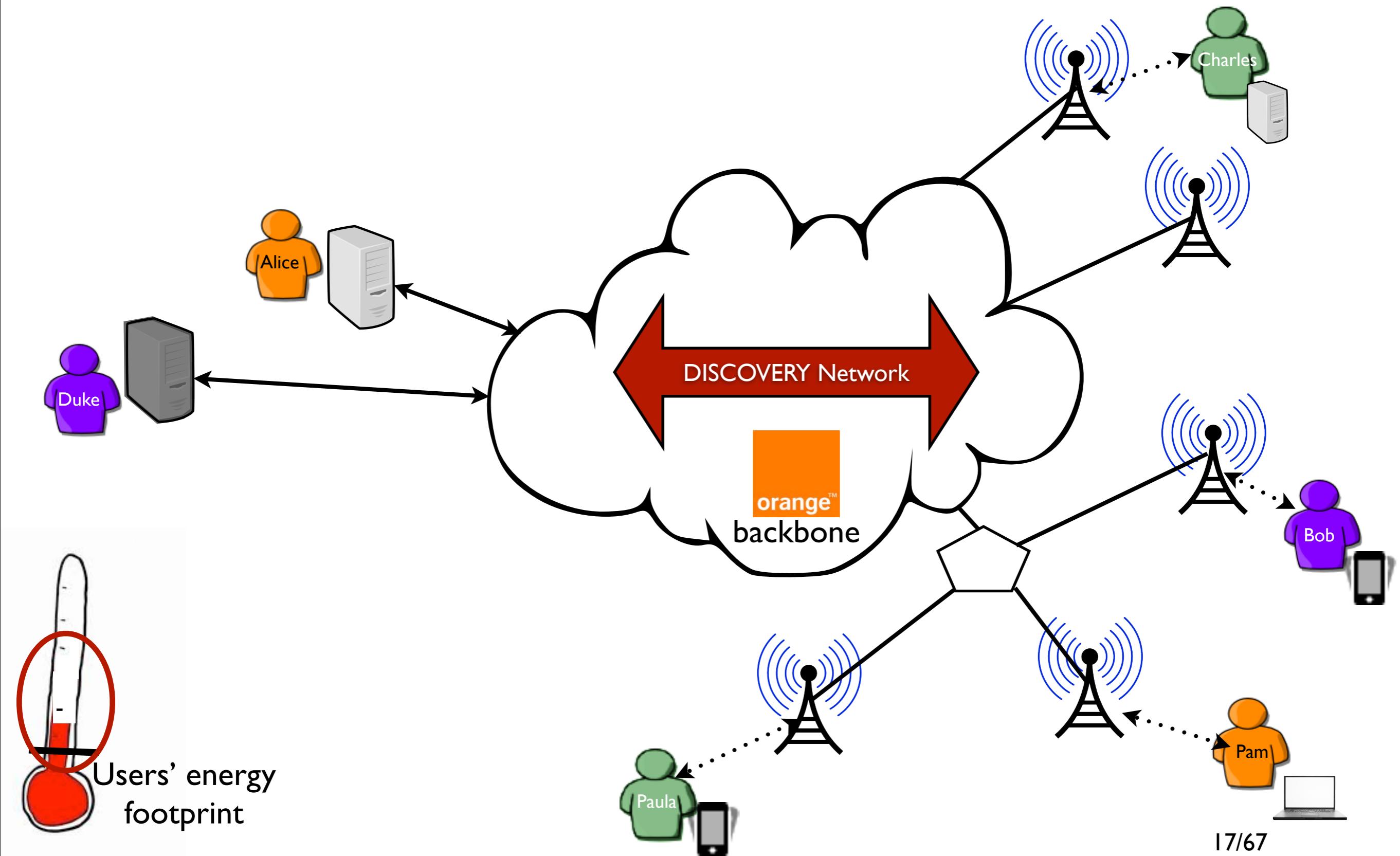


Beyond the Cloud, the DISCOVERY Initiative



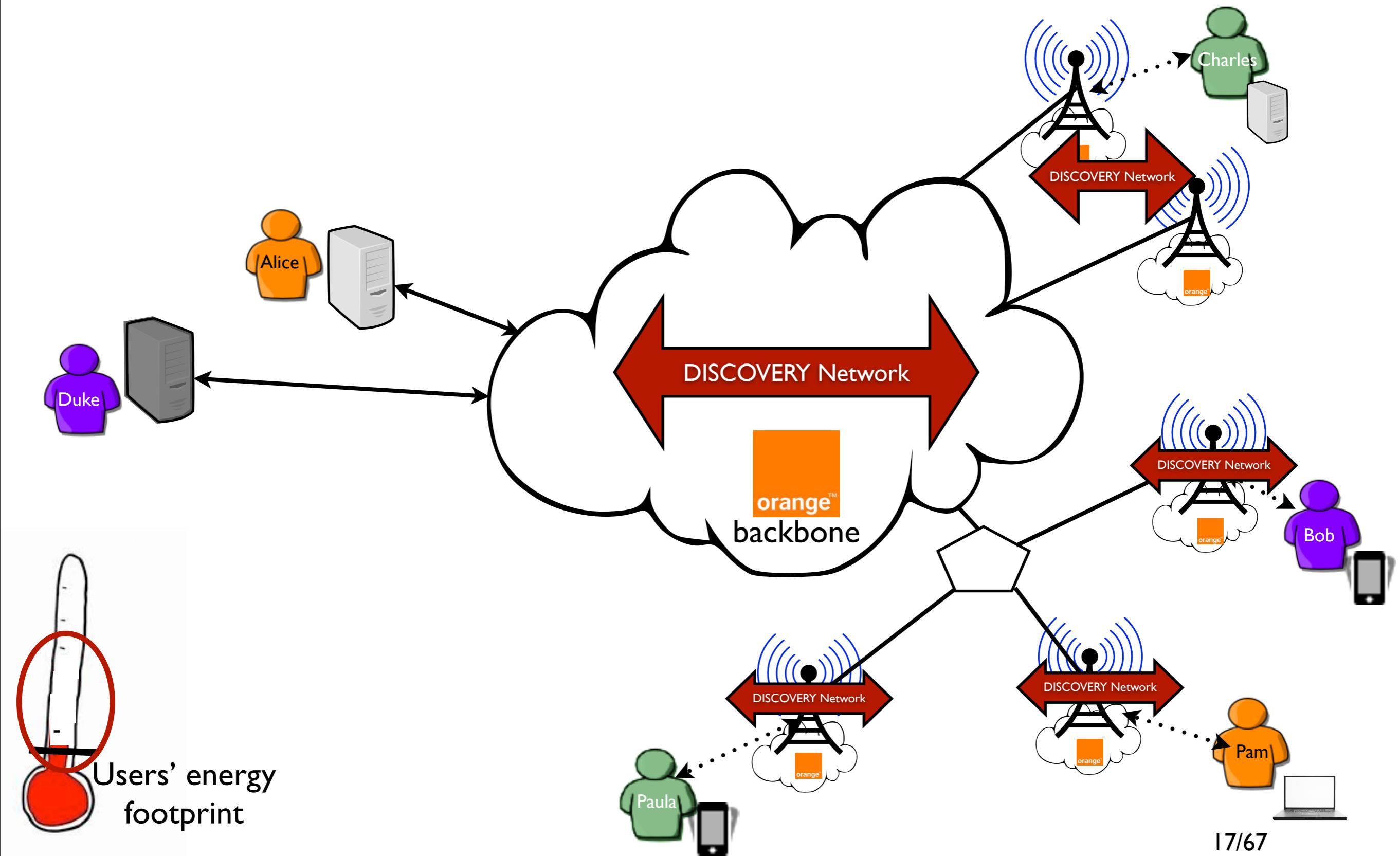
16/67

Beyond the Cloud, the DISCOVERY Initiative



17/67

Beyond the Cloud, the DISCOVERY Initiative



The DISCOVERY Initiative

- Leveraging former projects but still on the starting blocks!
- Strong interests of large companies
(SAP, Orange Lab, Citrix, ...)
- RENATER
- An important actor to follow: Akamai (micro DCs)
- Preliminary works with promising results
(especially on the LRT: a first POC)
- Long term objective: impact on the design of distributed applications in order to take advantage of the locality
(building S3 like system)

Conclusion

- Cloud Computing technology is changing every day
 - New features, new requirements

The main challenge of the Discovery Initiative is to ensure that such new features/mechanisms can run in a distributed manner.
- But Distributed Cloud Computing is happening !
 - Dist. CC workshop (collocated with IEEE/ACM UCC 2013)
 - FOG Computing workshop (collocated with IEEE ICC 2013)

Amazon is on the way !

[Sign Up](#)[My Account / Console](#)

English

[AWS Products & Solutions](#)[AWS Product Information](#)[Developers](#)[Support](#)

Global Infrastructure

Amazon Web Services serves hundreds of thousands of customers in more than 190 countries.

We are steadily expanding global infrastructure to help our customers achieve lower latency and higher throughput, and to ensure that their data resides only in the Region they specify. As our customers grow their businesses, AWS will continue to provide infrastructure that meets their global requirements.

[See detailed list of offerings at all AWS locations](#)



Europe / Middle East / Africa



EU (Ireland) Region

EC2 Availability Zones: 3 Launched 2007

AWS Edge Locations

Amsterdam,
The Netherlands (2)

Marseille, France

Dublin, Ireland

Milan, Italy

Frankfurt,
Germany (3)

Paris, France (2)

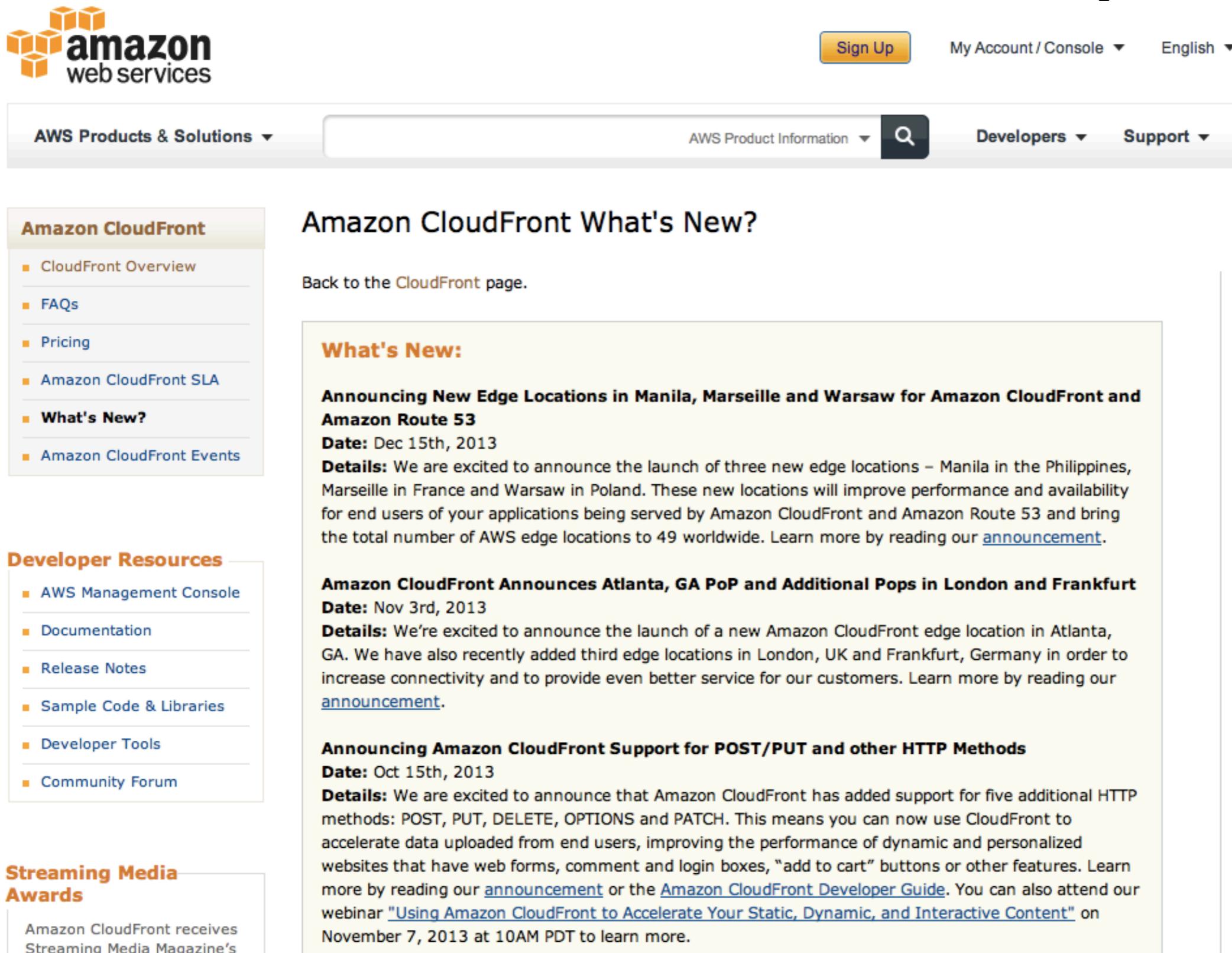
London, England (3)

Stockholm, Sweden

Madrid, Spain

Warsaw, Poland

Amazon is on the way !



The screenshot shows the Amazon CloudFront "What's New" page. At the top, there's a navigation bar with links for "Sign Up", "My Account / Console", and "English". Below that is a search bar and links for "AWS Products & Solutions", "AWS Product Information", "Developers", and "Support". On the left, there's a sidebar with sections for "Amazon CloudFront" (CloudFront Overview, FAQs, Pricing, SLA, What's New?, Events), "Developer Resources" (AWS Management Console, Documentation, Release Notes, Sample Code & Libraries, Developer Tools, Community Forum), and "Streaming Media Awards". The main content area features a section titled "What's New:" with three announcements:

- Announcing New Edge Locations in Manila, Marseille and Warsaw for Amazon CloudFront and Amazon Route 53**
Date: Dec 15th, 2013
Details: We are excited to announce the launch of three new edge locations – Manila in the Philippines, Marseille in France and Warsaw in Poland. These new locations will improve performance and availability for end users of your applications being served by Amazon CloudFront and Amazon Route 53 and bring the total number of AWS edge locations to 49 worldwide. Learn more by reading our [announcement](#).
- Amazon CloudFront Announces Atlanta, GA PoP and Additional Pops in London and Frankfurt**
Date: Nov 3rd, 2013
Details: We're excited to announce the launch of a new Amazon CloudFront edge location in Atlanta, GA. We have also recently added third edge locations in London, UK and Frankfurt, Germany in order to increase connectivity and to provide even better service for our customers. Learn more by reading our [announcement](#).
- Announcing Amazon CloudFront Support for POST/PUT and other HTTP Methods**
Date: Oct 15th, 2013
Details: We are excited to announce that Amazon CloudFront has added support for five additional HTTP methods: POST, PUT, DELETE, OPTIONS and PATCH. This means you can now use CloudFront to accelerate data uploaded from end users, improving the performance of dynamic and personalized websites that have web forms, comment and login boxes, "add to cart" buttons or other features. Learn more by reading our [announcement](#) or the [Amazon CloudFront Developer Guide](#). You can also attend our webinar "[Using Amazon CloudFront to Accelerate Your Static, Dynamic, and Interactive Content](#)" on November 7, 2013 at 10AM PDT to learn more.

20/67

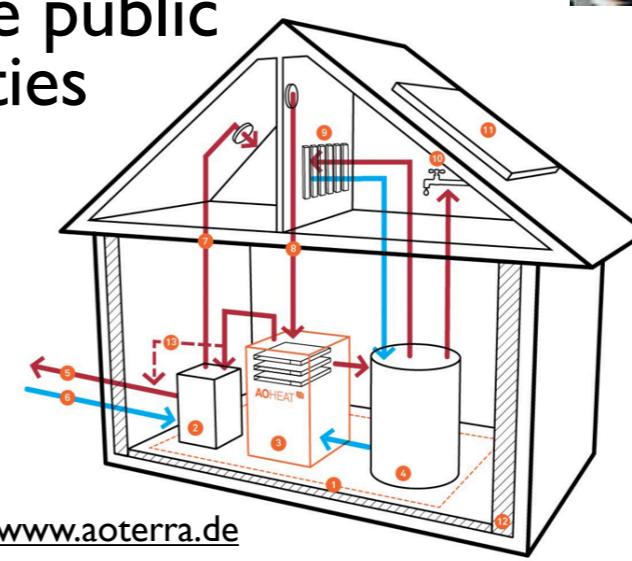
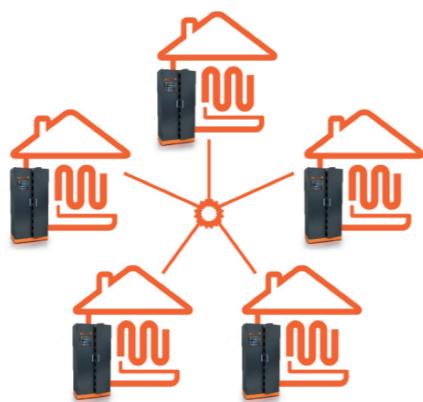
Beyond Discovery !

- From sustainable data centers to a new source of energy
The only way to deliver highly efficient and sustainable UC services is to provide UC platforms as close as possible to the end-users and to...
- Leverage “green” energy (solar, wind turbines...)
Transfer the green micro/nano DCs concept to the network PoP
Take the advantage of the geographical distribution
- Leveraging the data furnaces concept

Deploy UC servers in medium and large institutions and use them as sources of heat inside public buildings such as hospitals or universities



<http://parasol.cs.rutgers.edu>

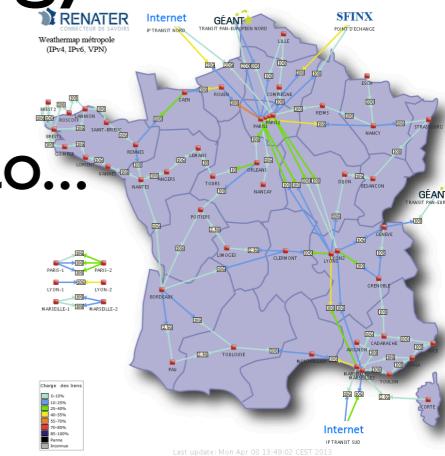


21/67

Beyond Discovery !

- From sustainable data centers to a new source of energy

The only way to deliver highly efficient and sustainable UC services is to provide UC platforms as close as possible to the end-users and to...



- Leverage “green” energy (solar, wind turbines...)

Transfer the green micro/nano DCs concept to the network PoP

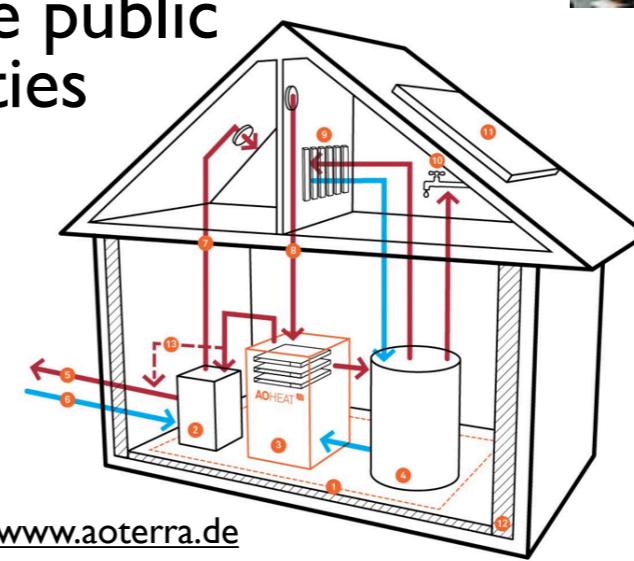
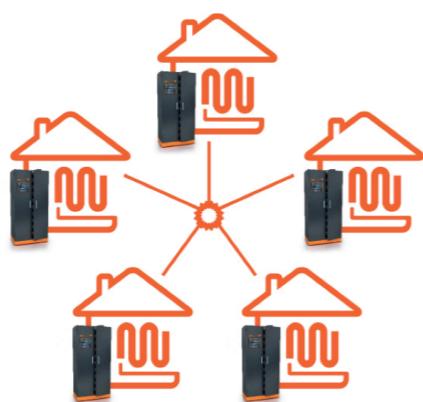
Take the advantage of the geographical distribution



<http://parasol.cs.rutgers.edu>

- Leveraging the data furnaces concept

Deploy UC servers in medium and large institutions and use them as sources of heat inside public buildings such as hospitals or universities



<https://www.aoterra.de>

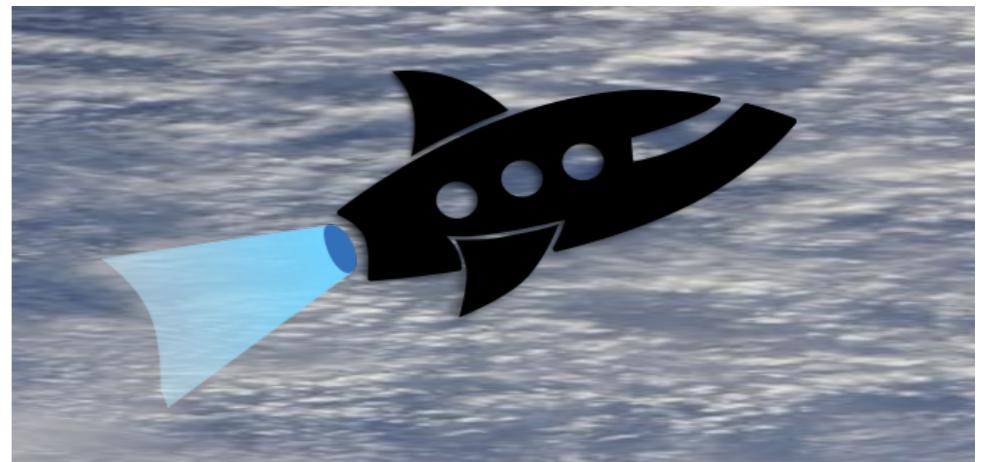
21/67

The DISCOVERY Initiative

- Thank you / Questions ?

- Additional materials

- Focus on LRT (Flavien Quesnel's Phd, ended in Feb 2013)
- Discovery internals in a nutshell
- On going work - The discovery framework from the Software Programming viewpoint (Jonathan Pastor's Phd, 2012/2015)



<http://beyondtheclouds.github.io/>

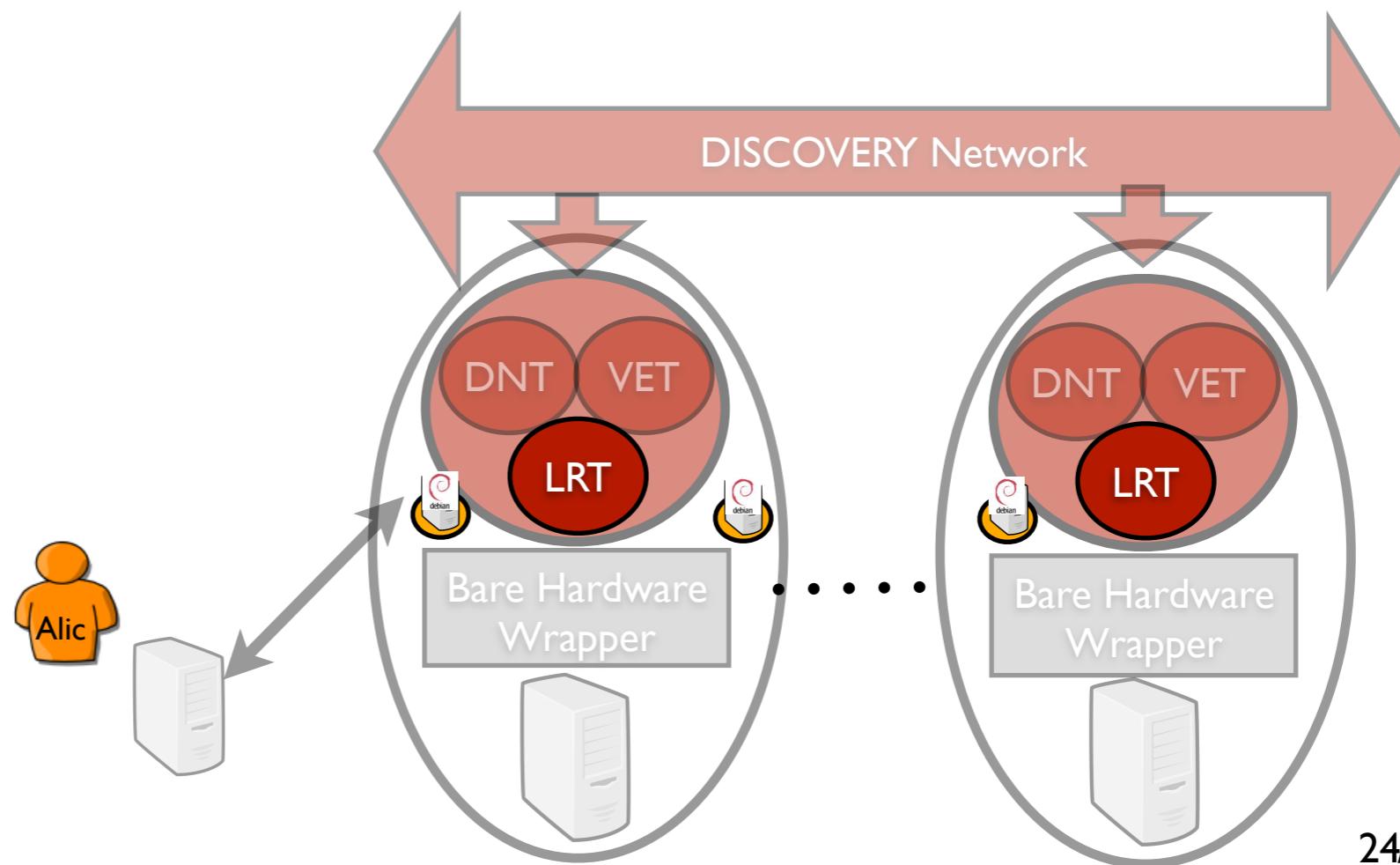
22/67

The LUC OS Agent

Focus on the LRT

The LUC OS Agent Local Resource Tracker

- The LRT is in charge of monitoring and dynamically balancing VMs according to their effective usage of the physical resources.

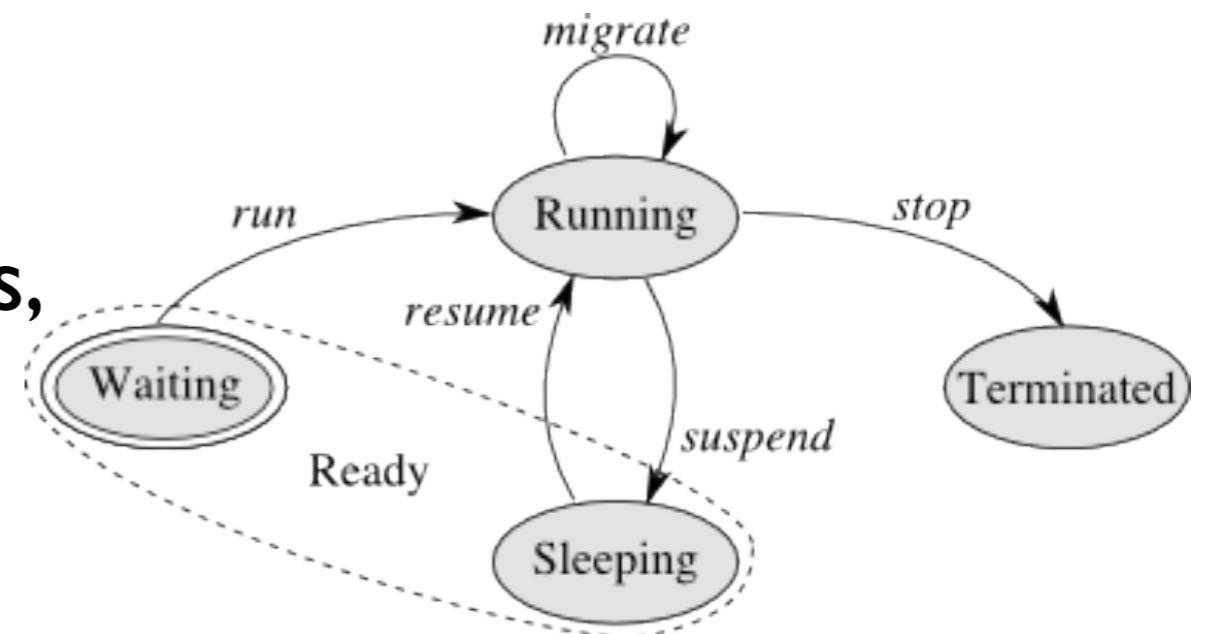


Background on VM dynamic scheduling

Background - a VE-based OS

- General idea: manipulate **VEs** instead of processes
(a VE is a users' working environment, possibly composed of several interconnected VMs)

- In a similar way of usual processes, each VE is in a particular state:



- Perform VE context switches (a set of VM context switches) to rebalance the LUC infrastructure according to the: scheduler objectives / available resources / waiting queue / ...

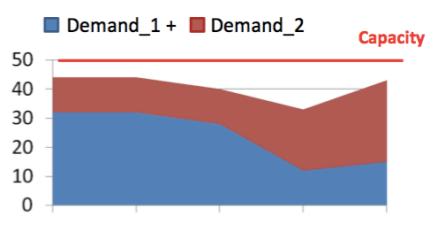
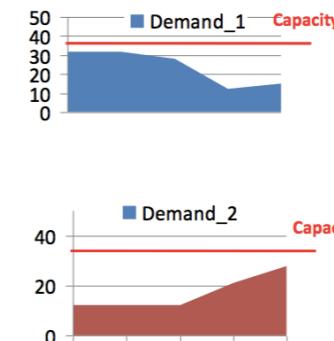
Background - The Entropy Proposal

- Fine management of resources (efficiency and energy constraints)
- Find the “right” mapping between needs of VMs and resources provided by PMs

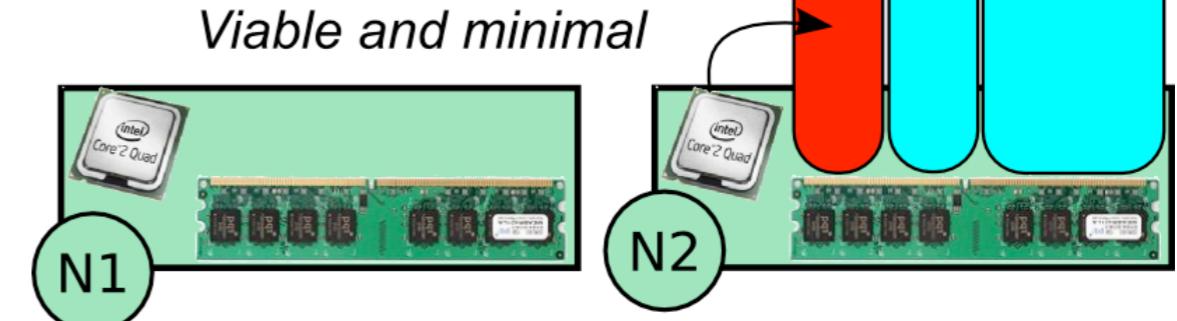
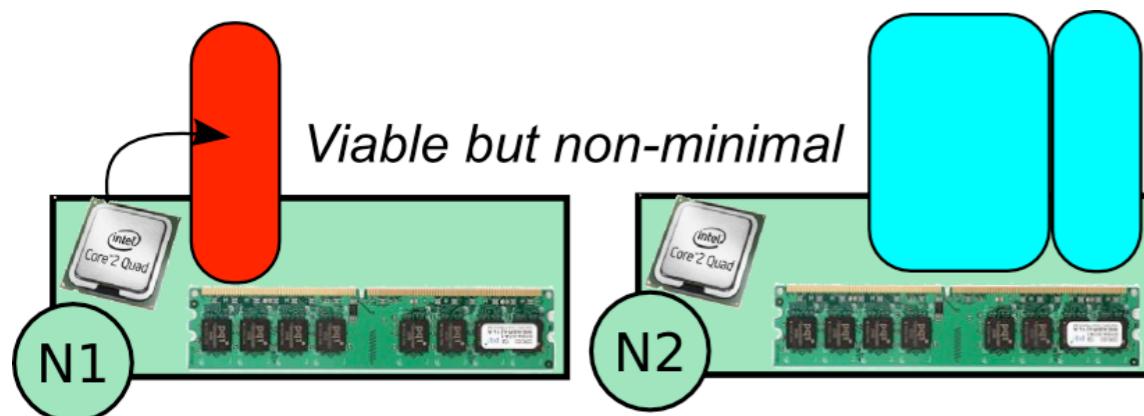
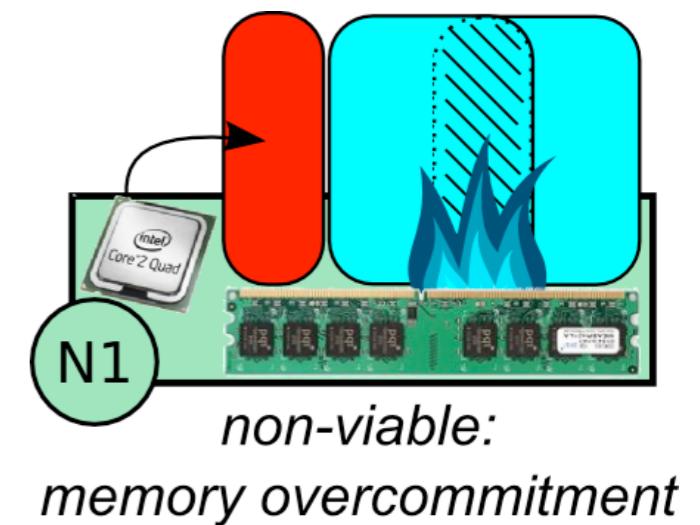
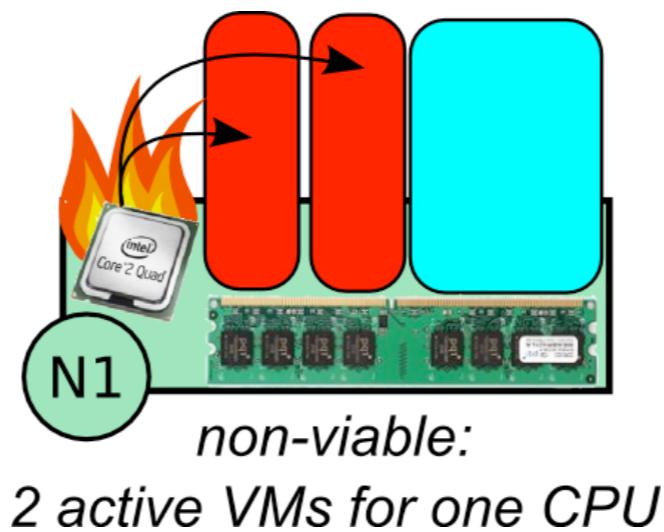
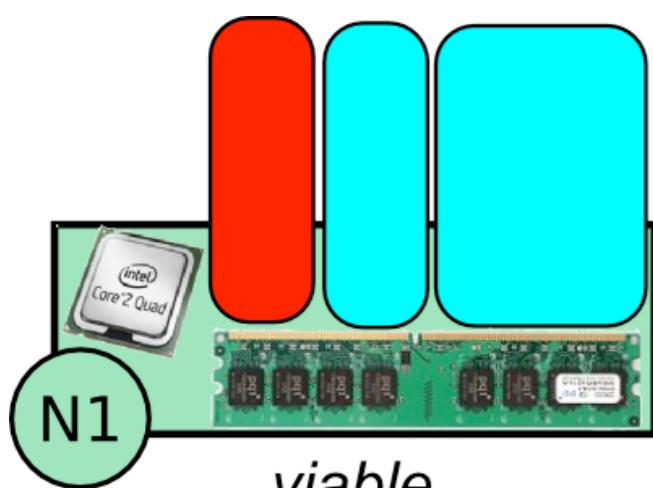


Cloud business model: Provider benefits

- Share capabilities (resources, services, etc.)



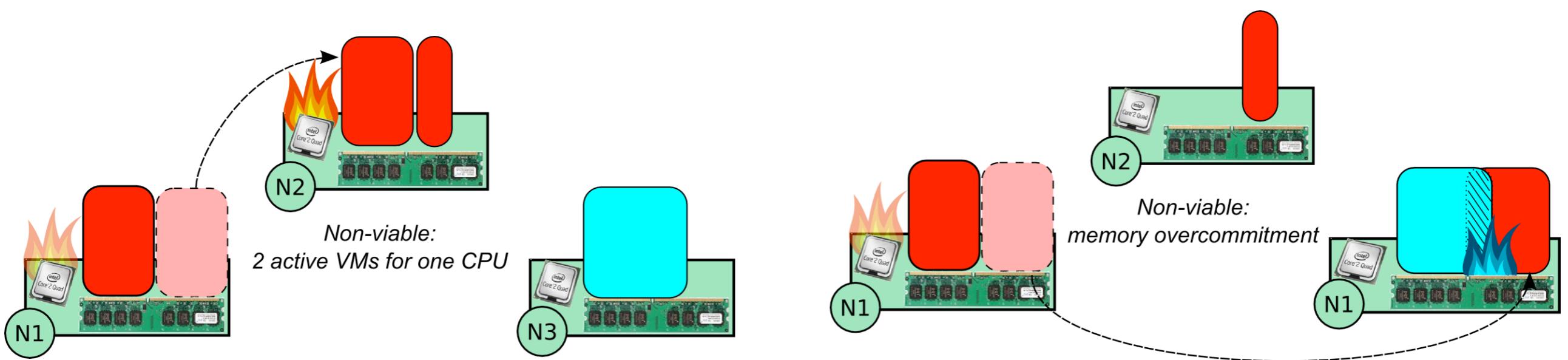
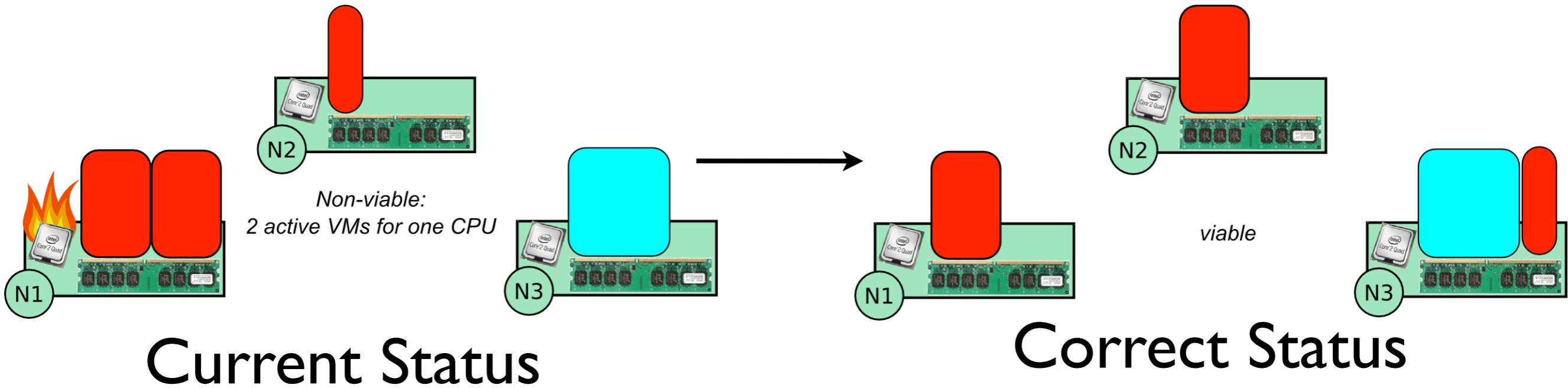
credits: S.Tata, Telecom Summer School 2013



credits: F. Hermenier, OSDI poster session 2008

27/67

Background - The Entropy Proposal

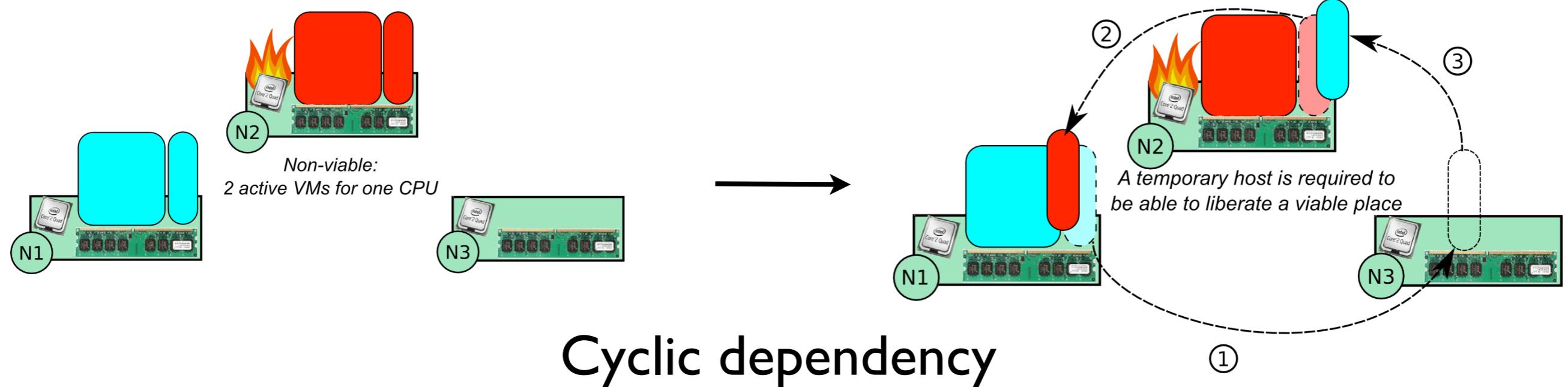
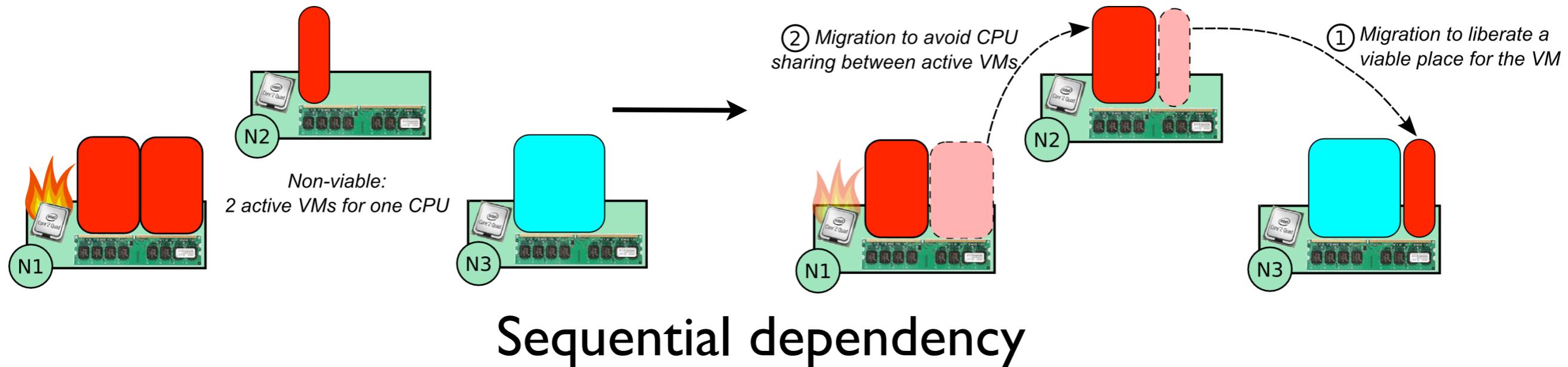


Non-viable manipulations

28/67

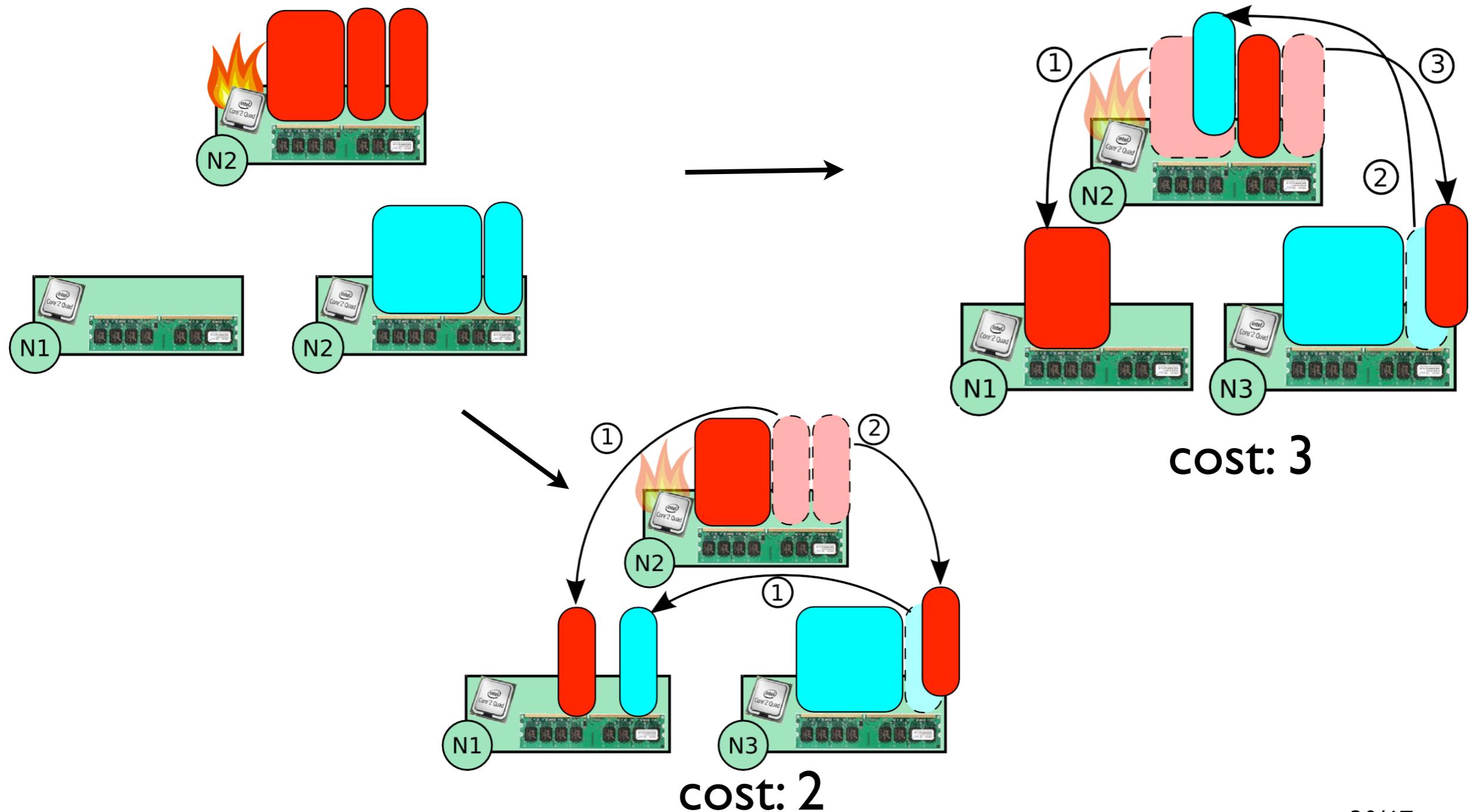
Background - The Entropy Proposal

- Order VM Operations



Background - The Entropy Proposal

- Optimizing the reconfiguration process



Background - The Entropy Proposal

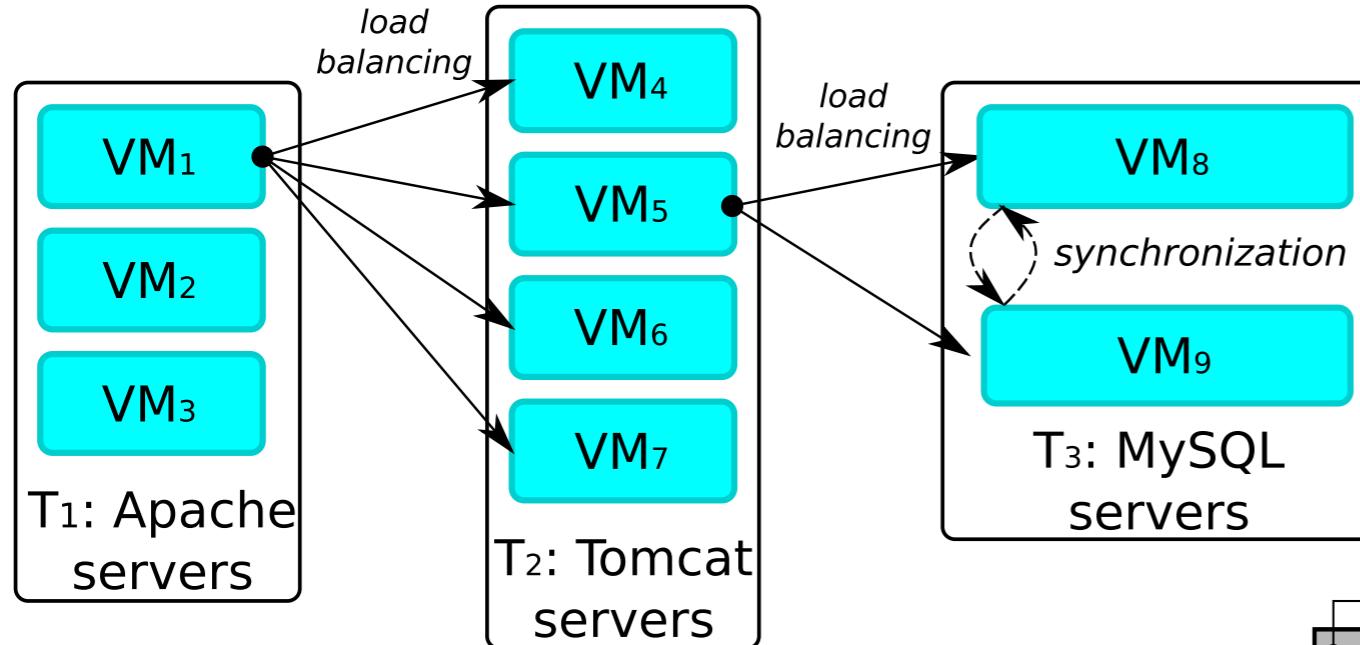
More Constraints

- Manipulate VEs dynamically can lead to non desired configurations
- Additional constraints should be considered

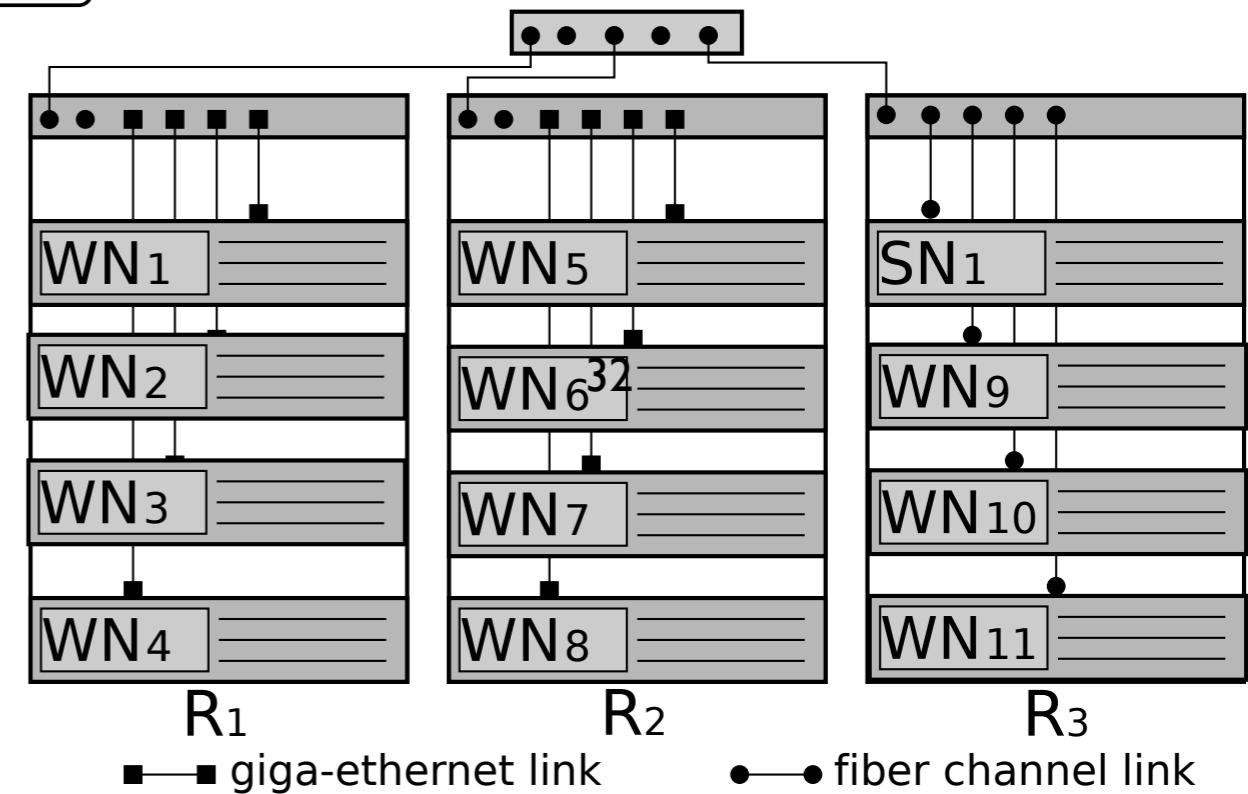
To take into account particular requirements according to the infrastructure (performance, HA, maintenance operations...)

To maintain VE “consistency” during reconfigurations

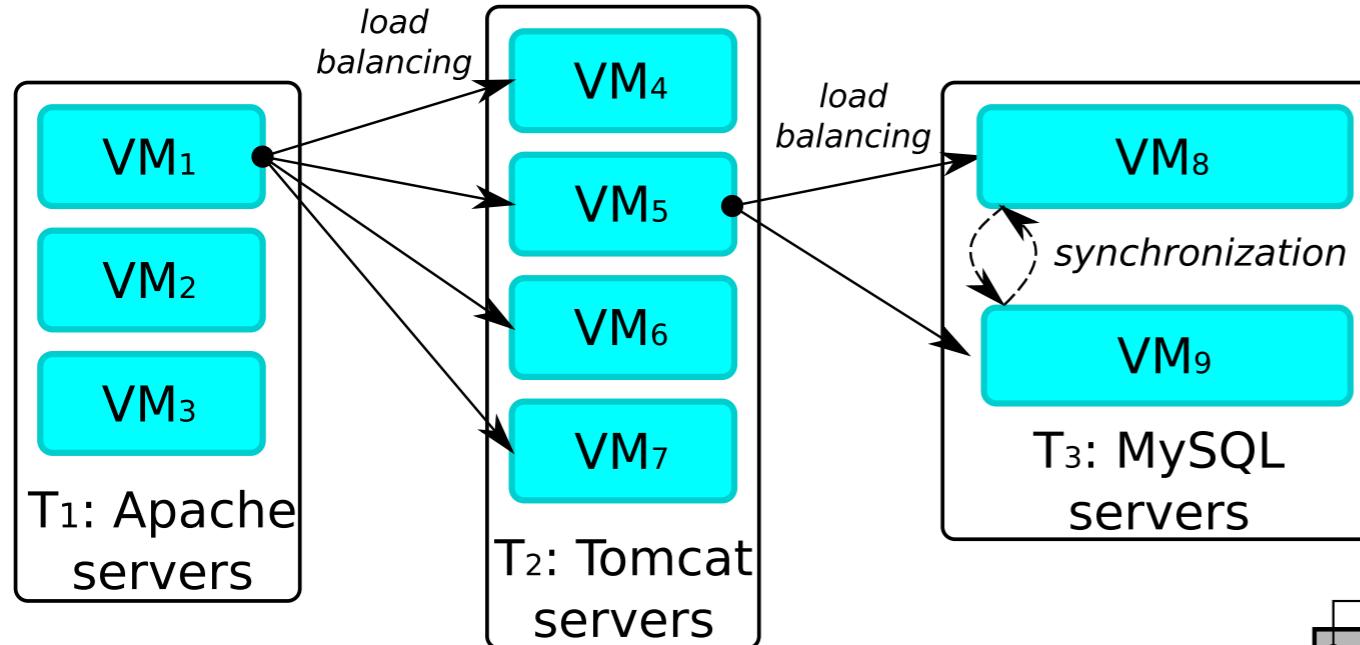
Background - Plasma and Entropy



Virtualized HA application

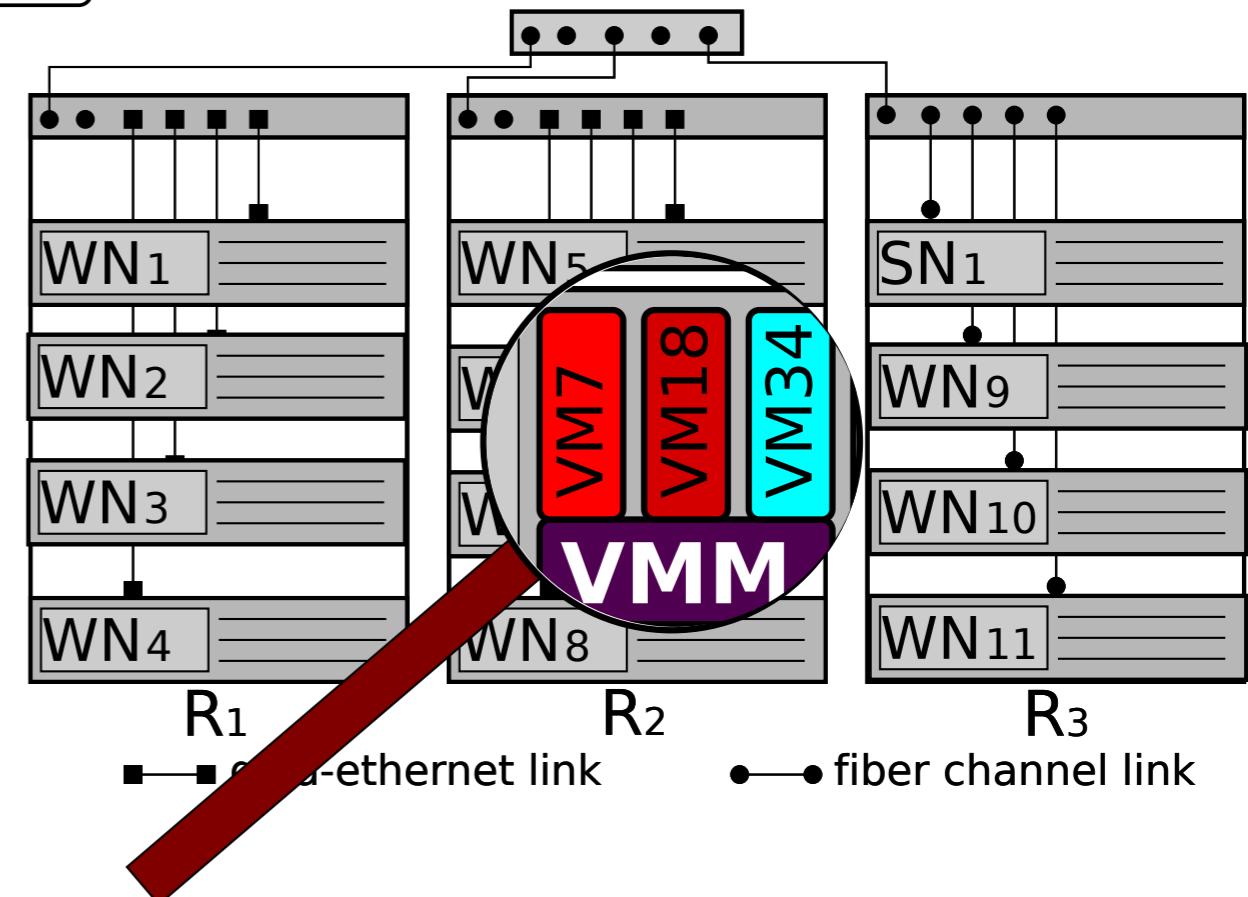


Background - Plasma and Entropy



Virtualized HA application

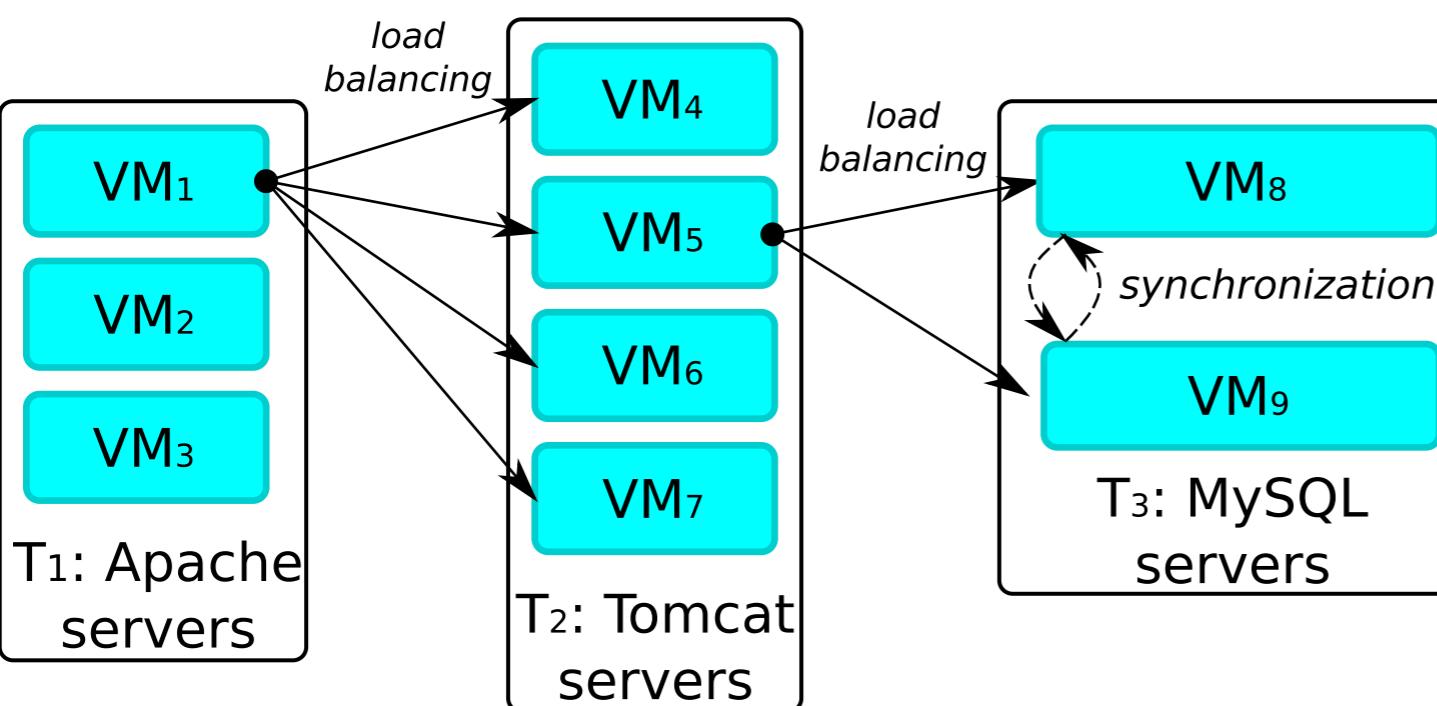
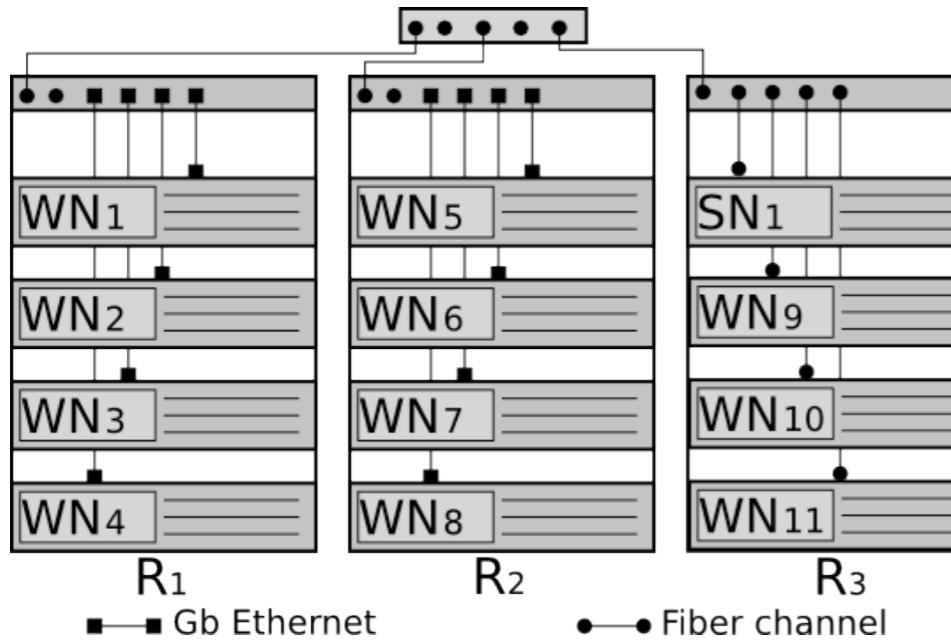
Plasma, a DSL to describe:
the infrastructure
the VEs and their placement
constraints



Background - Plasma and Entropy

- **ban({VM1,VM2}, {N1, N2})**
Prevents a set of VMs from being hosted on a given set of nodes
- **fence({VM1,VM2}, {N1, N2})**
Forces a set of VMs to be hosted on a set of nodes
- **spread({VM1,VM2})**
Ensures that the specified VMs are never hosted on the same node at the same time
- **latency({VM1,VM2}, {{N1,N2}, {N3,N4}})**
Forces a set of VMs to be hosted on a single group of nodes
- See more on <http://btrp.inria.fr/>

Infrastructure/Application Description



```
// Infrastructure  
$R1 = {WN1 ,WN2 ,WN3 ,WN4 };  
$R2 = WN [5..8];  
$R3 = WN [9..11] + {SN1 };
```

```
// Classes of latency  
$small = {$R3 };  
$medium = $R [1..3];
```

```
// Constraints  
ban ( $ALL_VMS ,{SN1 } );  
ban ( $ALL_VMS ,{WN5 } );  
fence ($A1 ,$R2 + $R3 );
```

```
// The 3 tiers  
$T1 = {VM1 ,VM2 ,VM3 };  
$T2 = VM [4..7];  
$T3 = VM [8..9];
```

```
// Fault tolerance to hw. failures  
spread($T1);  
spread($T2);  
spread($T3);
```

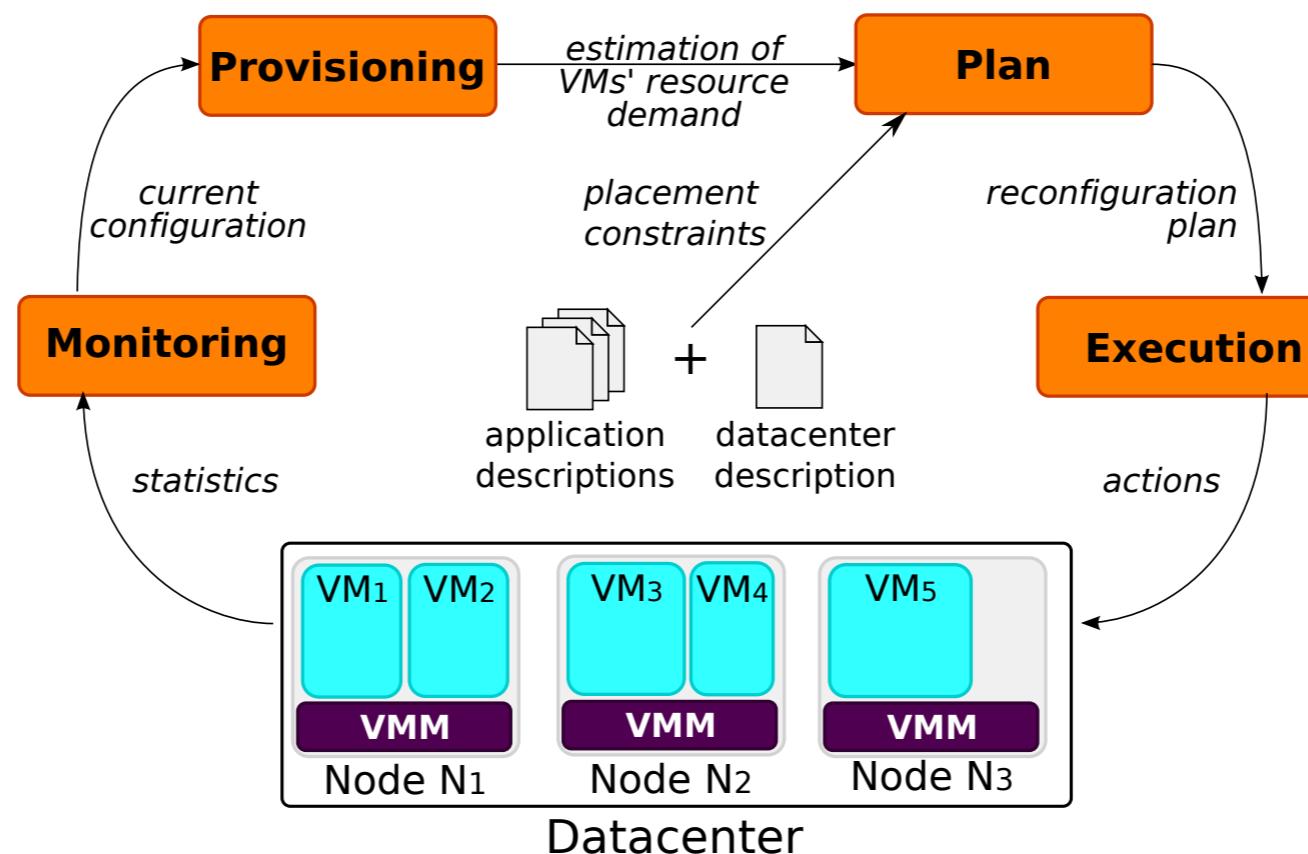
```
// Efficient synchronization  
latency ($T3 , $small );
```

Background - Entropy / btrPlace



An autonomic framework to maintain viable VE placements

Developed since 2006 (ANR SelfXL / ANR Emergence, 10 persons / EasyVirt)
ASCOLA Research Group (Mines Nantes)
Oasis Research Team (University Of Nice Sophia Antipolis)

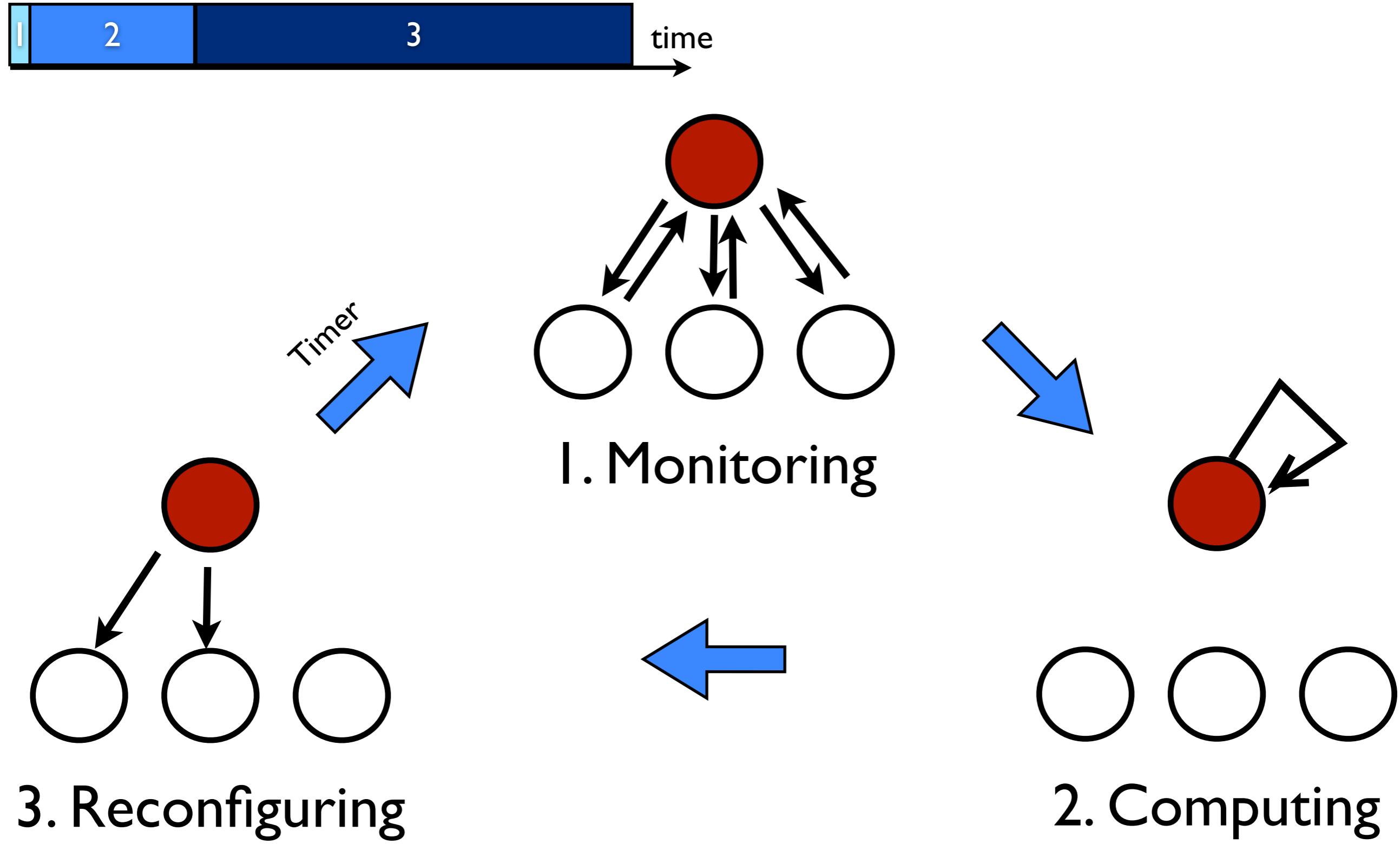


Scalability / Reactivity / Cost of reconfigurations

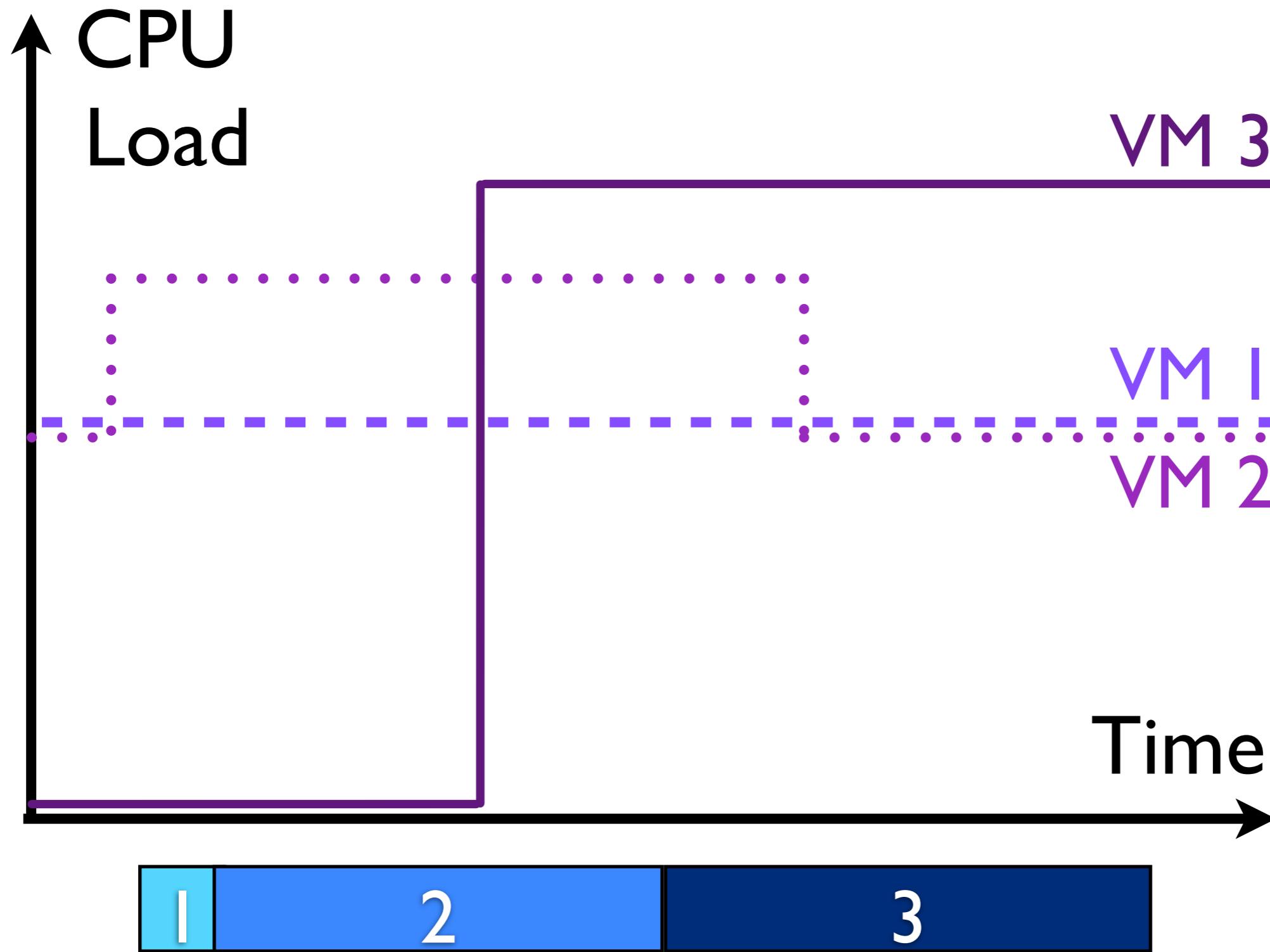


Dynamic Scheduling of VMs

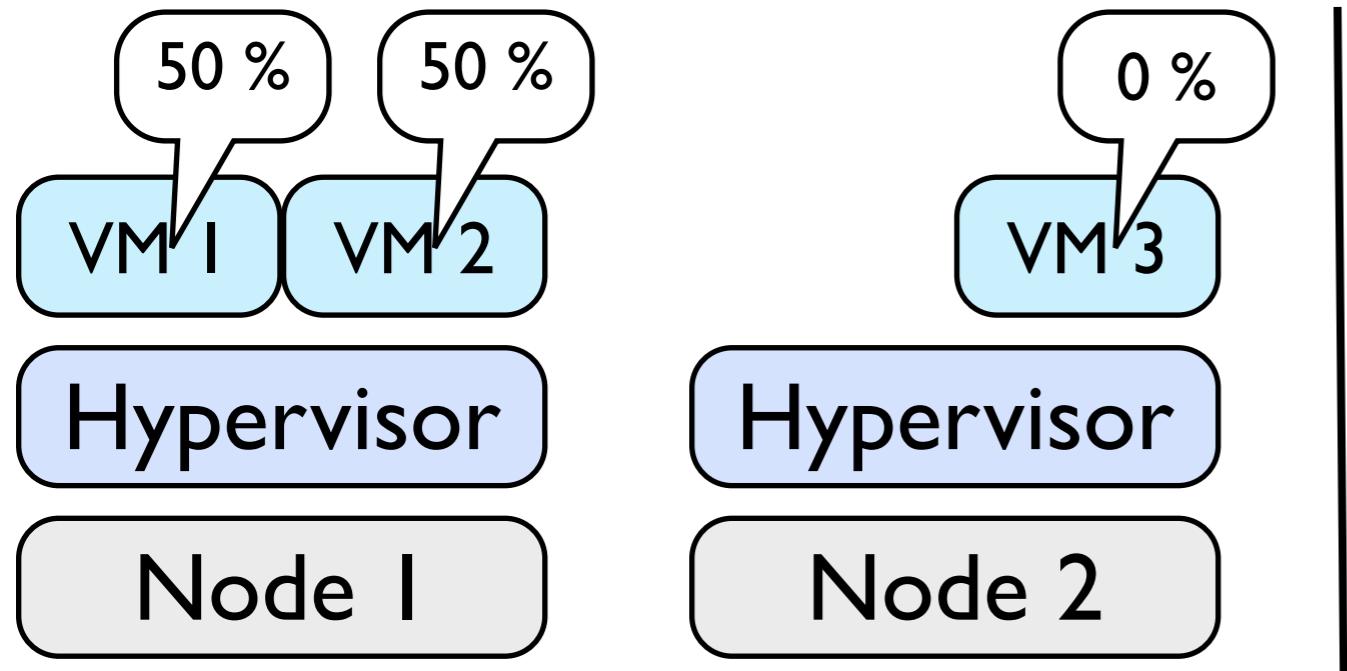
Centralized approach



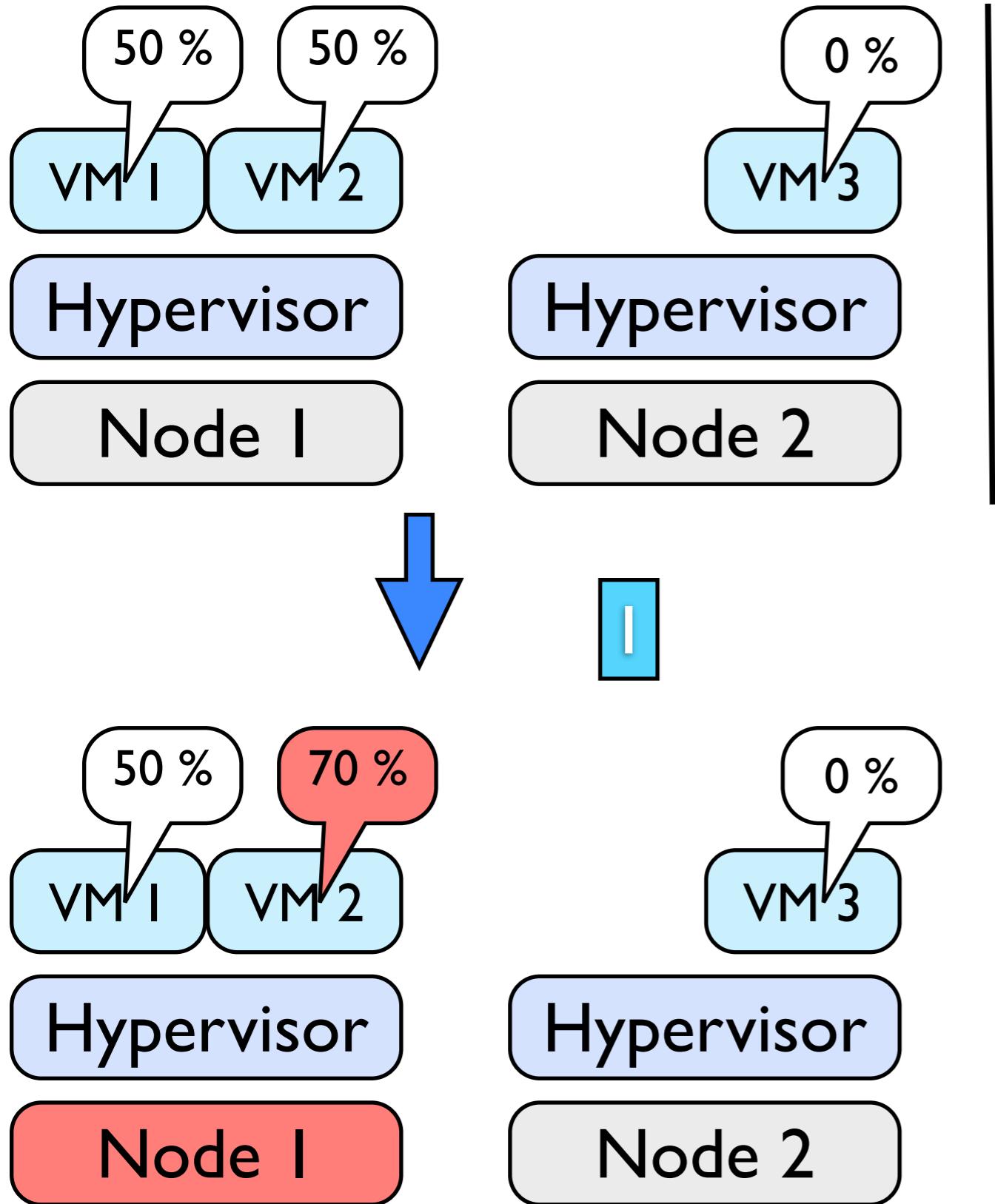
Scalability vs Reactivity



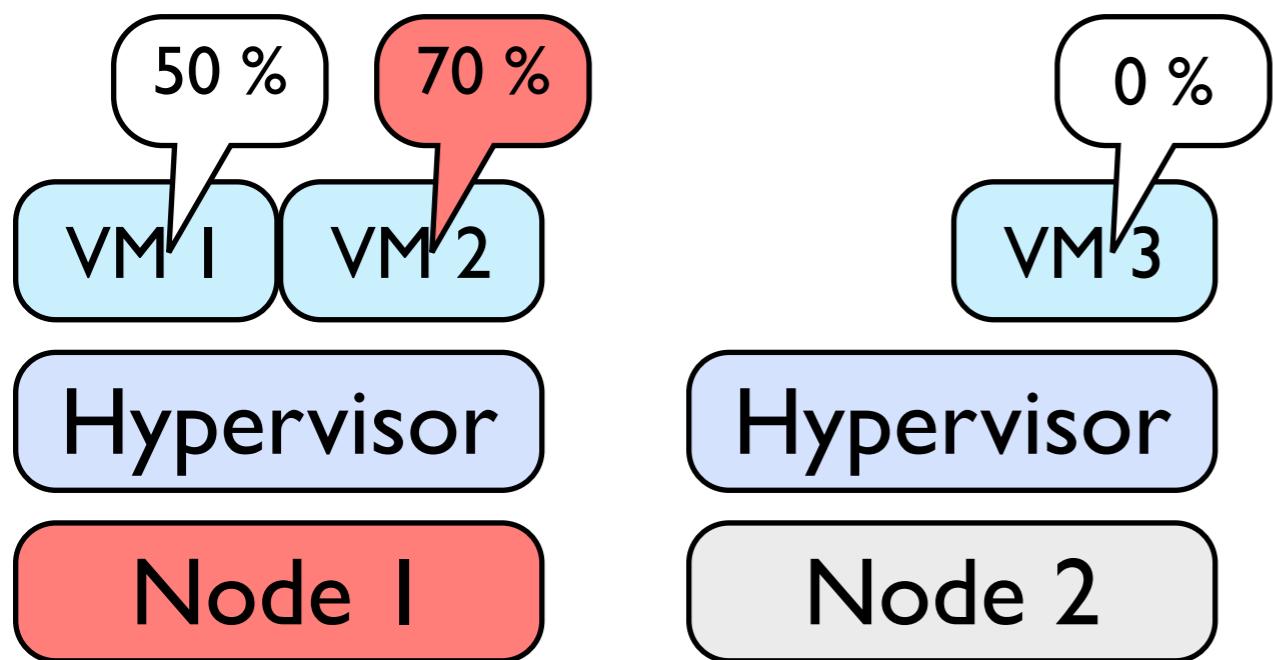
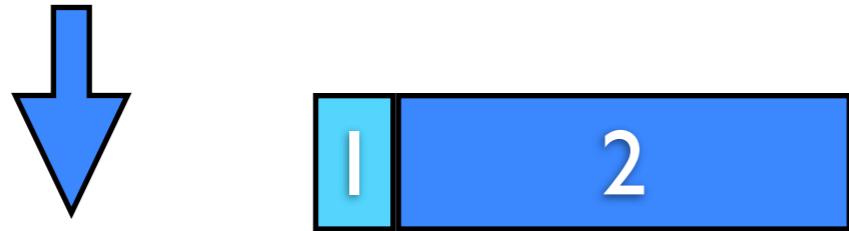
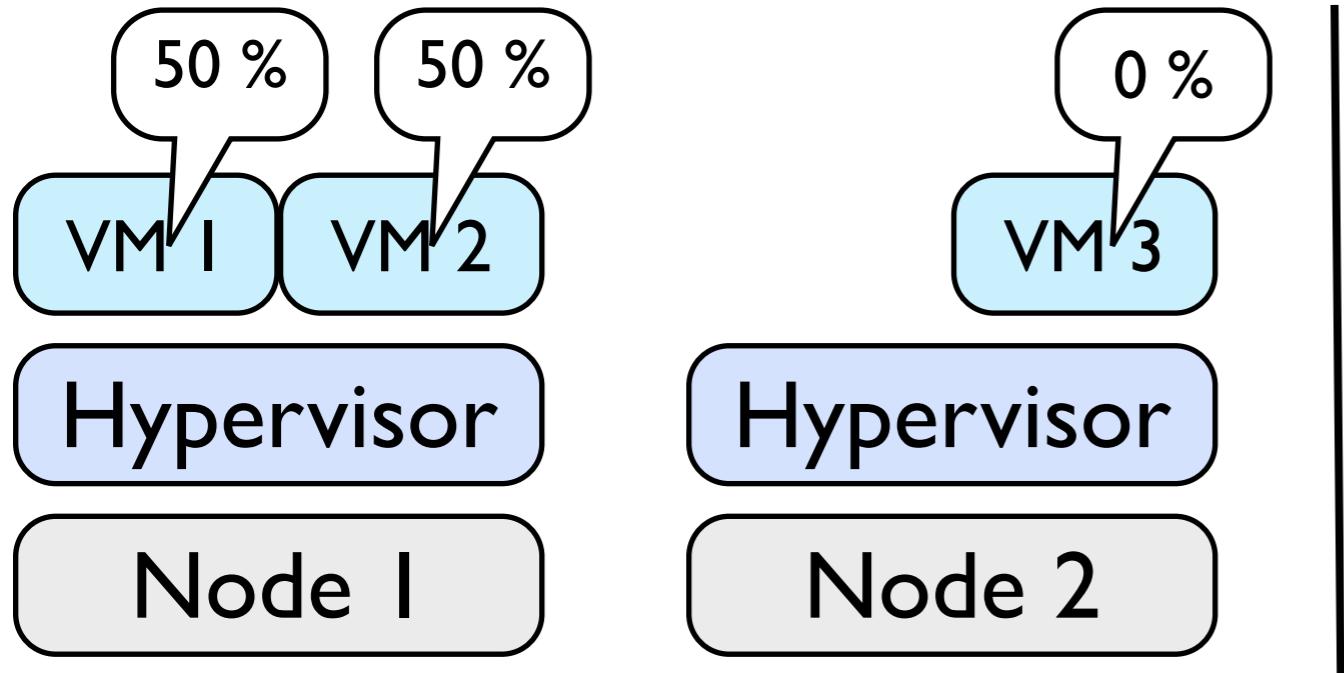
Scalability vs Reactivity



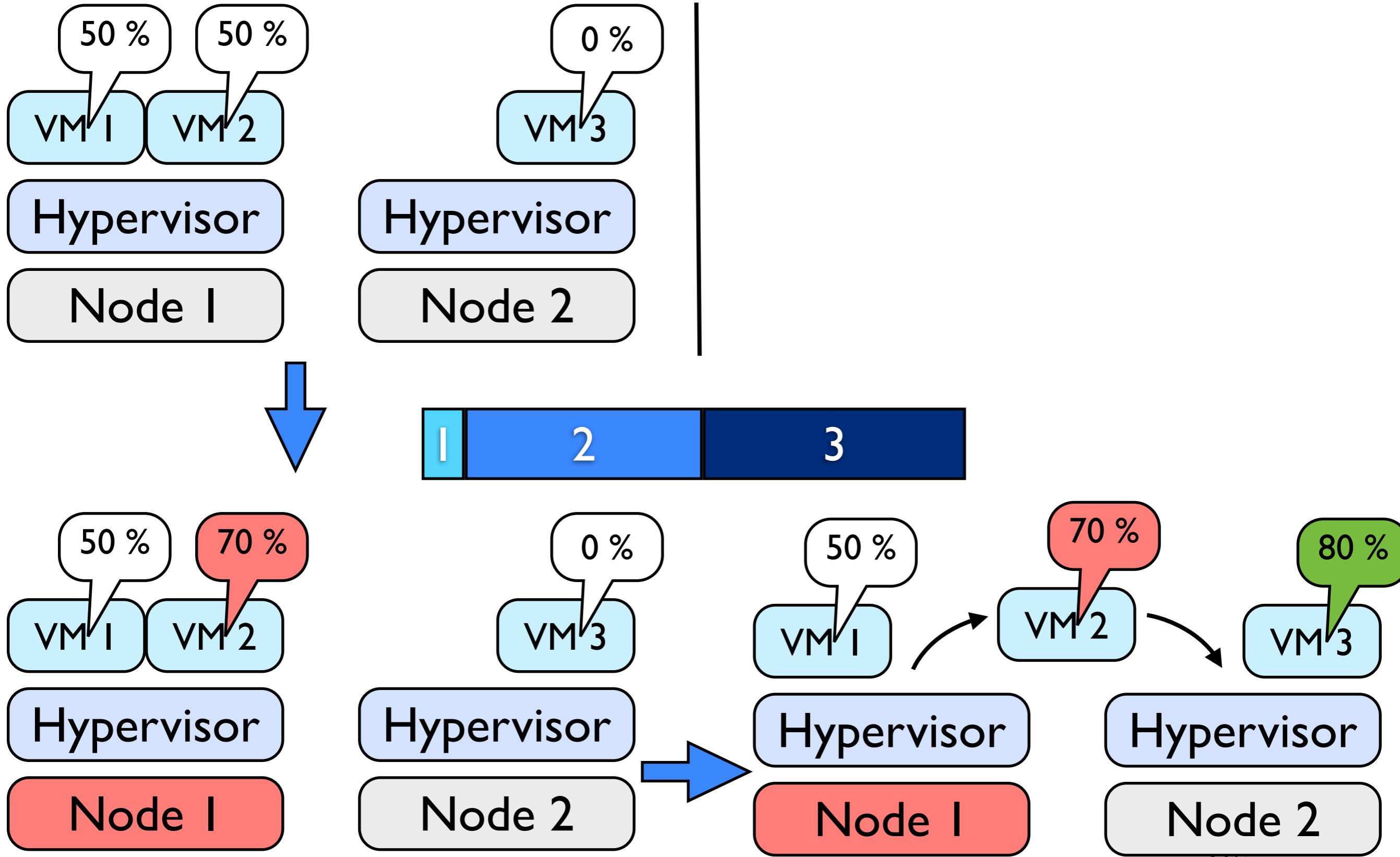
Scalability vs Reactivity



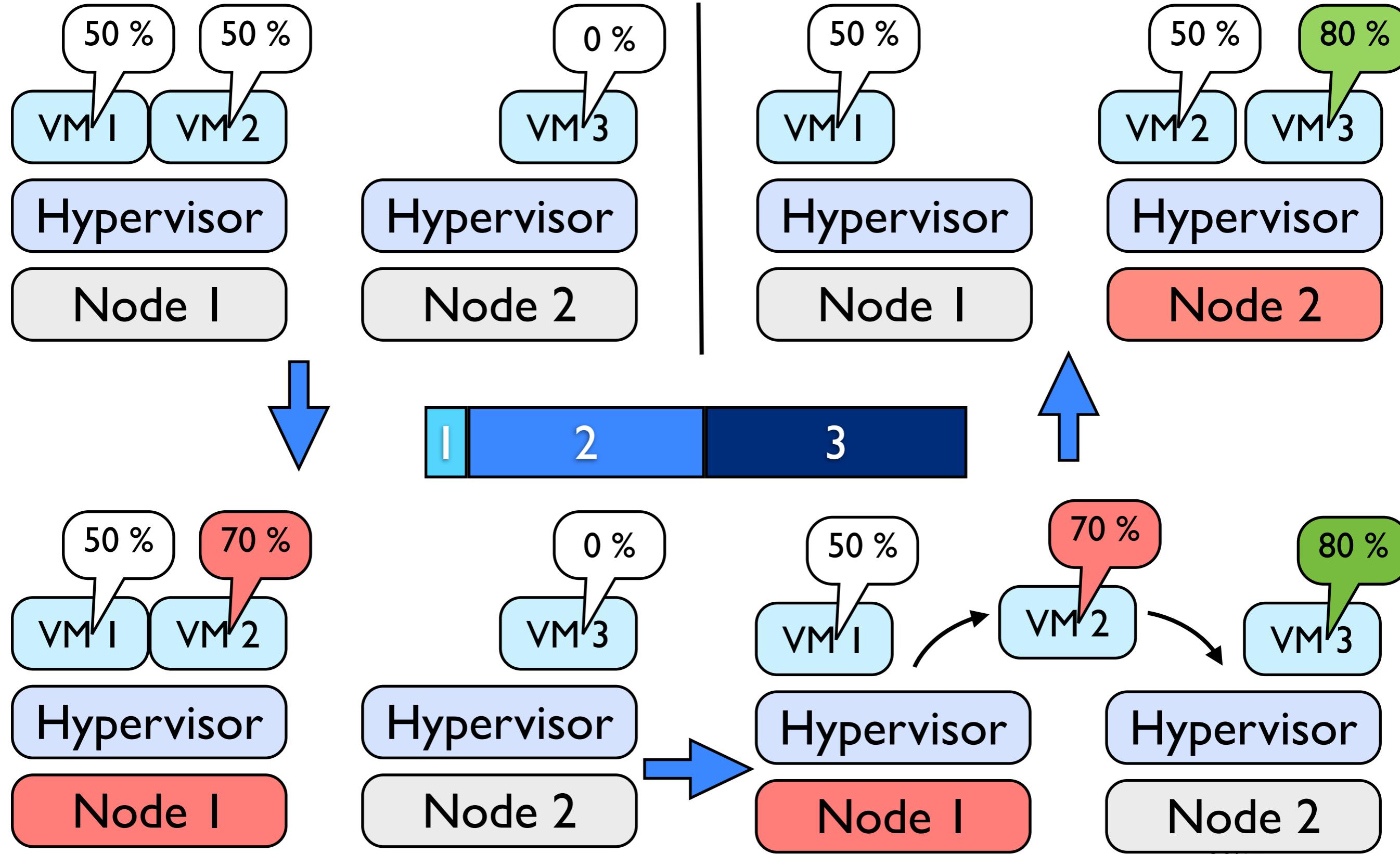
Scalability vs Reactivity



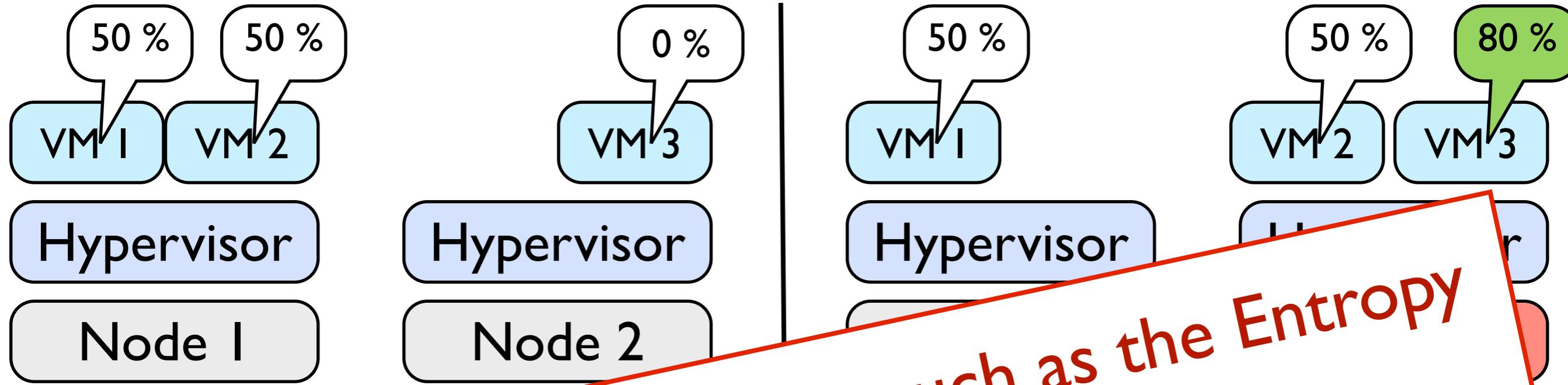
Scalability vs Reactivity



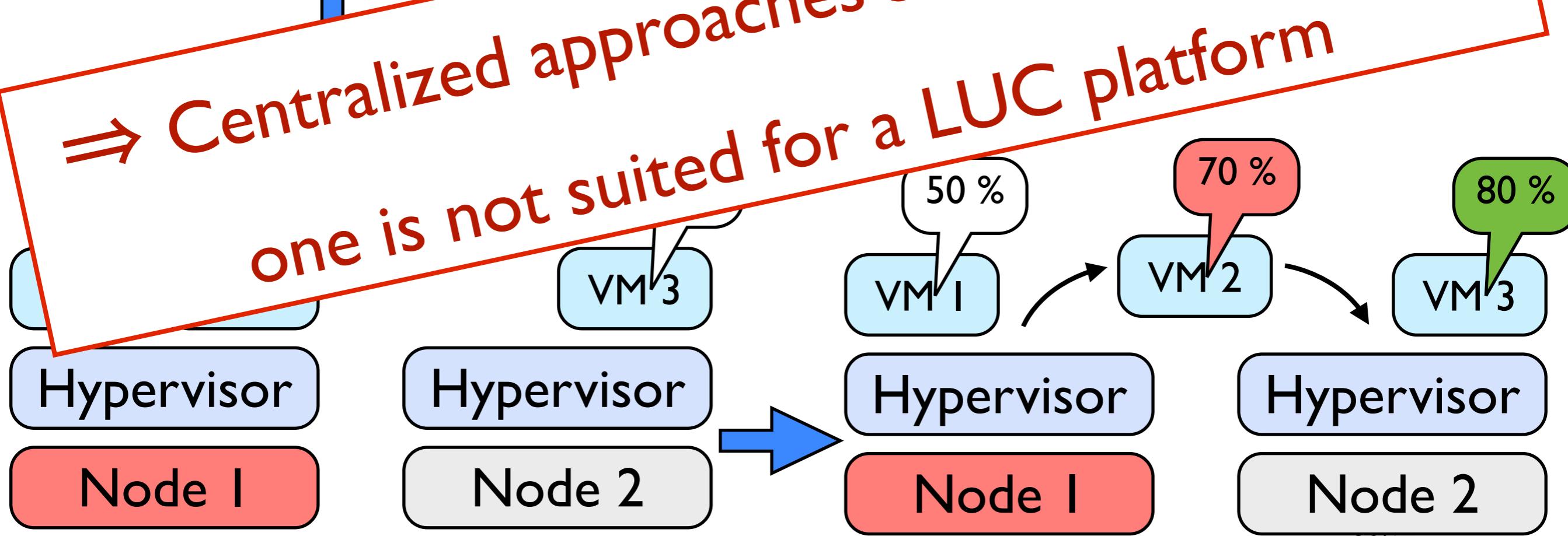
Scalability vs Reactivity

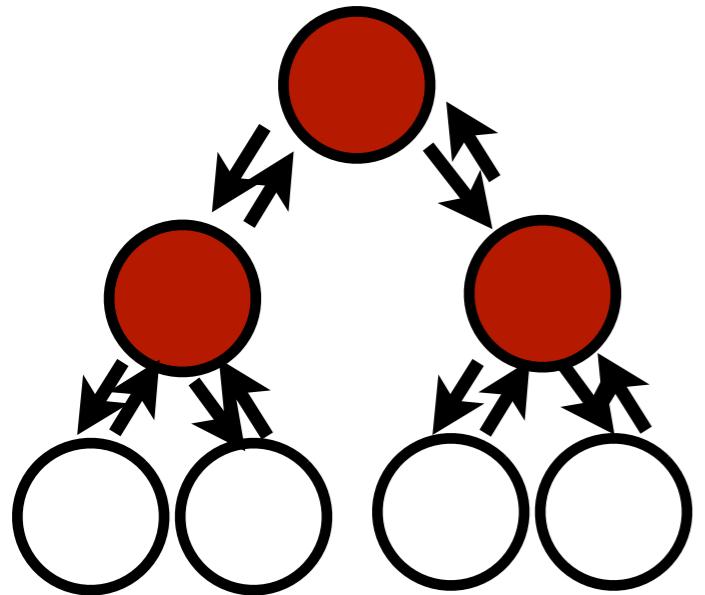


Scalability vs Reactivity



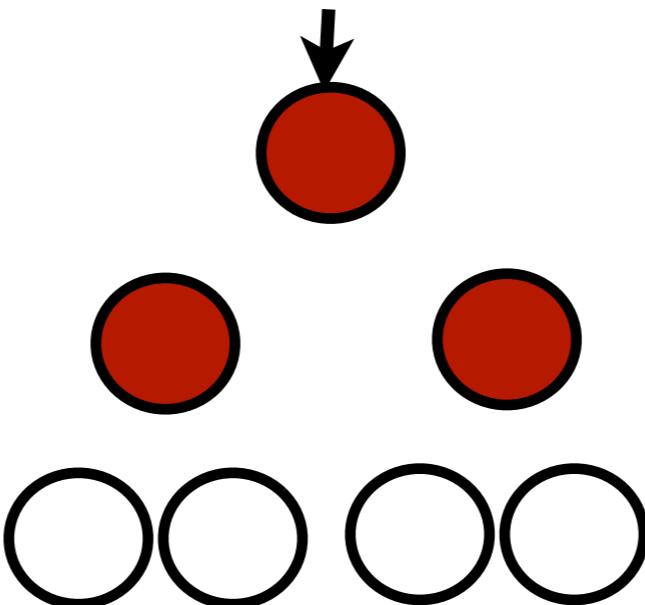
⇒ Centralized approaches such as the Entropy



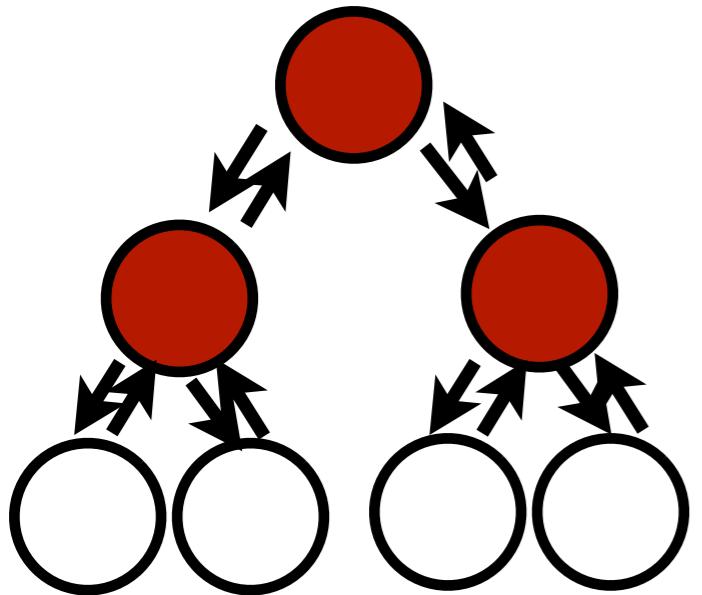


Monitoring

Hierarchical architecture
Snooze [Feller et al., CCGRID'12]

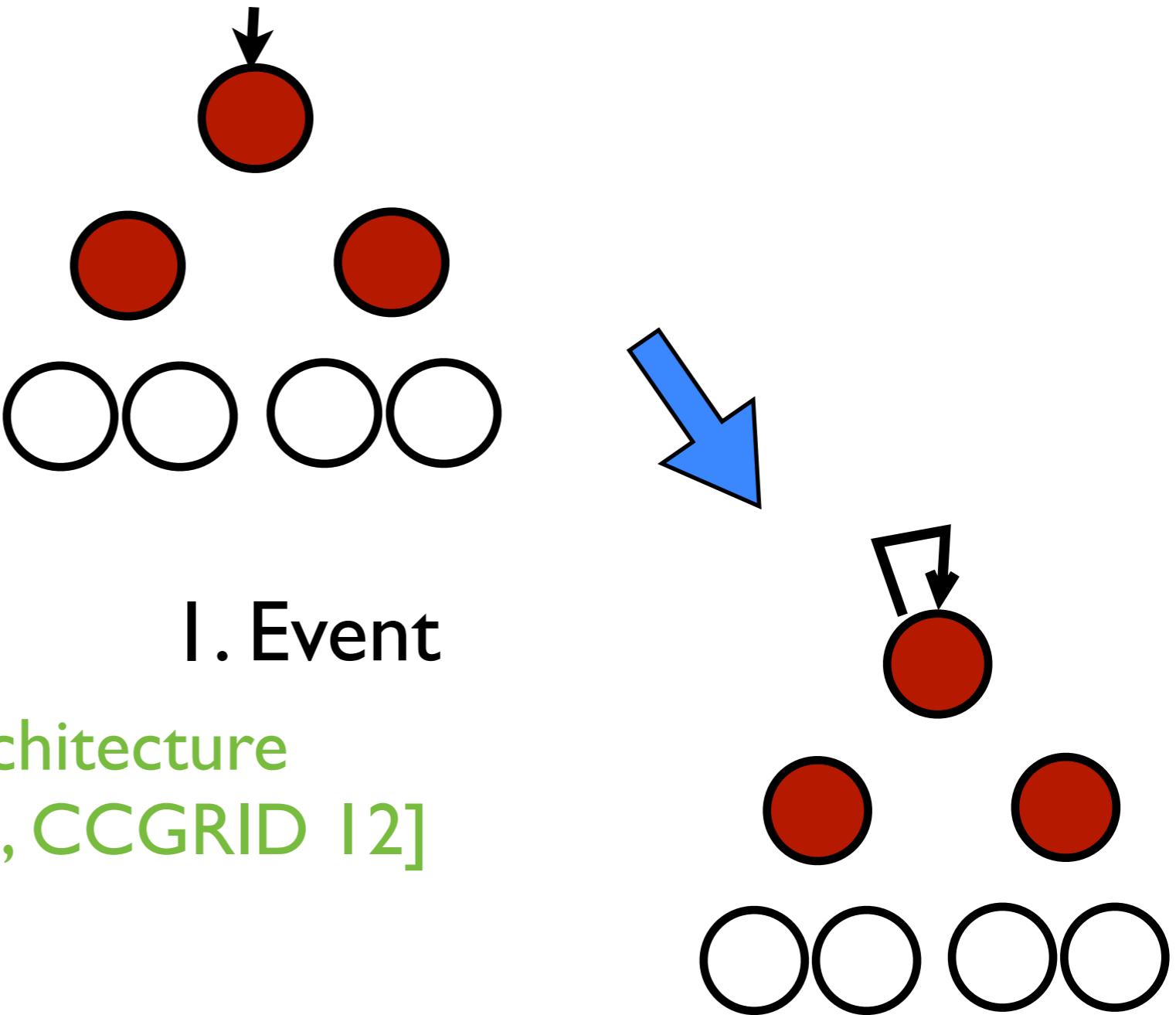


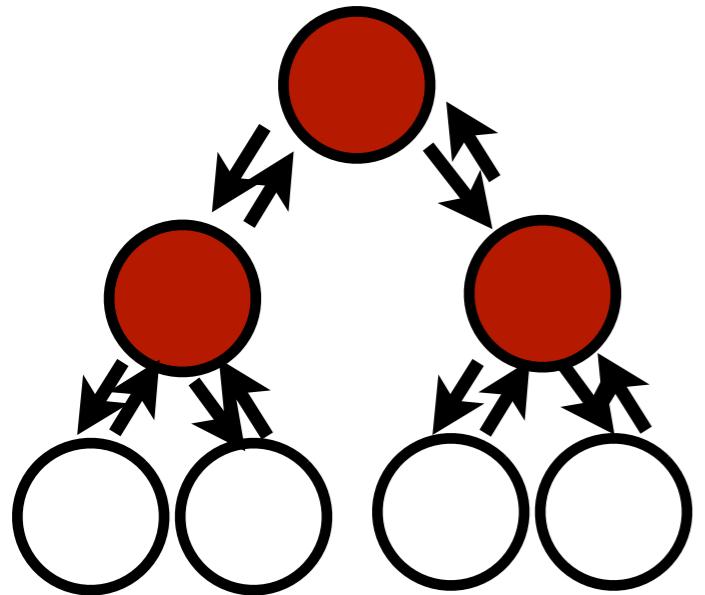
I. Event



Monitoring

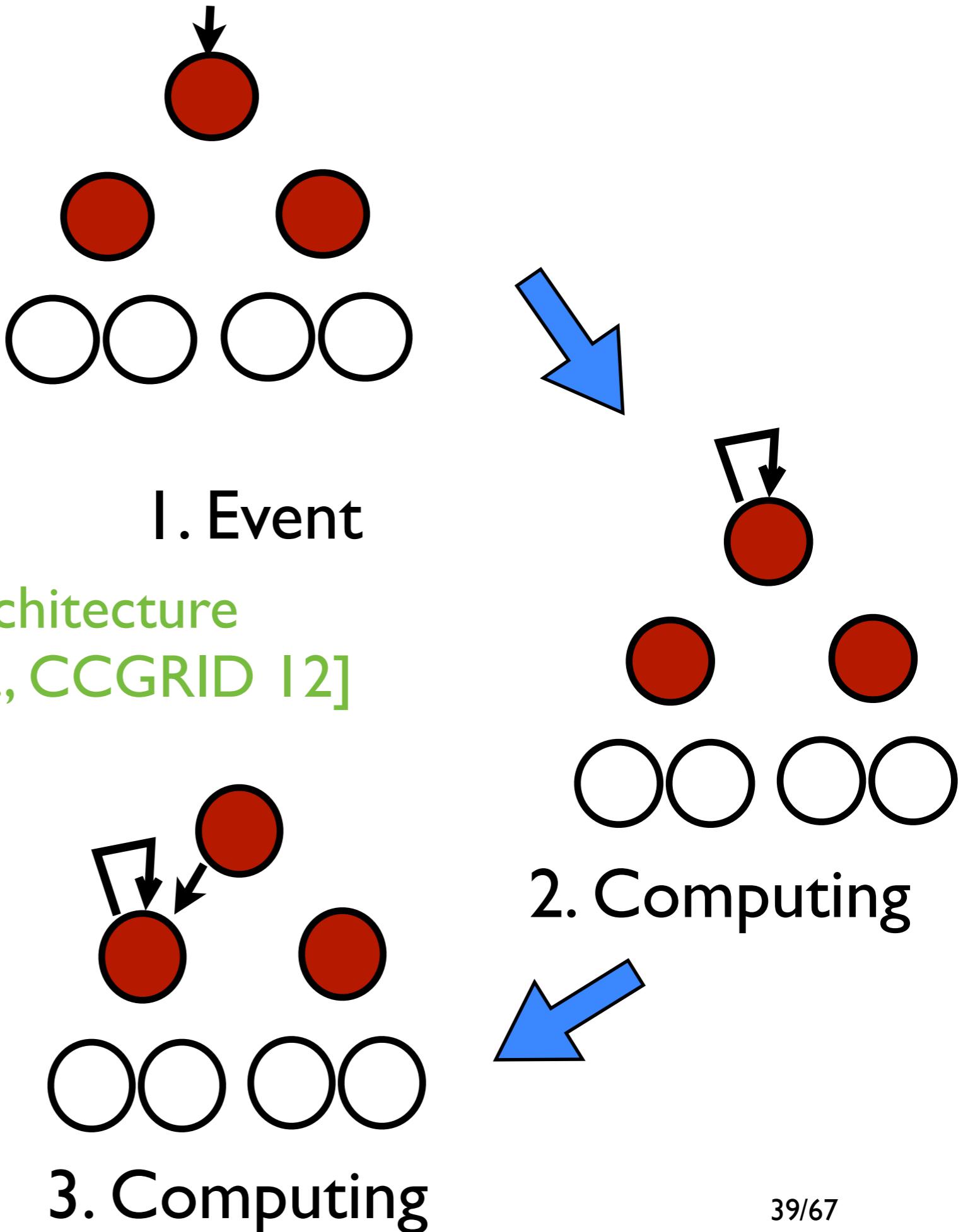
Hierarchical architecture
Snooze [Feller et al., CCGRID 12]

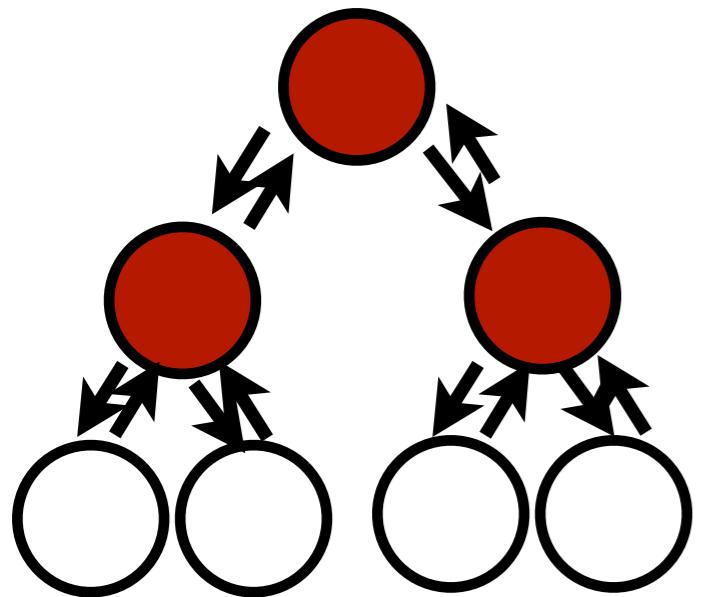




Monitoring

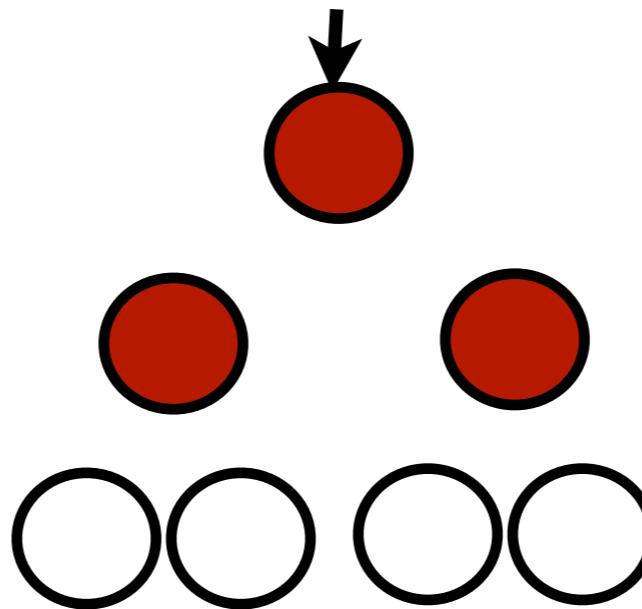
Hierarchical architecture
Snooze [Feller et al., CCGRID 12]



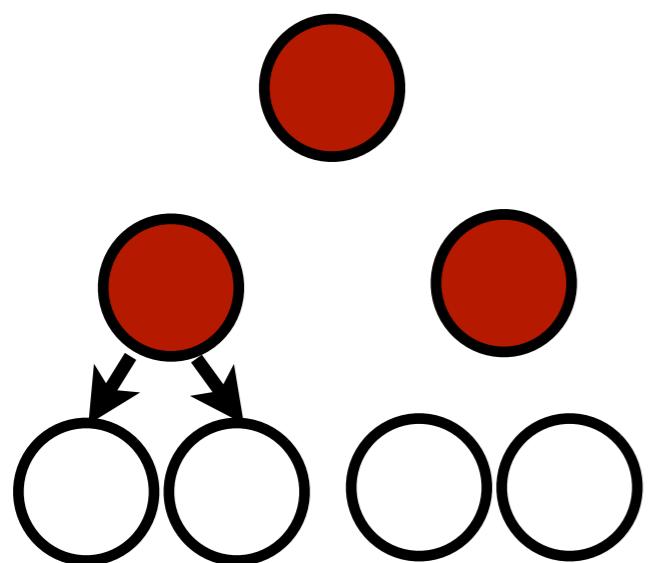


Monitoring

Hierarchical architecture
Snooze [Feller et al., CCGRID 12]

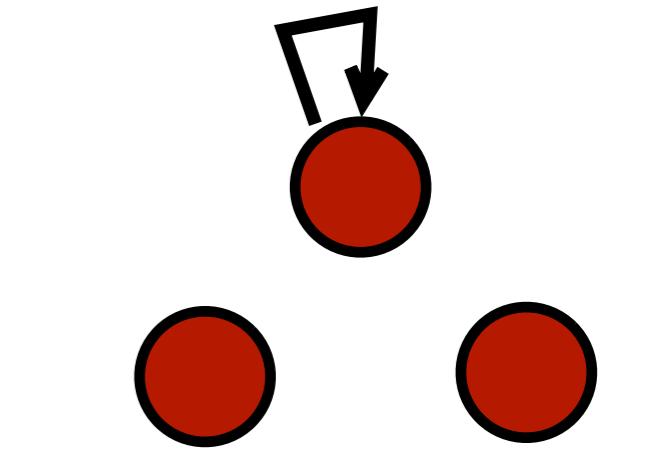


I. Event

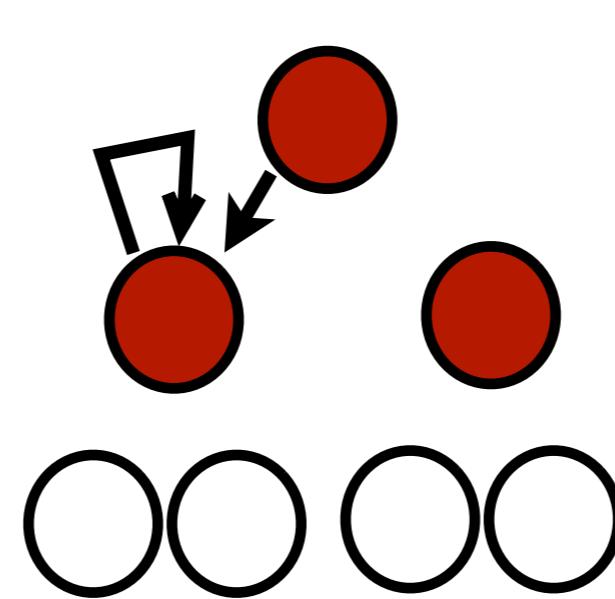


4. Reconfiguring

I. Event



2. Computing



3. Computing

39/67

The LUC OS - VEs Scheduling

- Make dynamic partitioning of the system according to the effective usage of resources
- Make direct cooperations between hypervisors (no service node)
- The DVMS Proposal

Event driven

P2P Like system

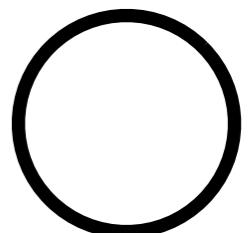
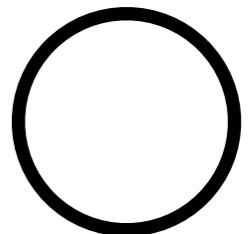
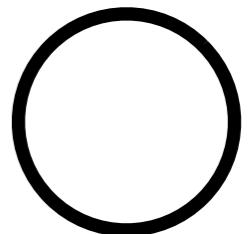
Local interactions between nodes

Scheduling performed on partitions of the system, created dynamically (nodes are reserved for an exclusive use by a scheduler, to prevent several schedulers from migrating the same VMs)

The LUC OS - VEs Scheduling

Event occurs on node_i

Event



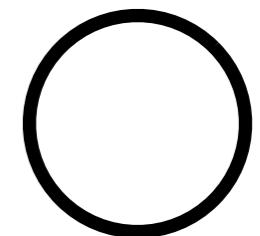
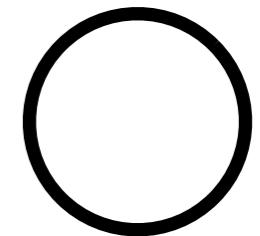
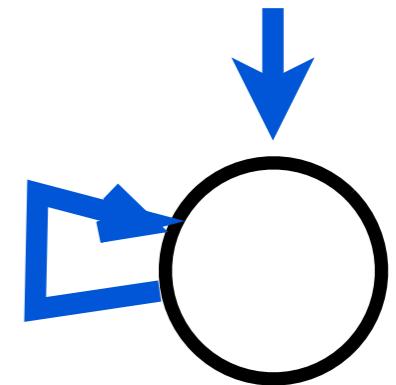
The LUC OS - VEs Scheduling

Event occurs on node_i



Can current node scheduler
calculate valid schedule?

Event



The LUC OS - VEs Scheduling

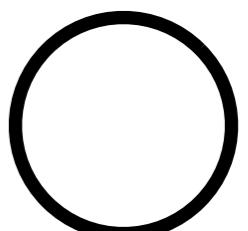
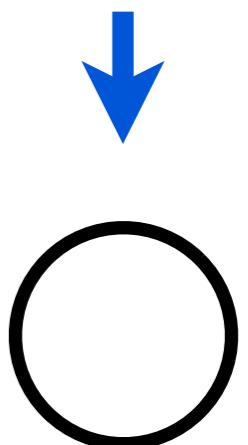
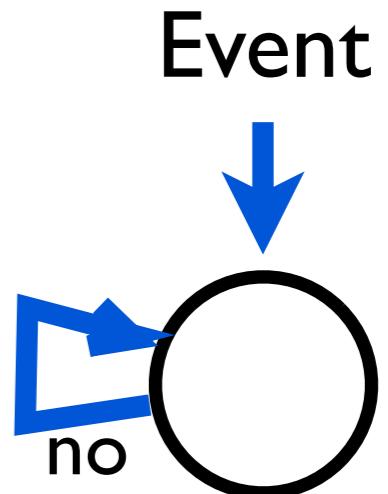
Event occurs on node_i



Can current node scheduler
calculate valid schedule?



Contact neighbor
and ask it to solve
the problem



41/67

The LUC OS - VEs Scheduling

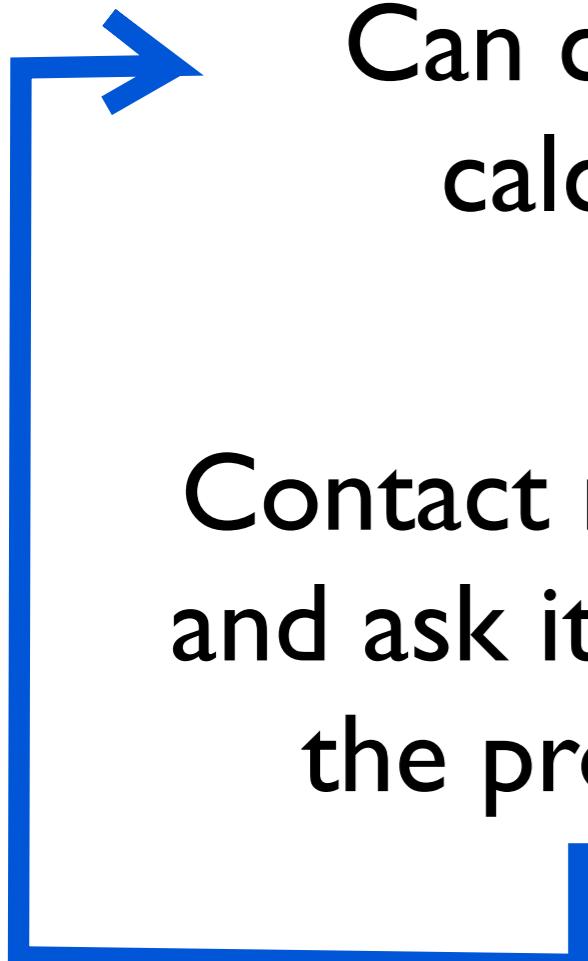
Event occurs on node_i



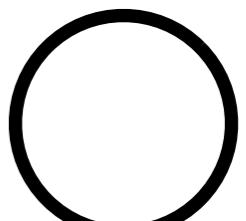
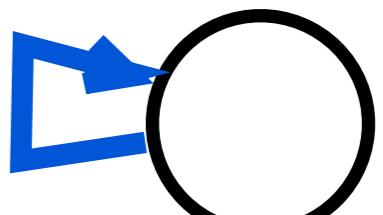
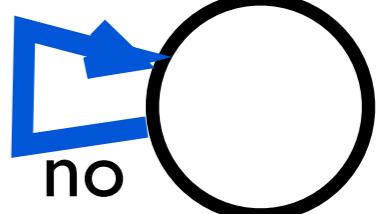
Can current node scheduler
calculate valid schedule?



Contact neighbor
and ask it to solve
the problem



Event



41/67

The LUC OS - VEs Scheduling

Event occurs on node_i



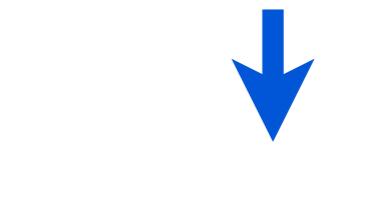
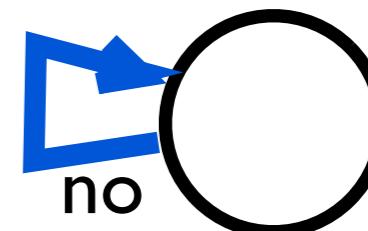
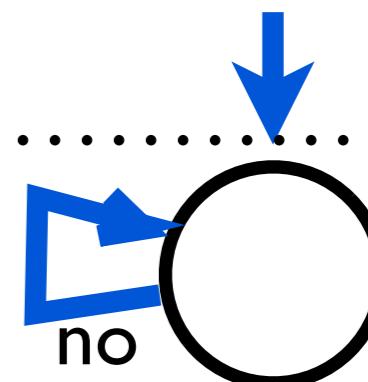
Can current node scheduler
calculate valid schedule?



Contact neighbor
and ask it to solve
the problem



Event



41/67

The LUC OS - VEs Scheduling

Event occurs on node_i

Can current node scheduler calculate valid schedule?

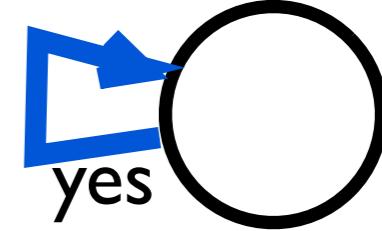
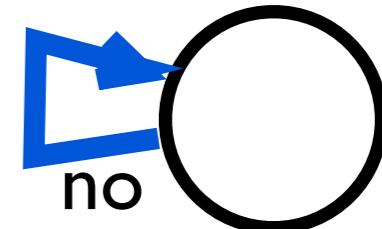
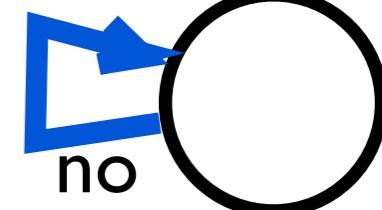
no

Contact neighbor and ask it to solve the problem

yes

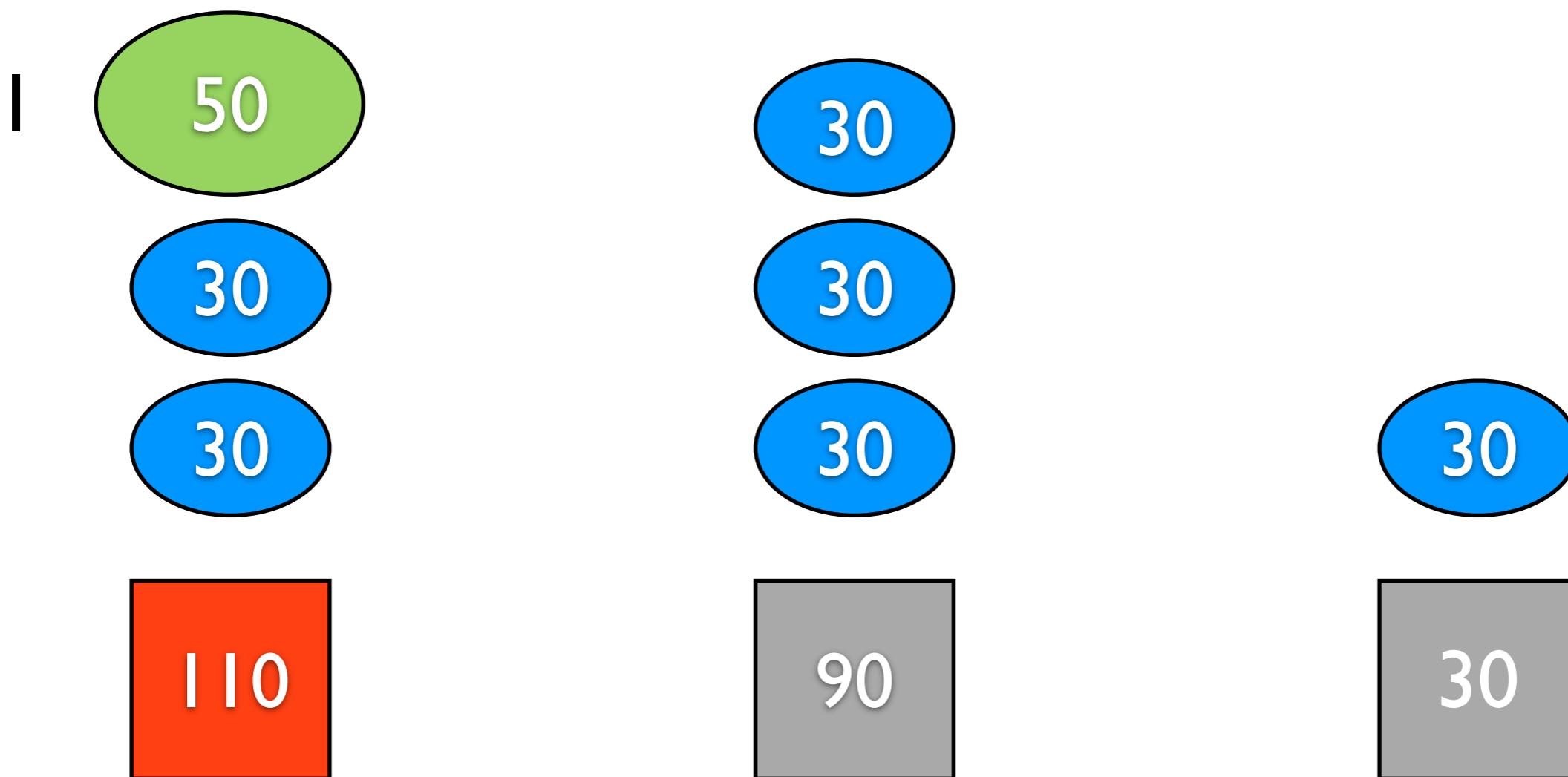
Apply the schedule

Event

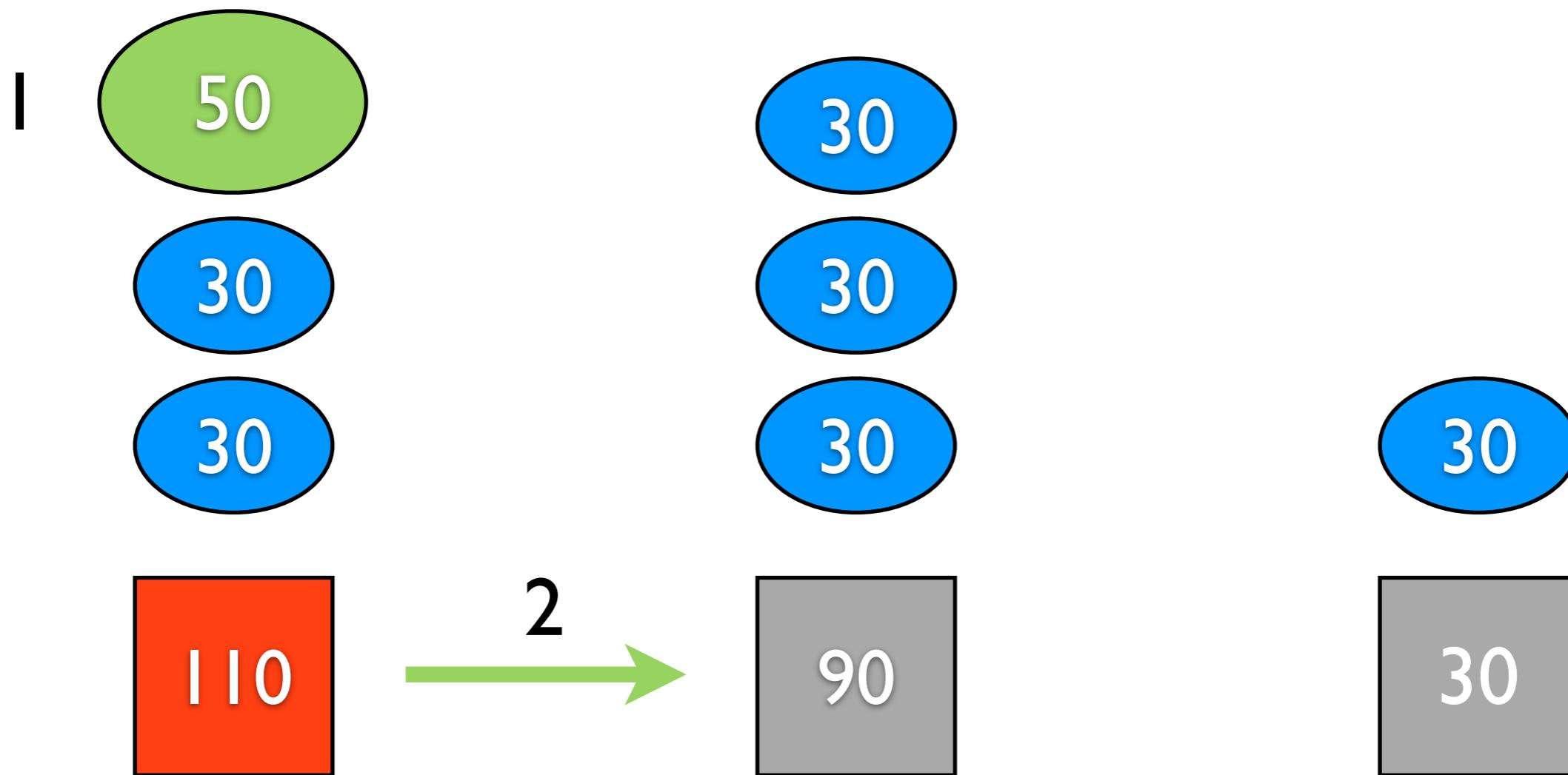


41/67

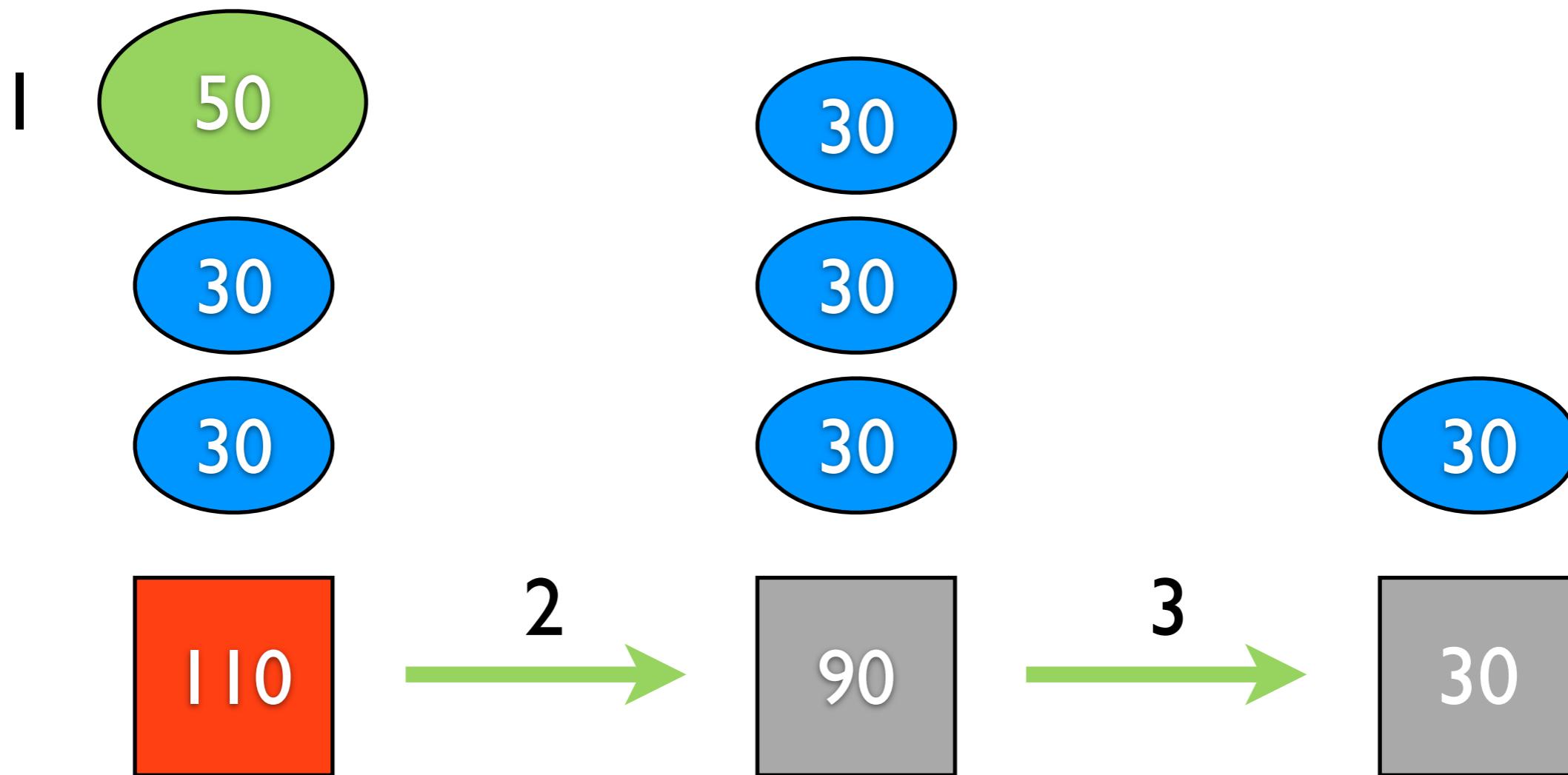
Example I: overloaded event



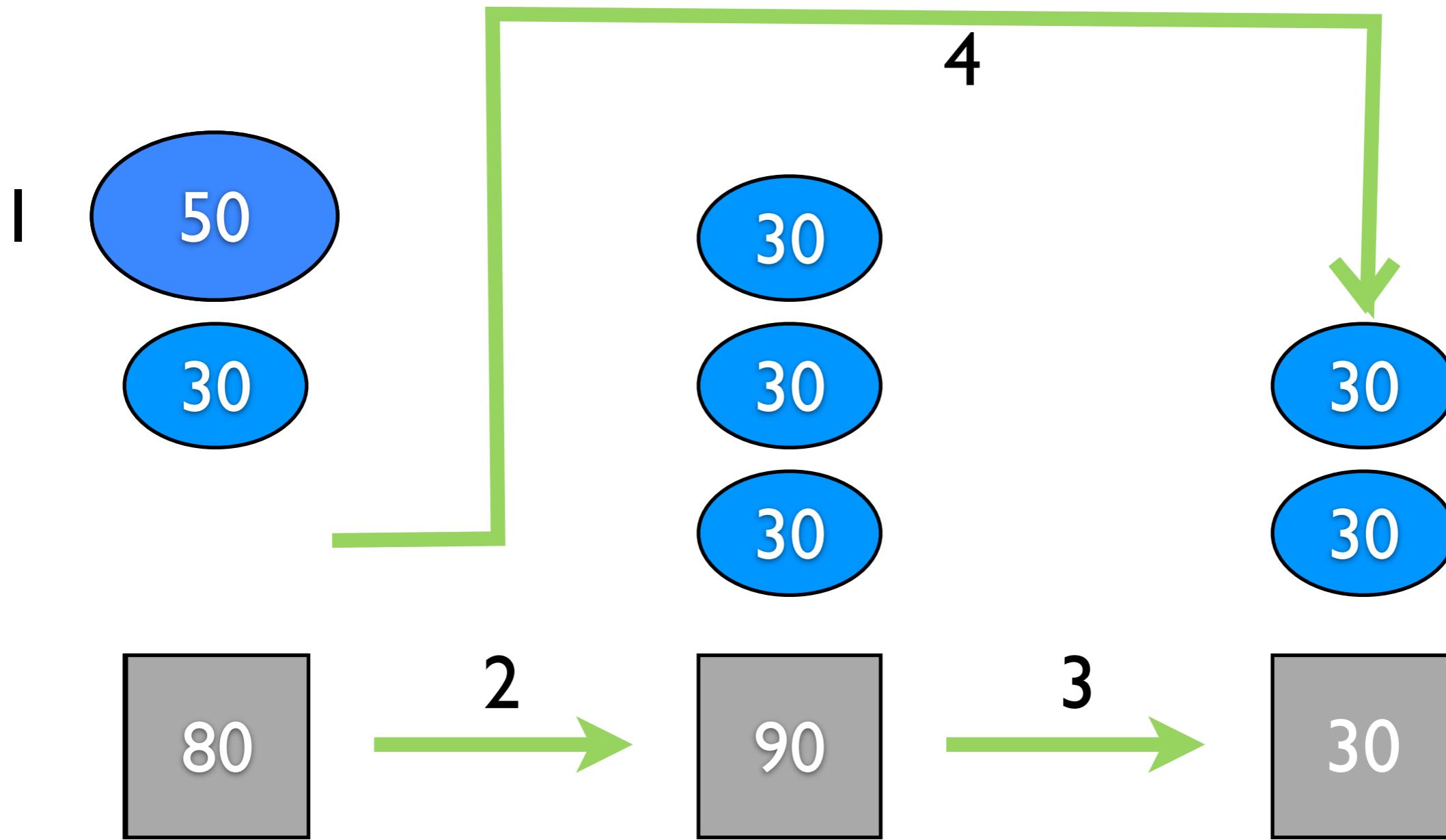
Example I: overloaded event



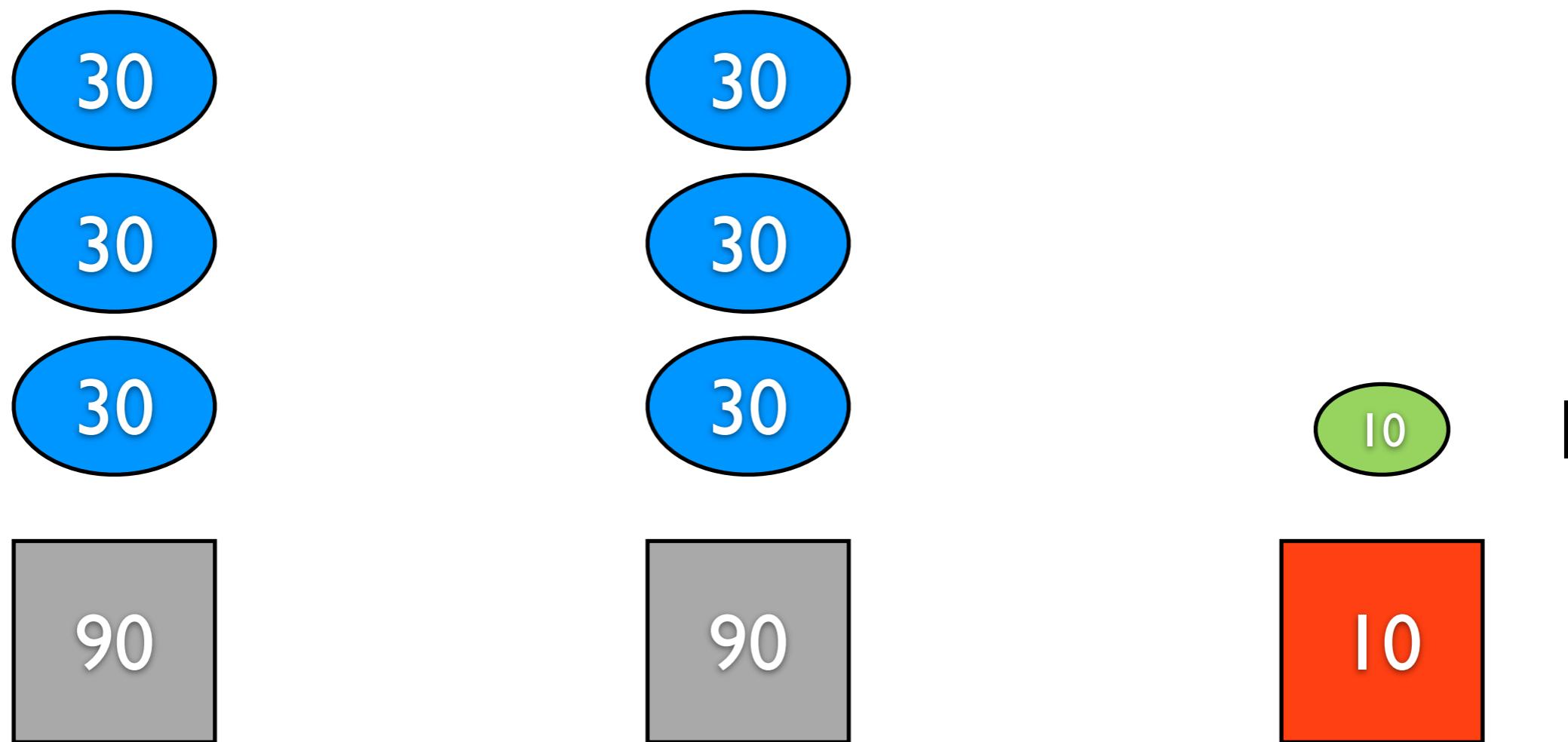
Example I: overloaded event



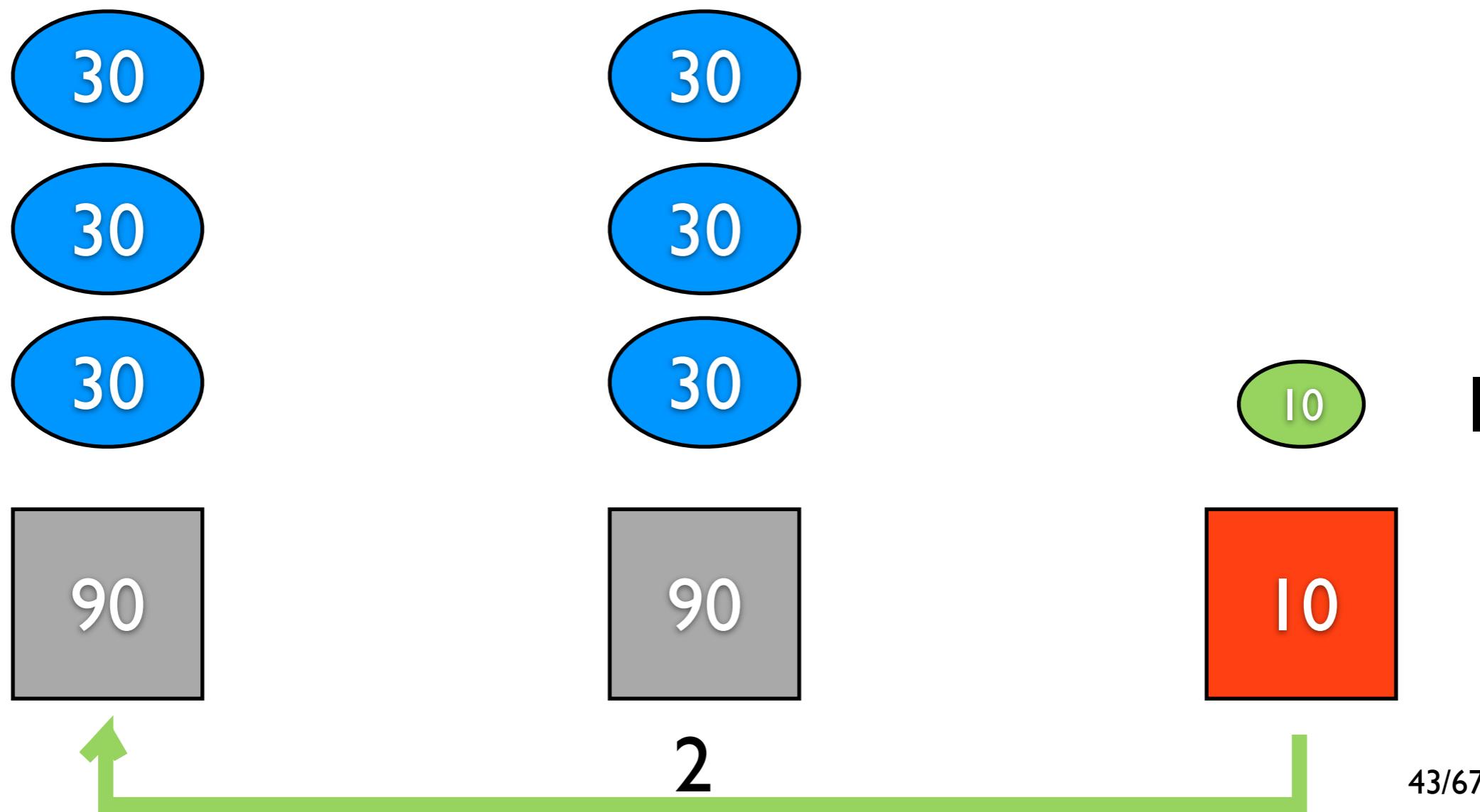
Example I: overloaded event



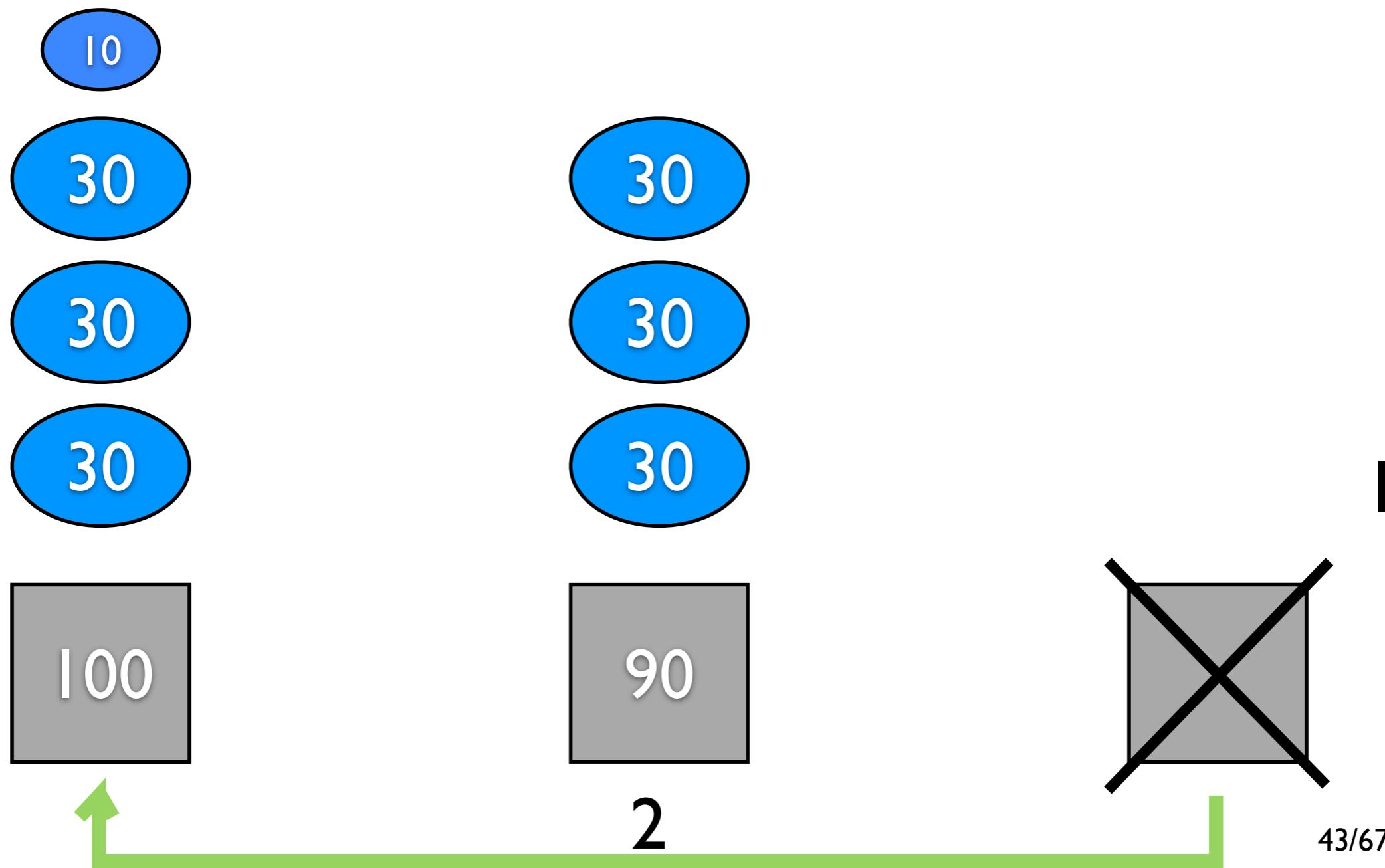
Example 2: underloaded event



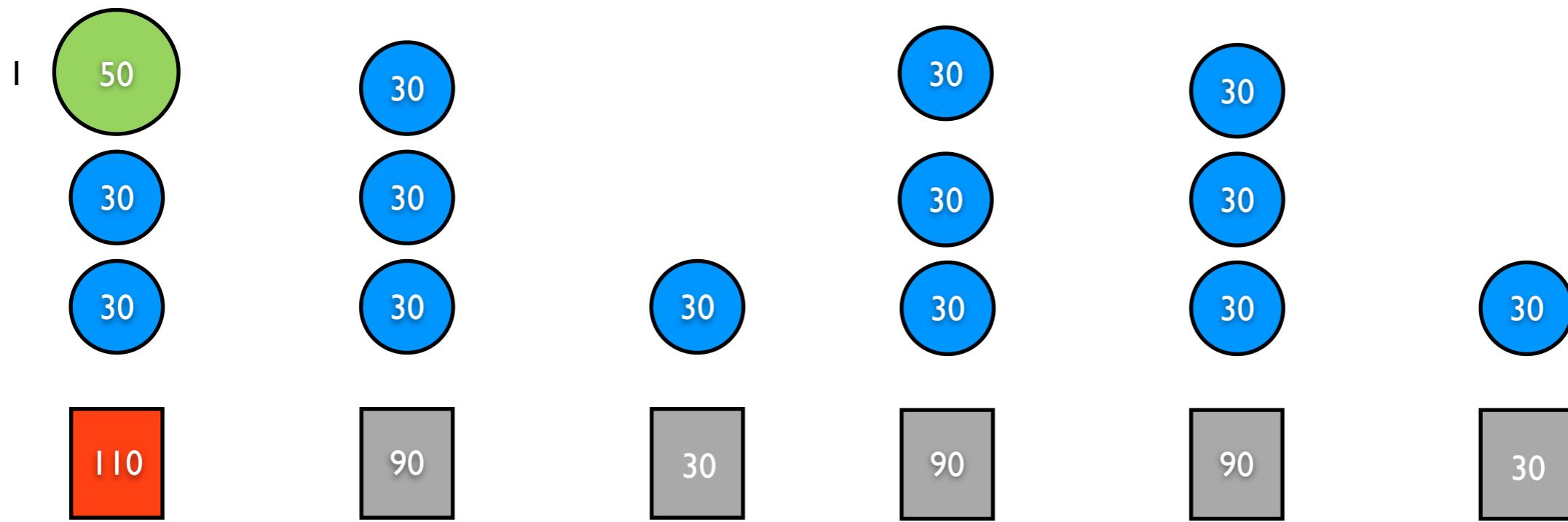
Example 2: underloaded event



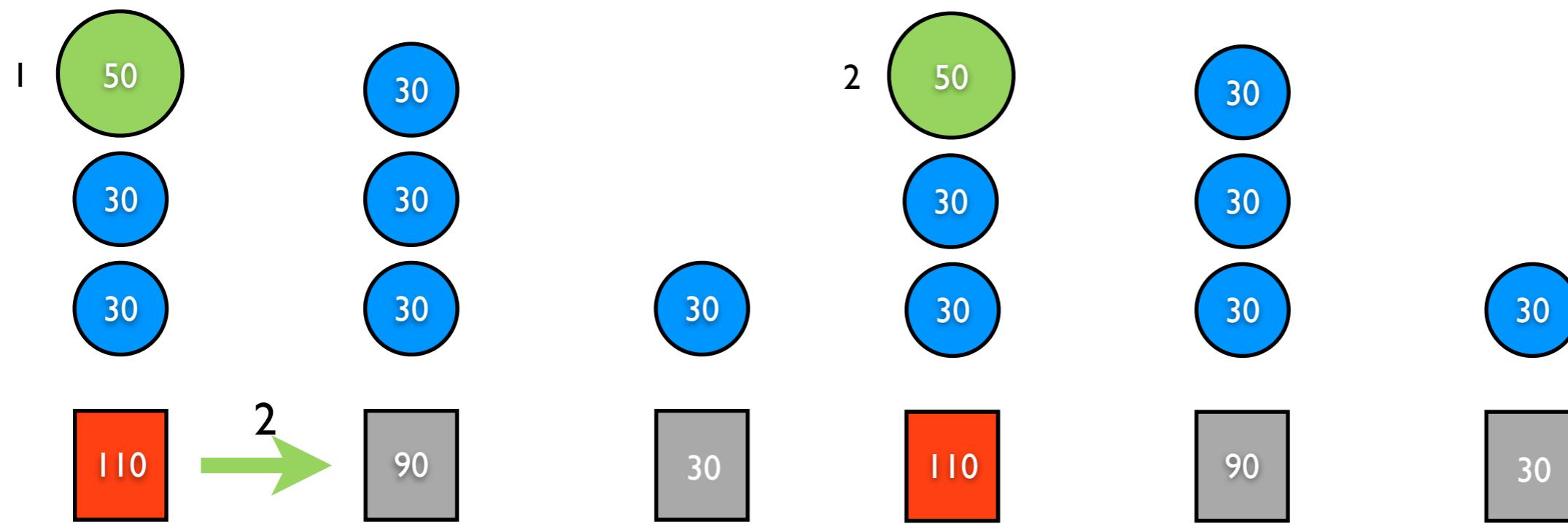
Example 2: underloaded event



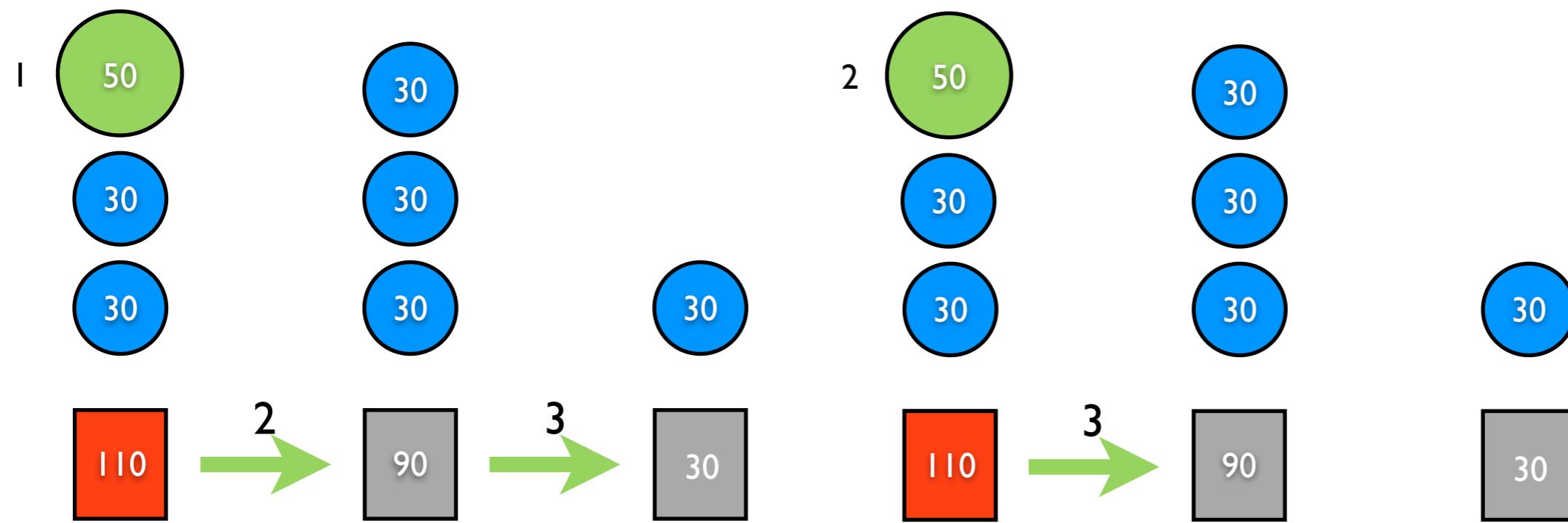
Example 3: shifted overloaded events



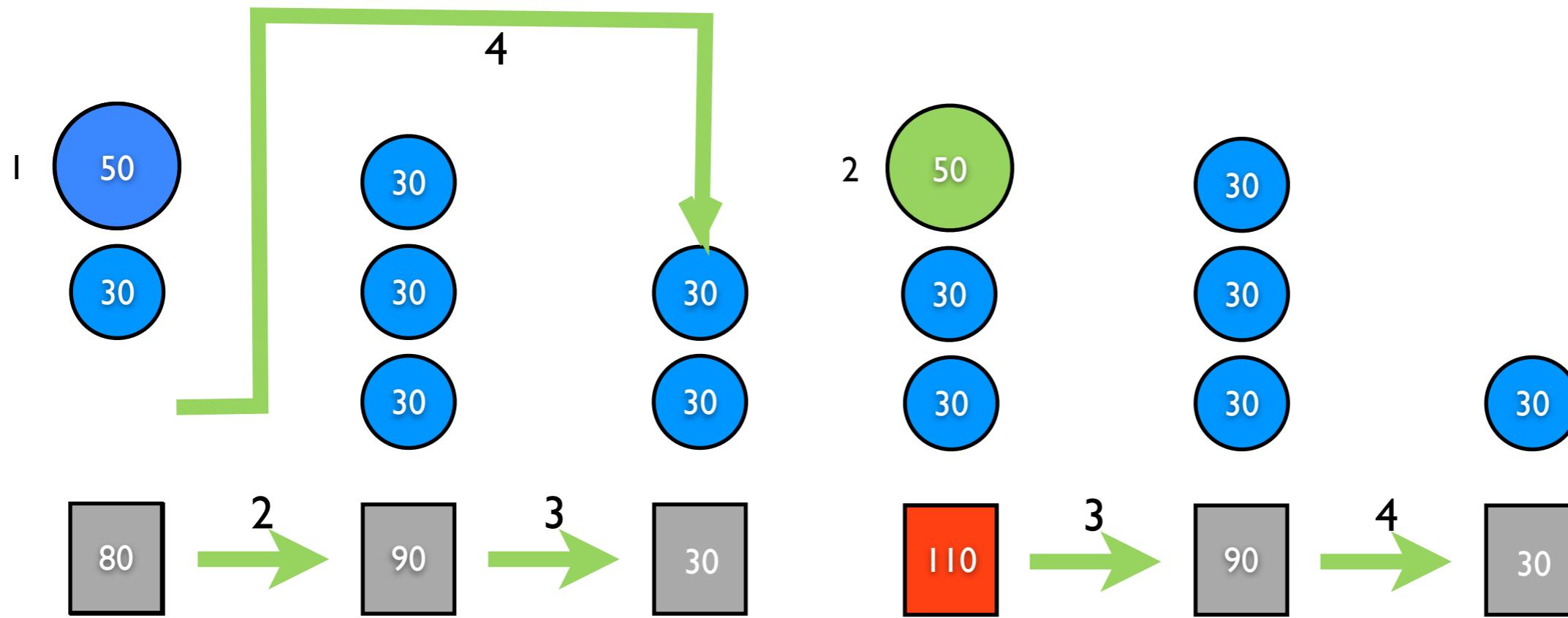
Example 3: shifted overloaded events



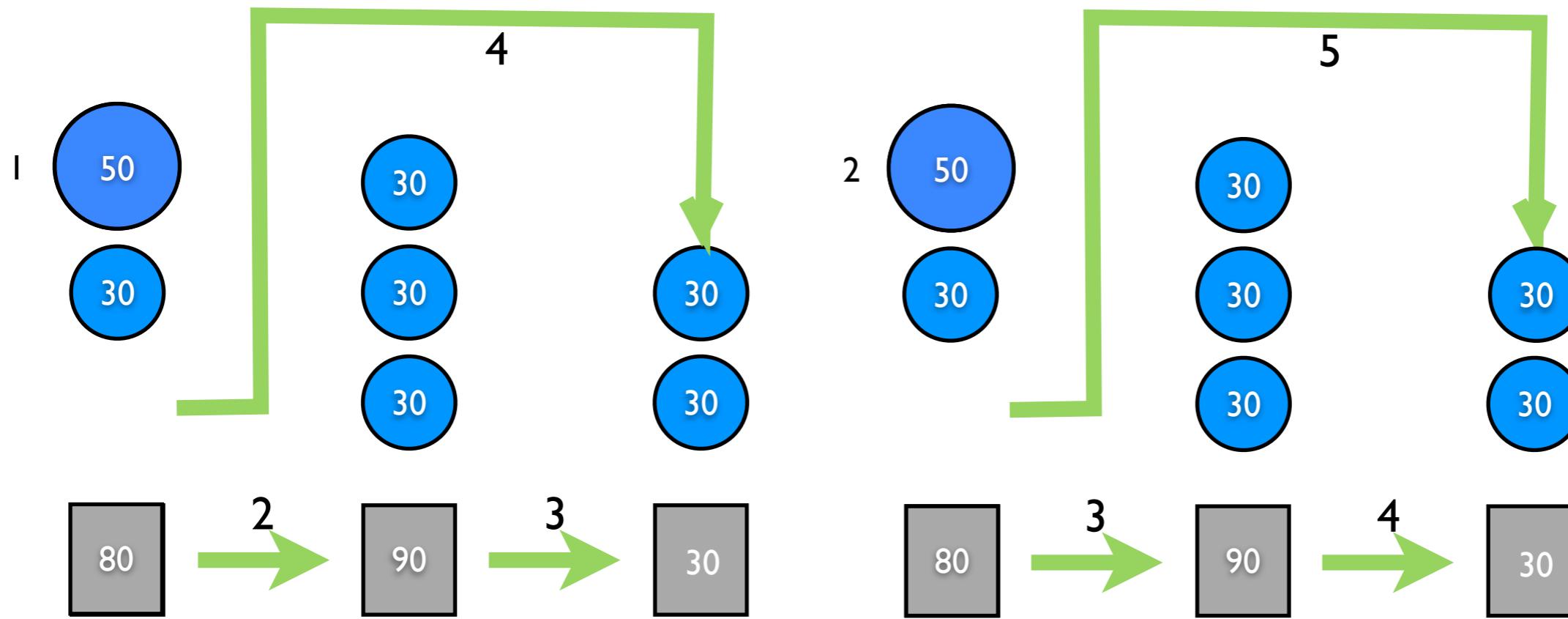
Example 3: shifted overloaded events



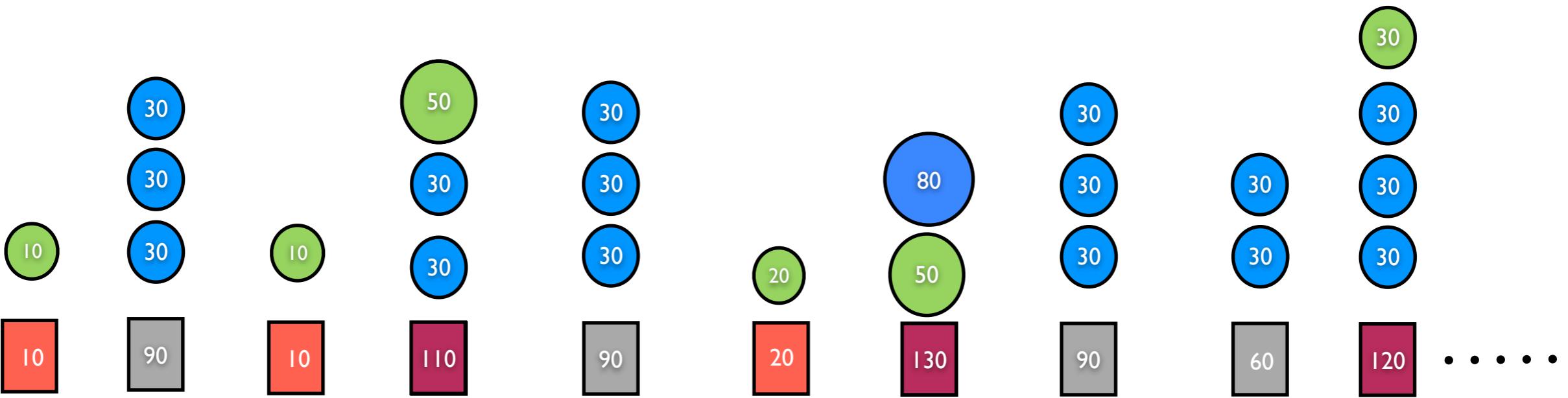
Example 3: shifted overloaded events



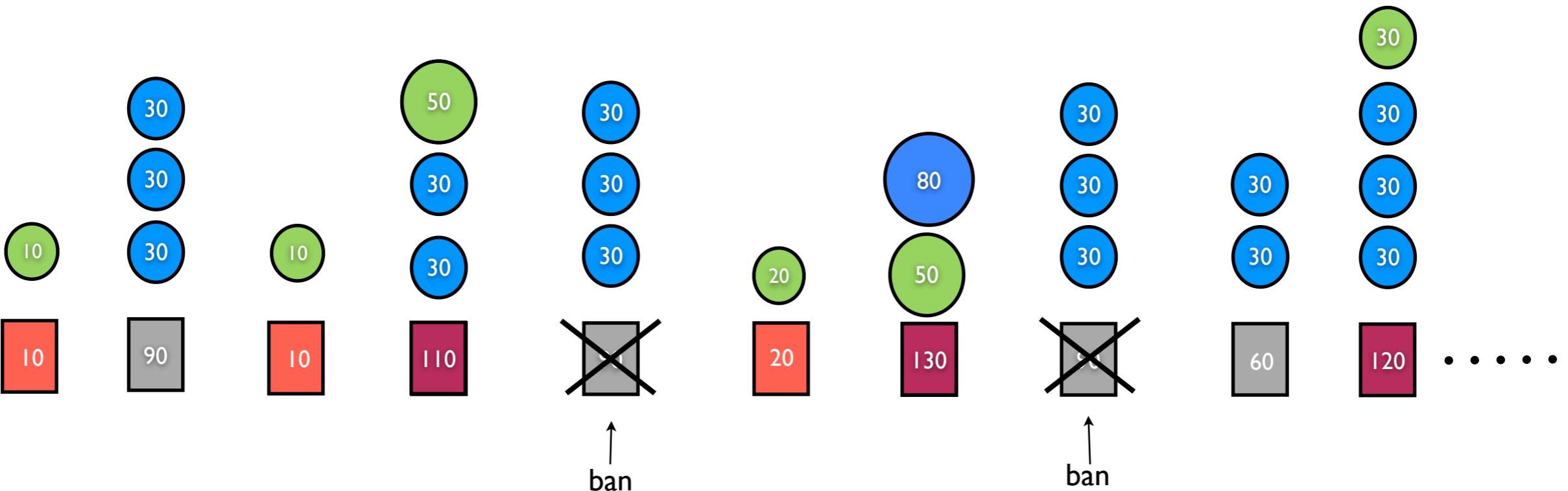
Example 3: shifted overloaded events



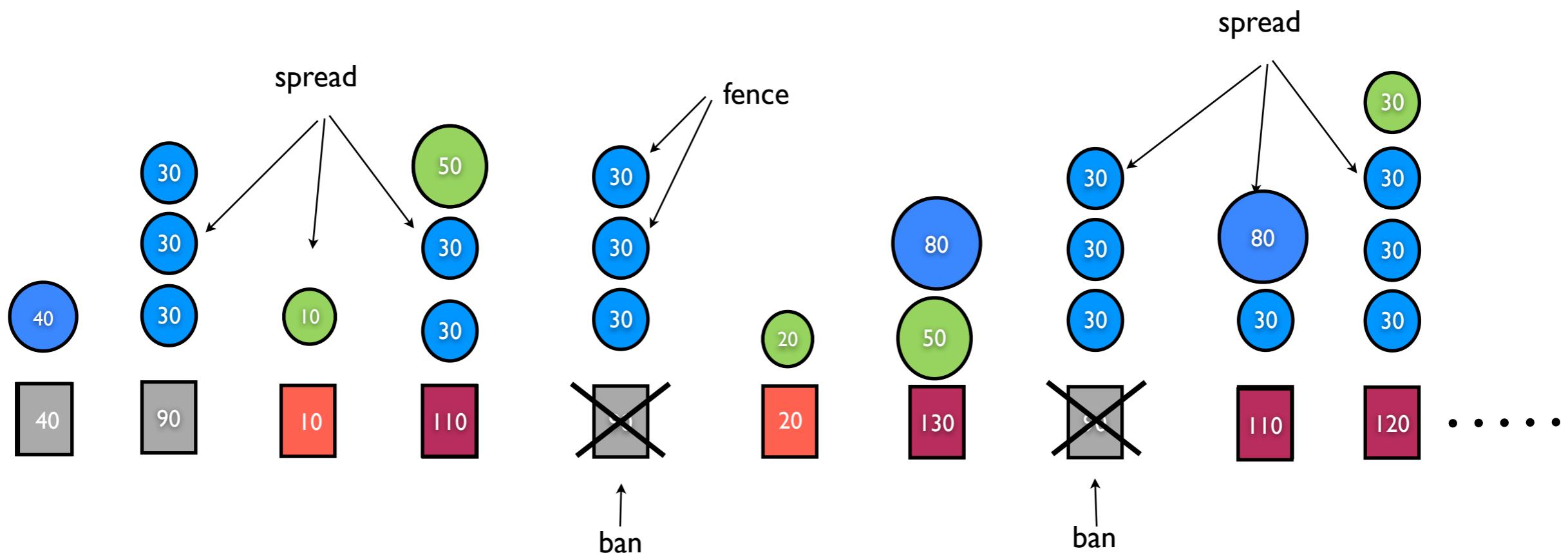
Example 4: your turn ?



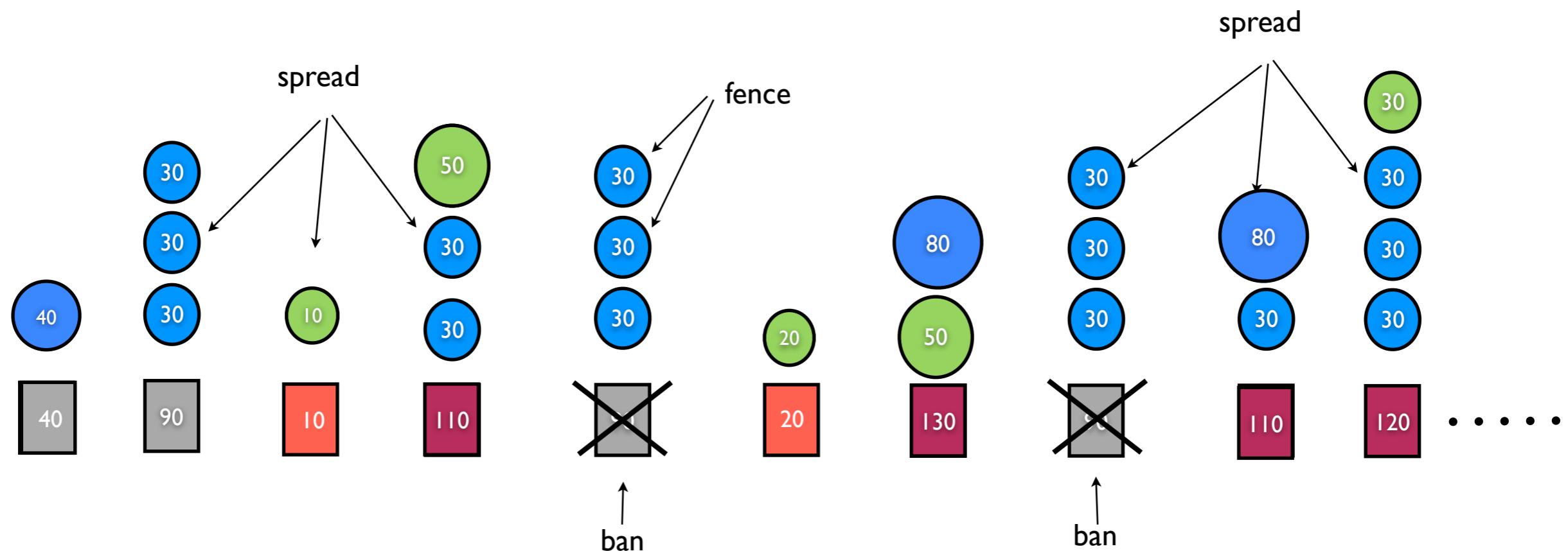
Example 4: your turn ?



Example 4: your turn ?



Example 4: your turn ?



Only CPU is considered in this simple example

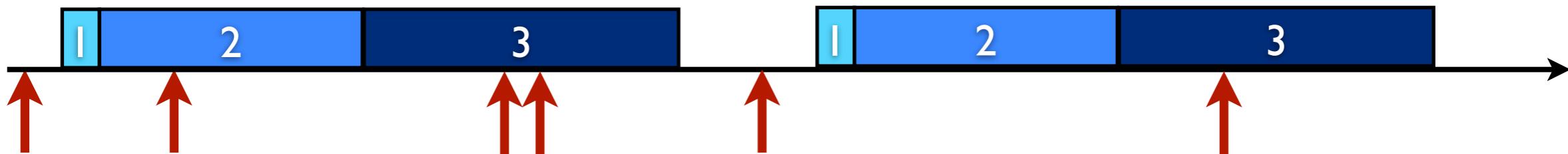


The LUC OS - VEs Scheduling

- POC (leveraging Entropy to solve non viable configuration)
- 100K VMs / 10K PMs (simulation using the SimGrid framework)
- Load injector

Events based on an Exponential law

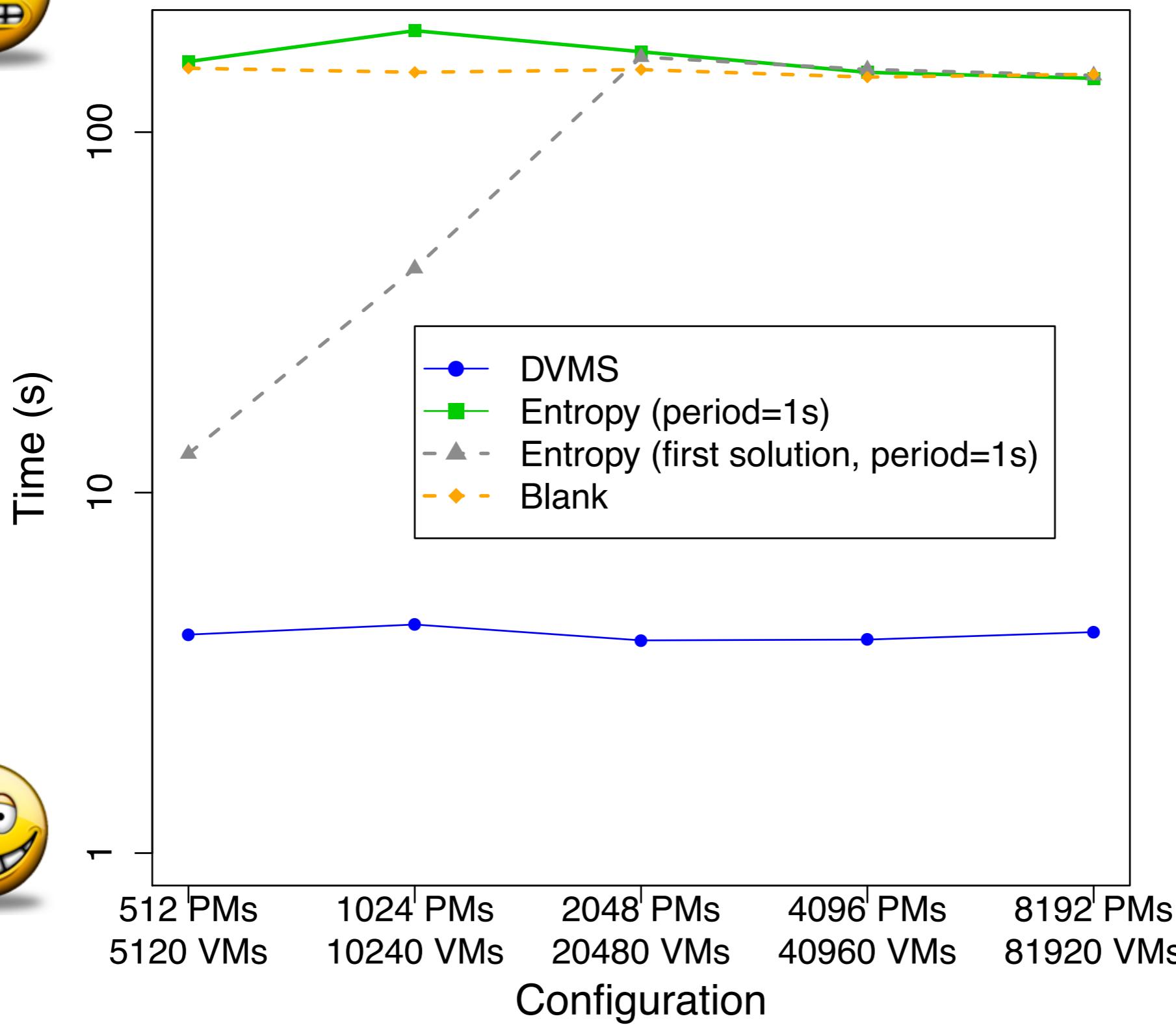
$$(\lambda = \text{Nb VMs} / 300 \text{ sec})$$



VM Load based on a Gaussian Law
($\mu = 70$ / $\sigma = 30$)

- Simulation confirmed through in vivo experiments (leveraging a JAVA POC and manipulating 10K VMs / 512 PMs on G5K)

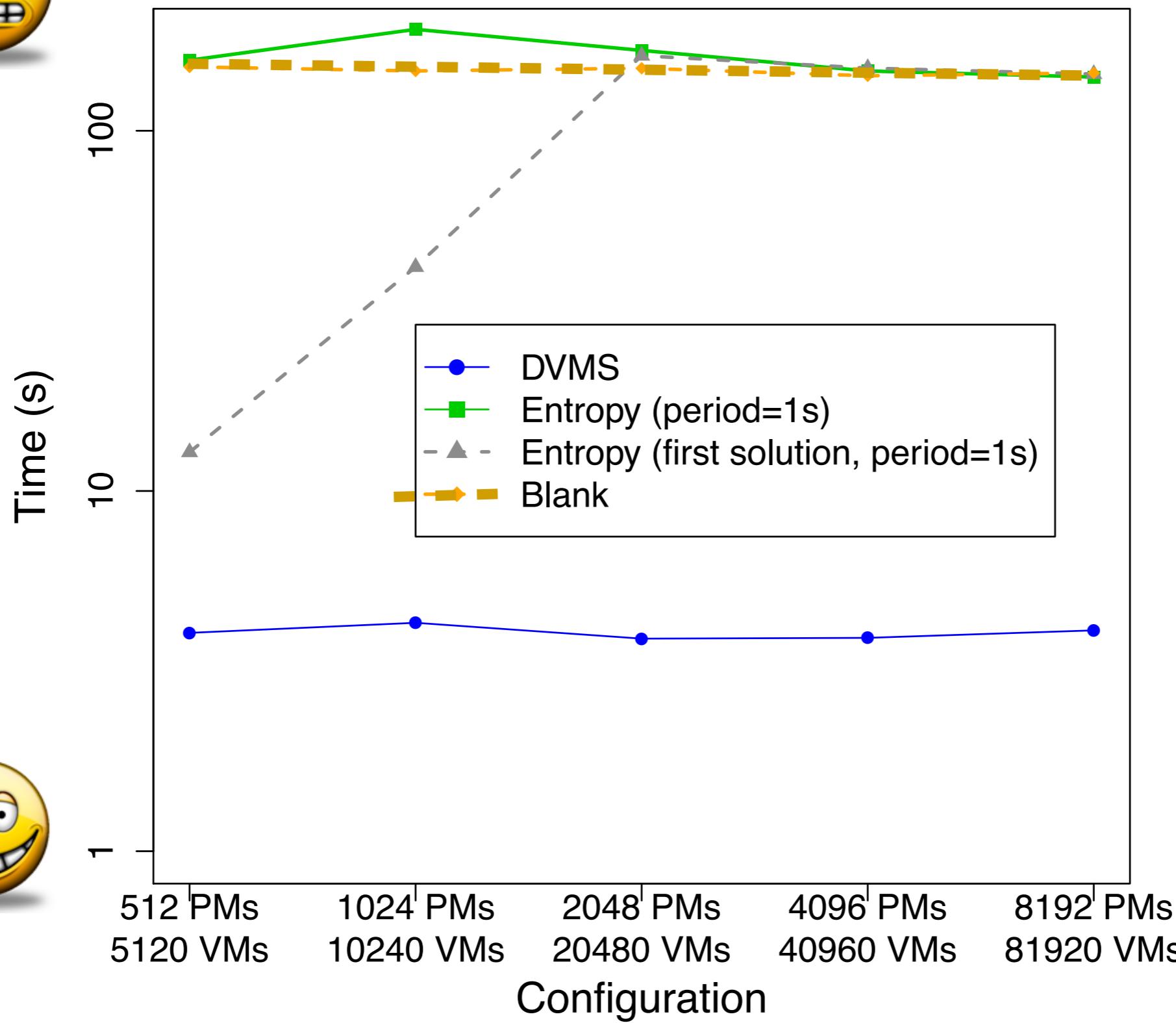
Violation Time Per Node



AVG Cluster Load
85 %
Simulation Time
3600 sec



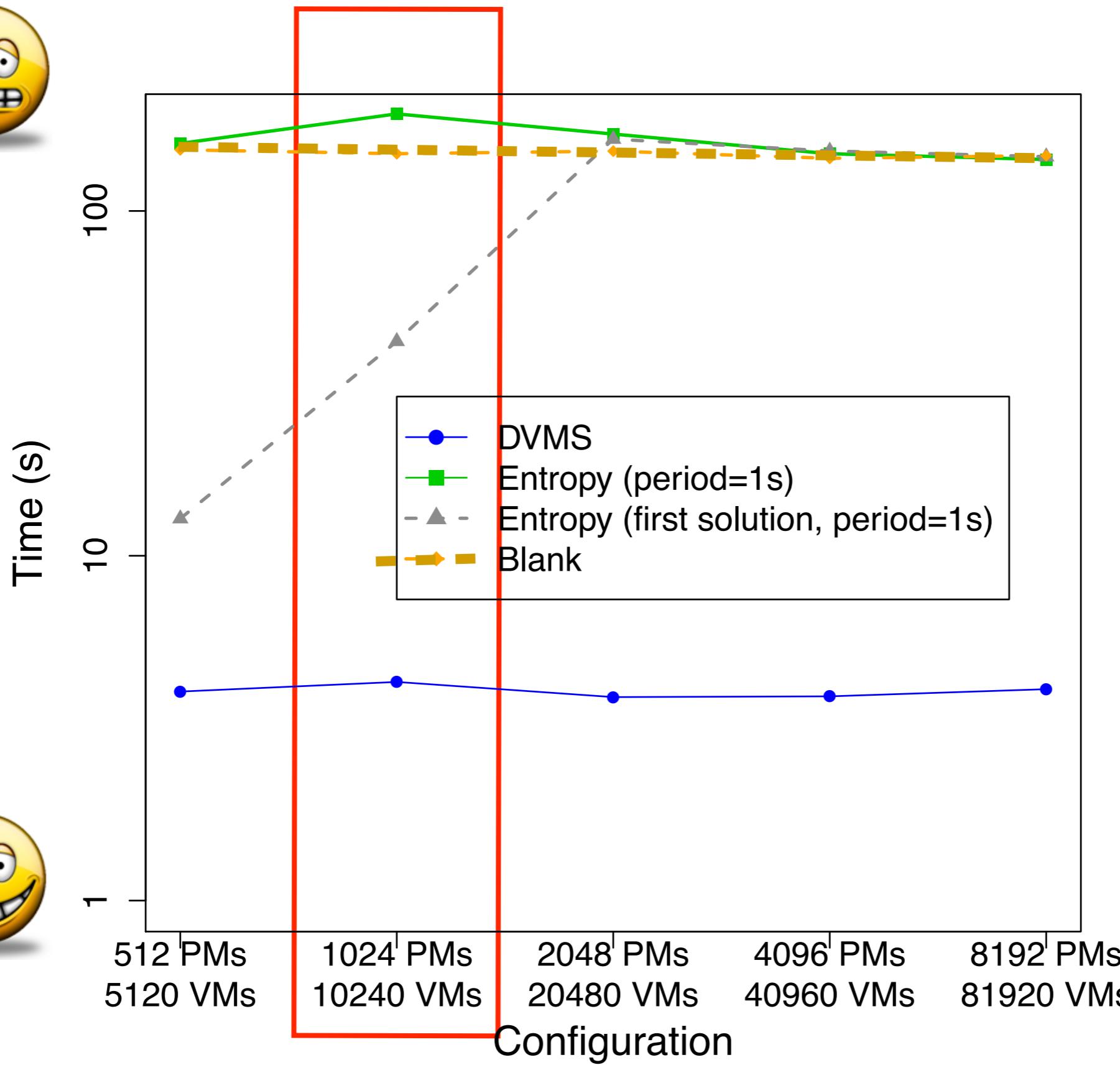
Violation Time Per Node



AVG Cluster Load
85 %
Simulation Time
3600 sec



Violation Time Per Node

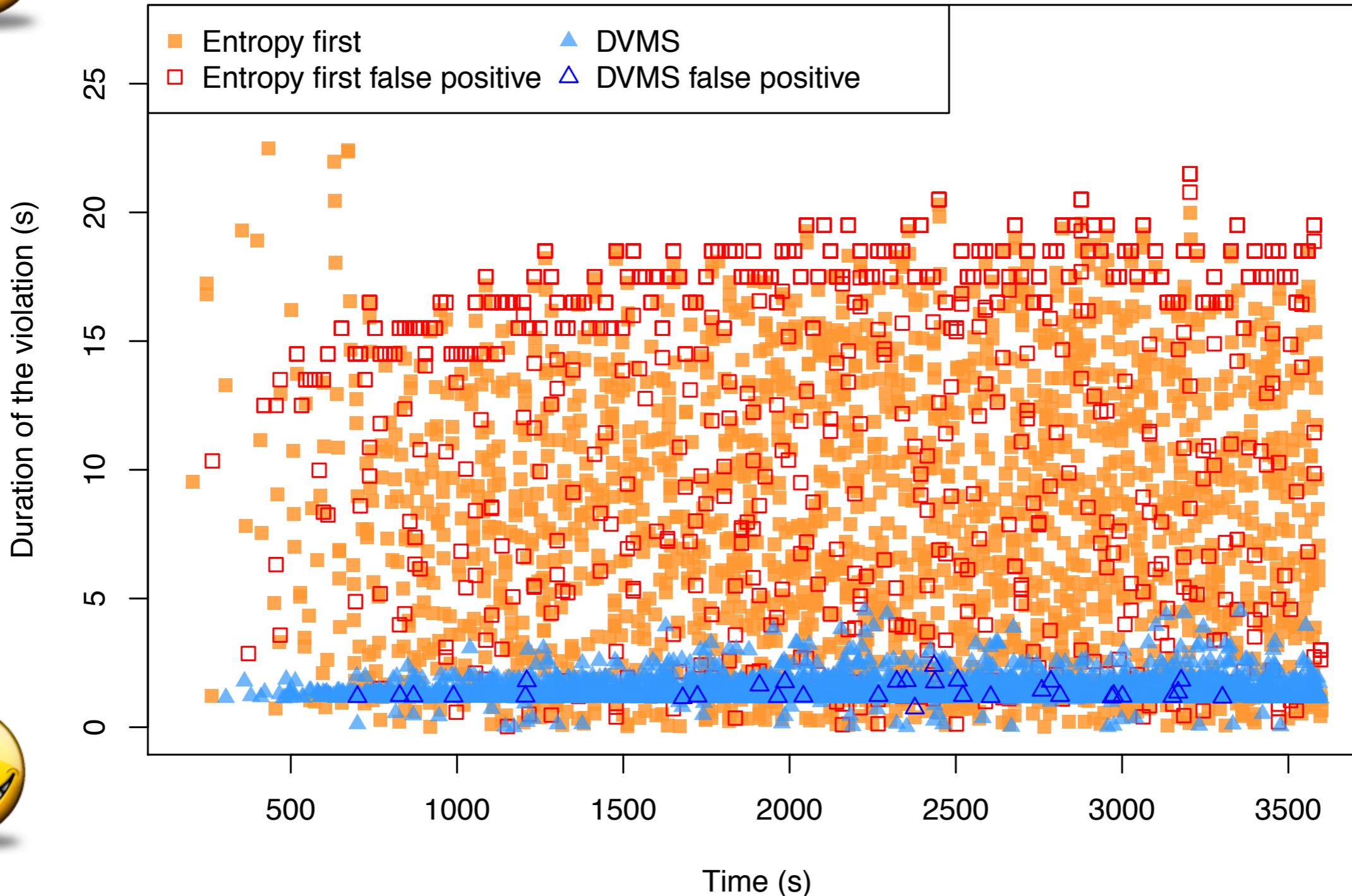


AVG Cluster Load
85 %
Simulation Time
3600 sec



SimGrid PJTDUMP - Devil is in details

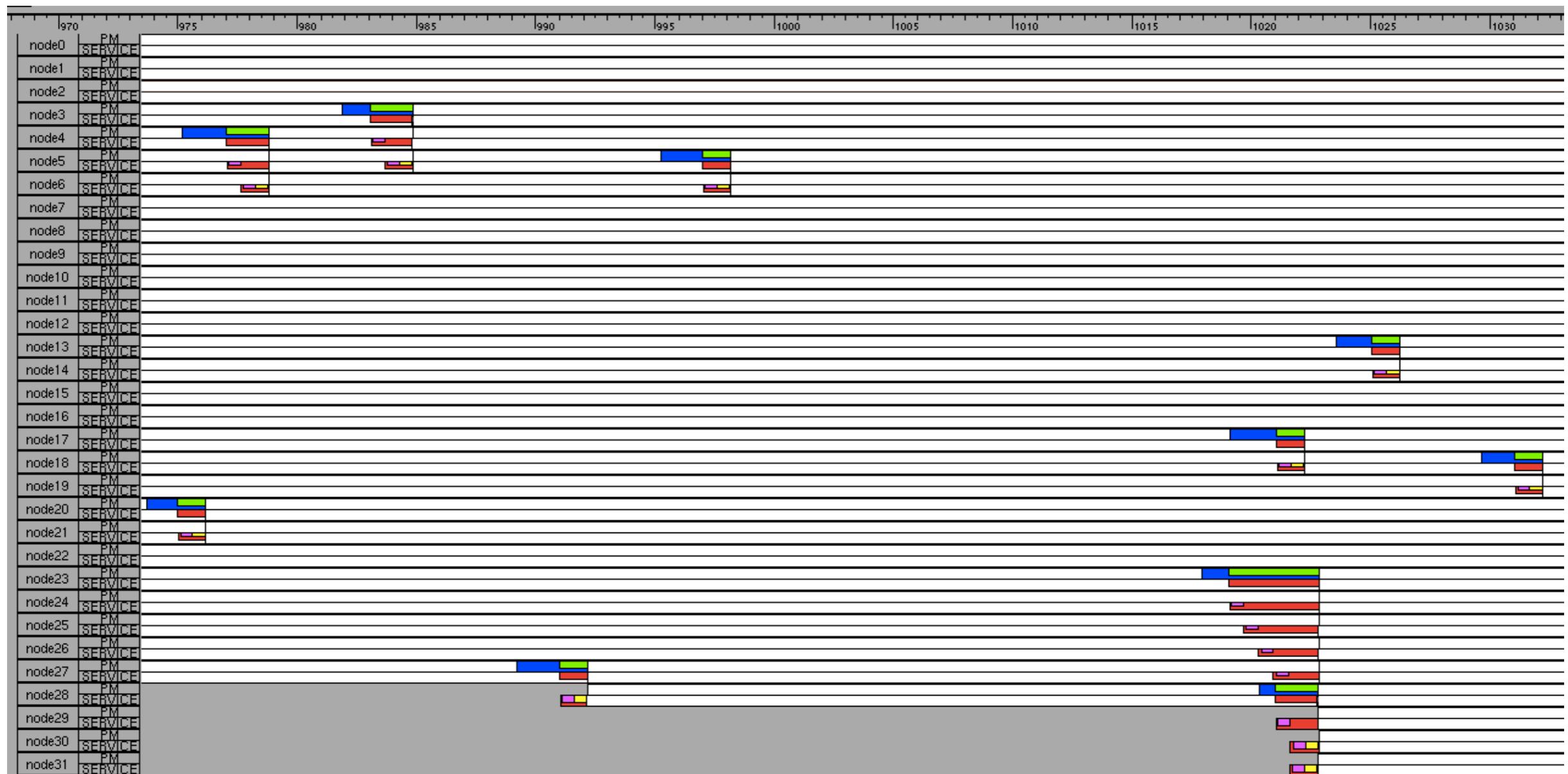
10240 VMs / 1024 Nodes



48/67

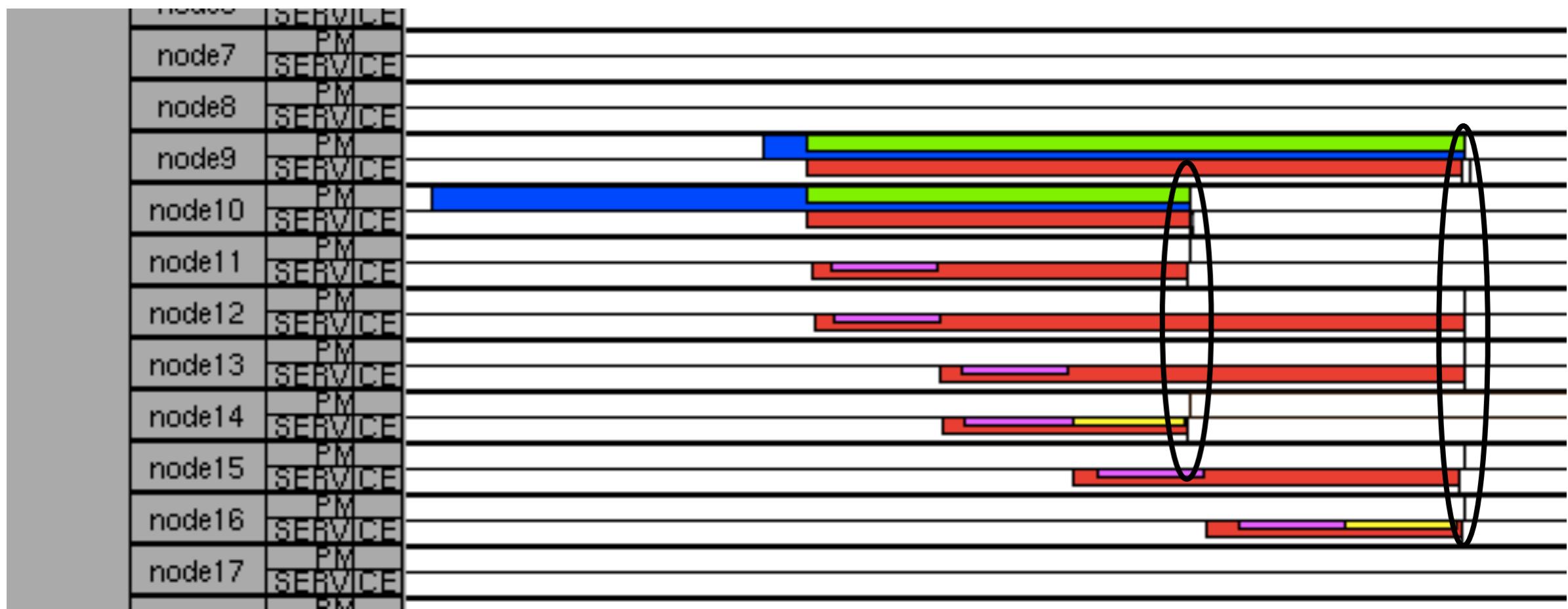
The LUC OS - VEs Scheduling

- Devil is in details
- Paje Traces (collecting during the SimGrid simulation)



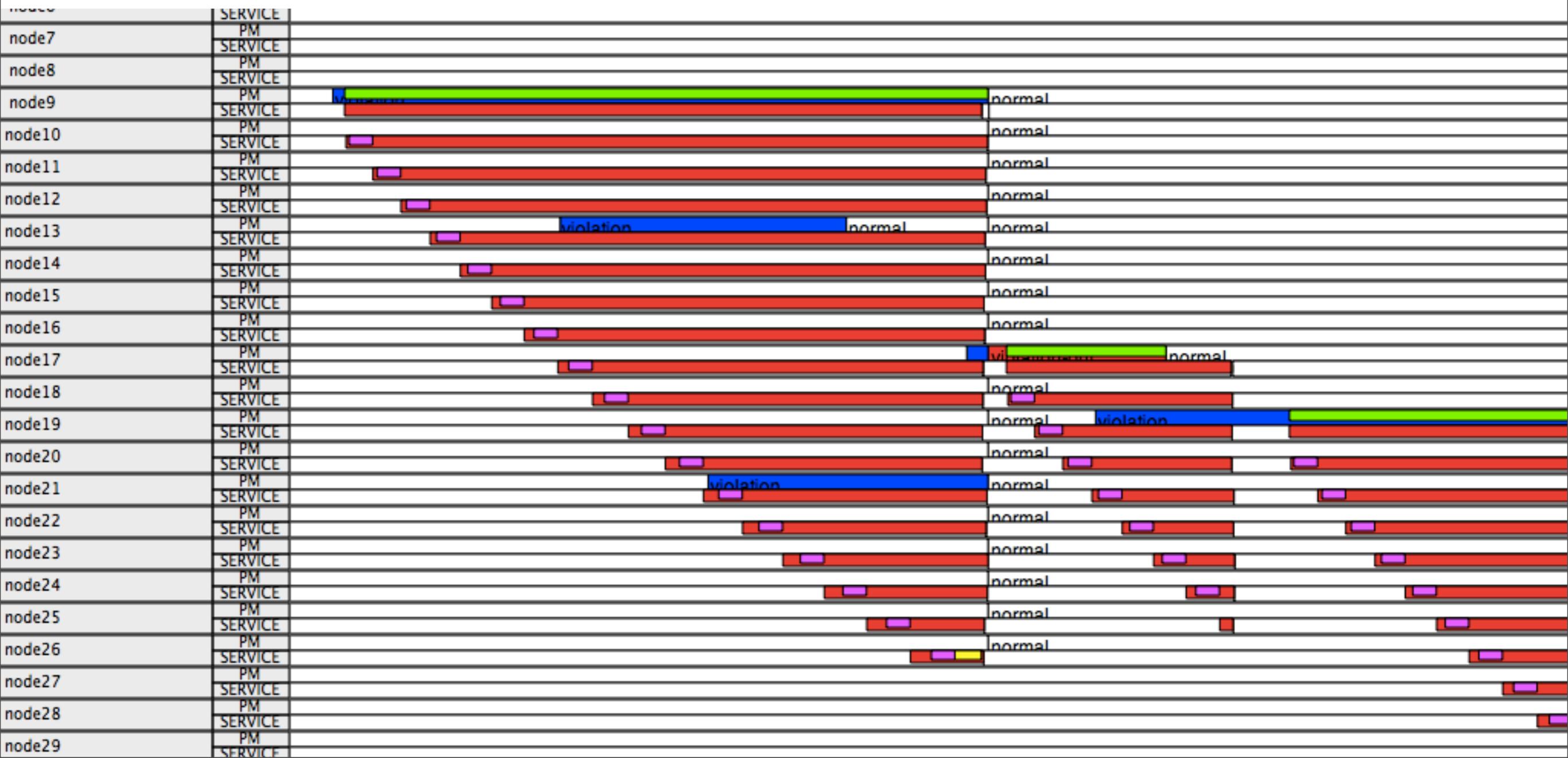
The LUC OS - VEs Scheduling

- Devil is in details
- Paje Traces (collecting during the SimGrid simulation)



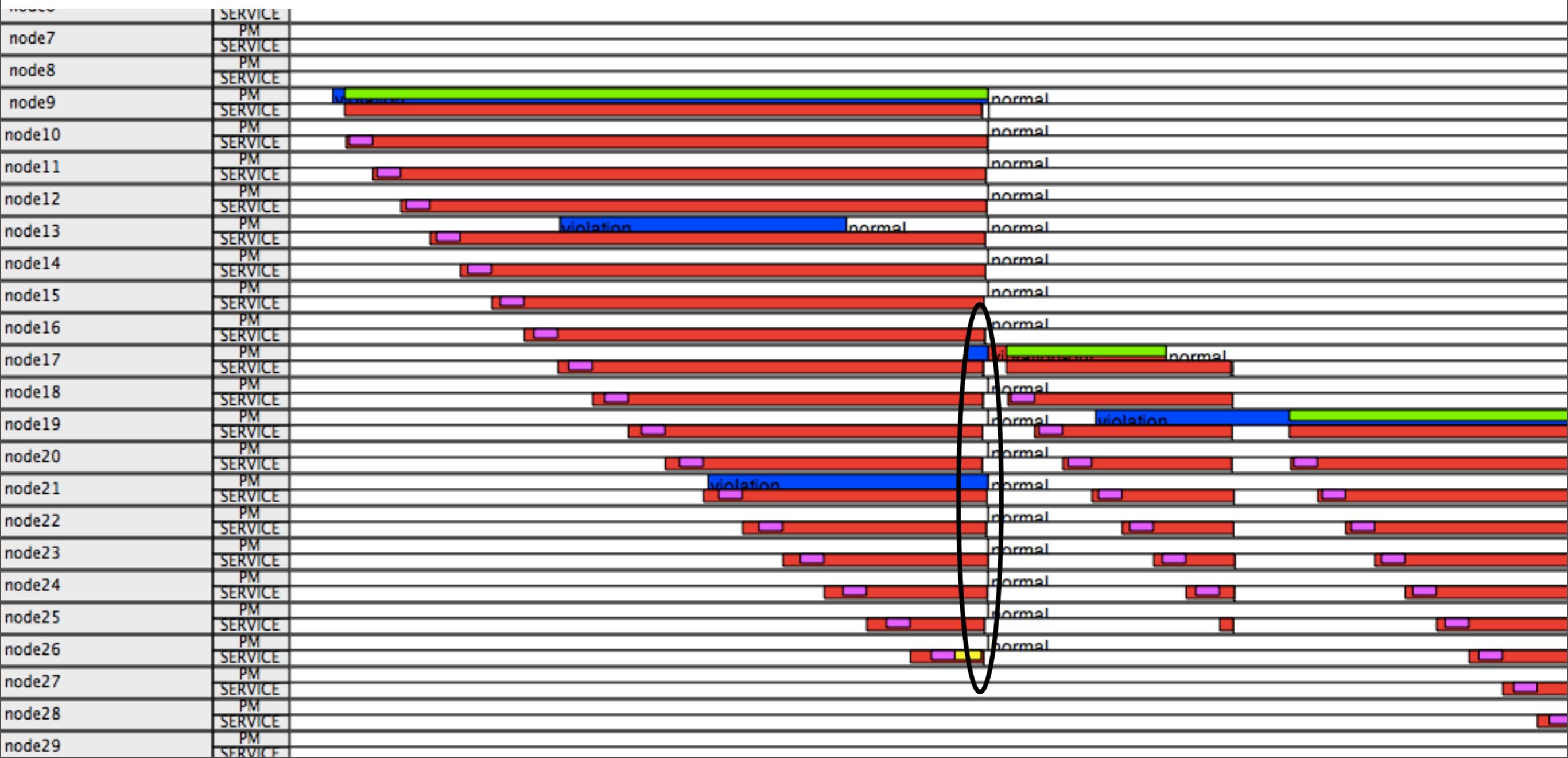
The LUC OS - VEs Scheduling

- Devil is in details
- Paje Traces (collecting during the SimGrid simulation)



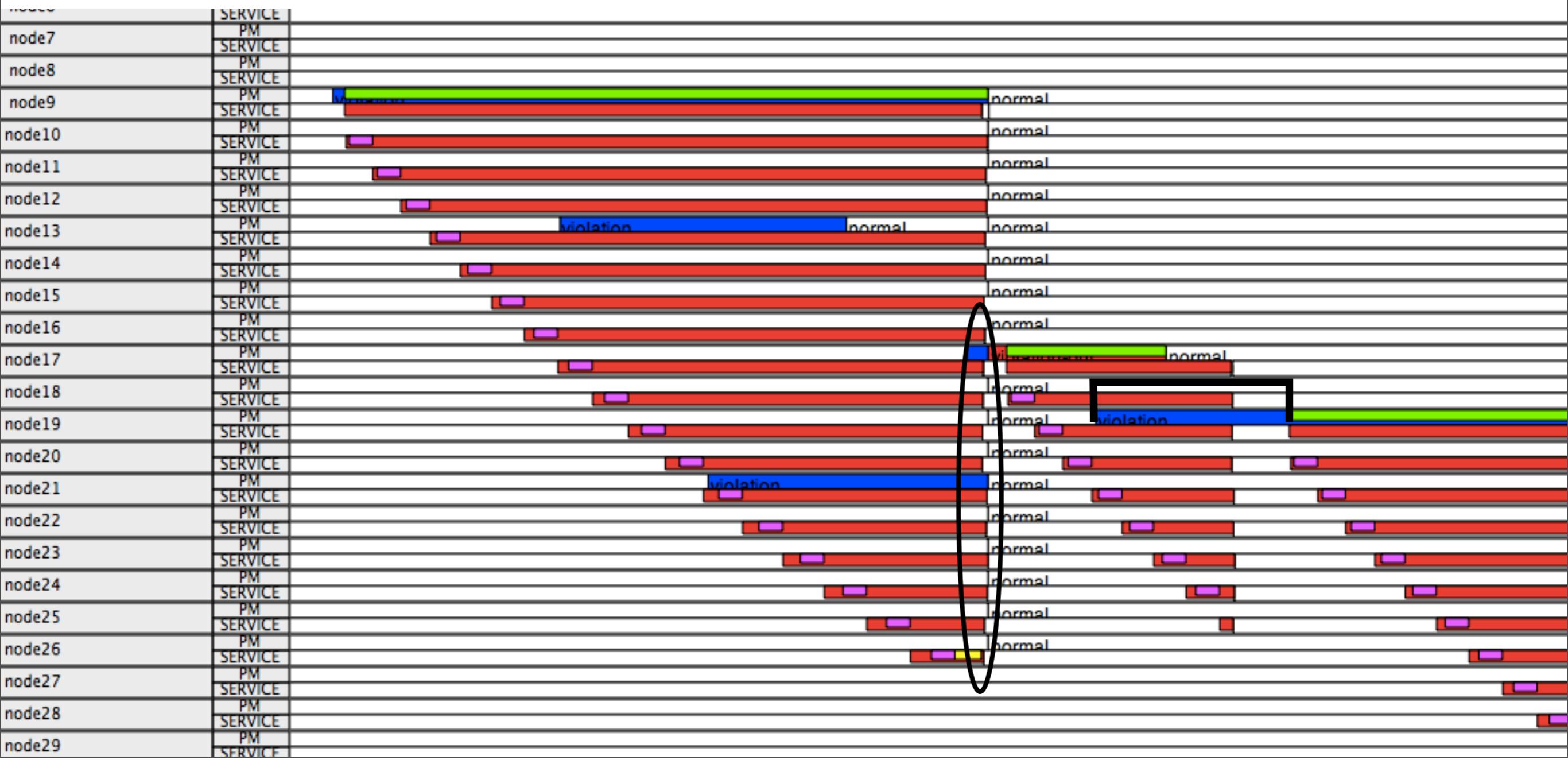
The LUC OS - VEs Scheduling

- Devil is in details
- Paje Traces (collecting during the SimGrid simulation)



The LUC OS - VEs Scheduling

- Devil is in details
- Paje Traces (collecting during the SimGrid simulation)



The LUC OS - DVMS as a first LRT

- DIStributed and COoperative approach to schedule VEs

Reactivity/scalability

Scheduling started when an event is generated

Few nodes considered for scheduling

⇒ much faster computation

Parallelism

Several events can be processed simultaneously/independently

Published in [CCPE'12, ISPA'13]

A POC in JAVA (however no more maintained)

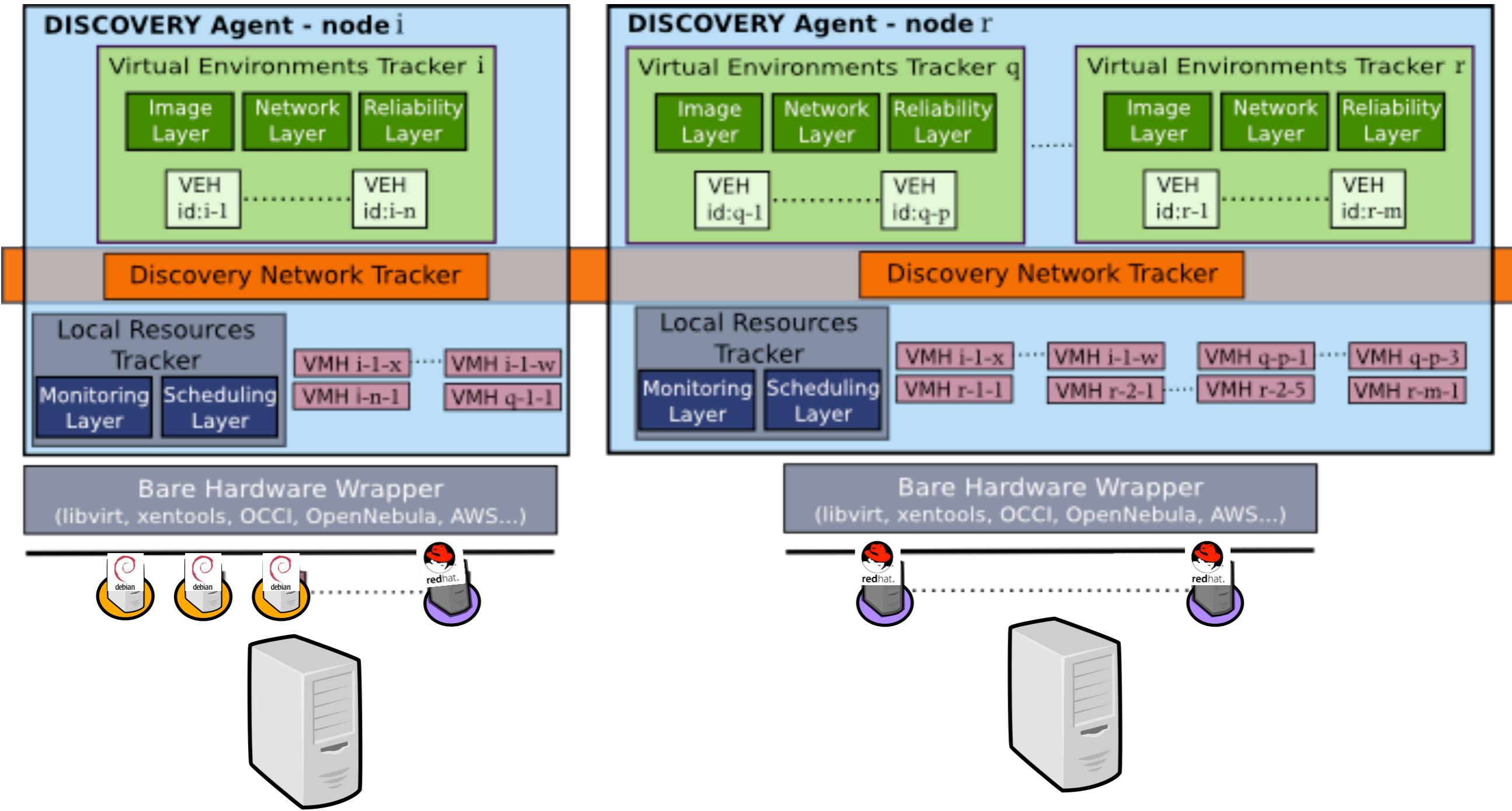
A new implementation based on SCALA/Akka

<http://beyondtheclouds.github.io/>

50/67

Discovery Internals in a Nutshell

Understanding the DISCOVERY Agent



DISCOVERY - Basic Usage



53/67

DISCOVERY - Basic Usage

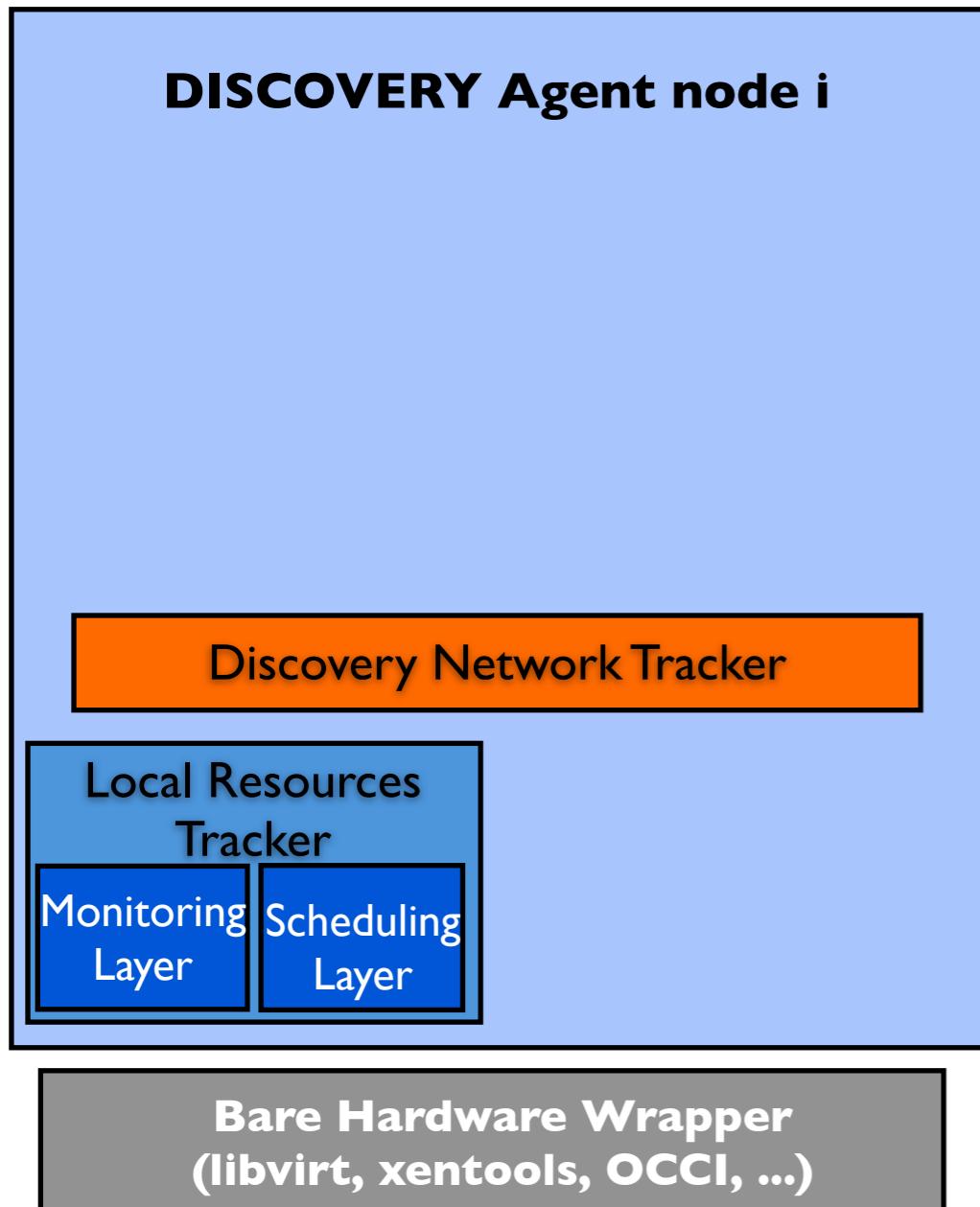
DISCOVERY Agent node i

Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



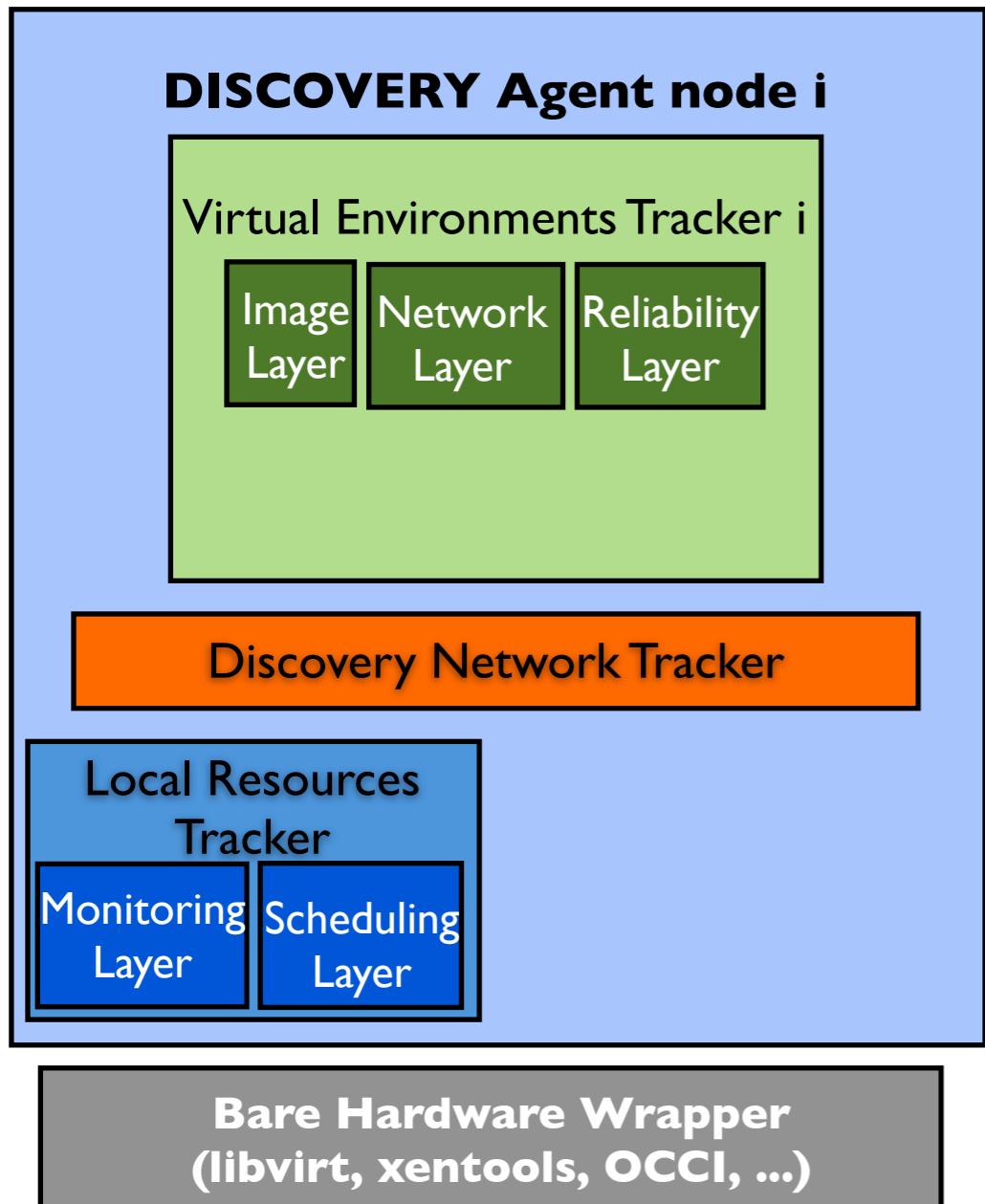
53/67

DISCOVERY - Basic Usage

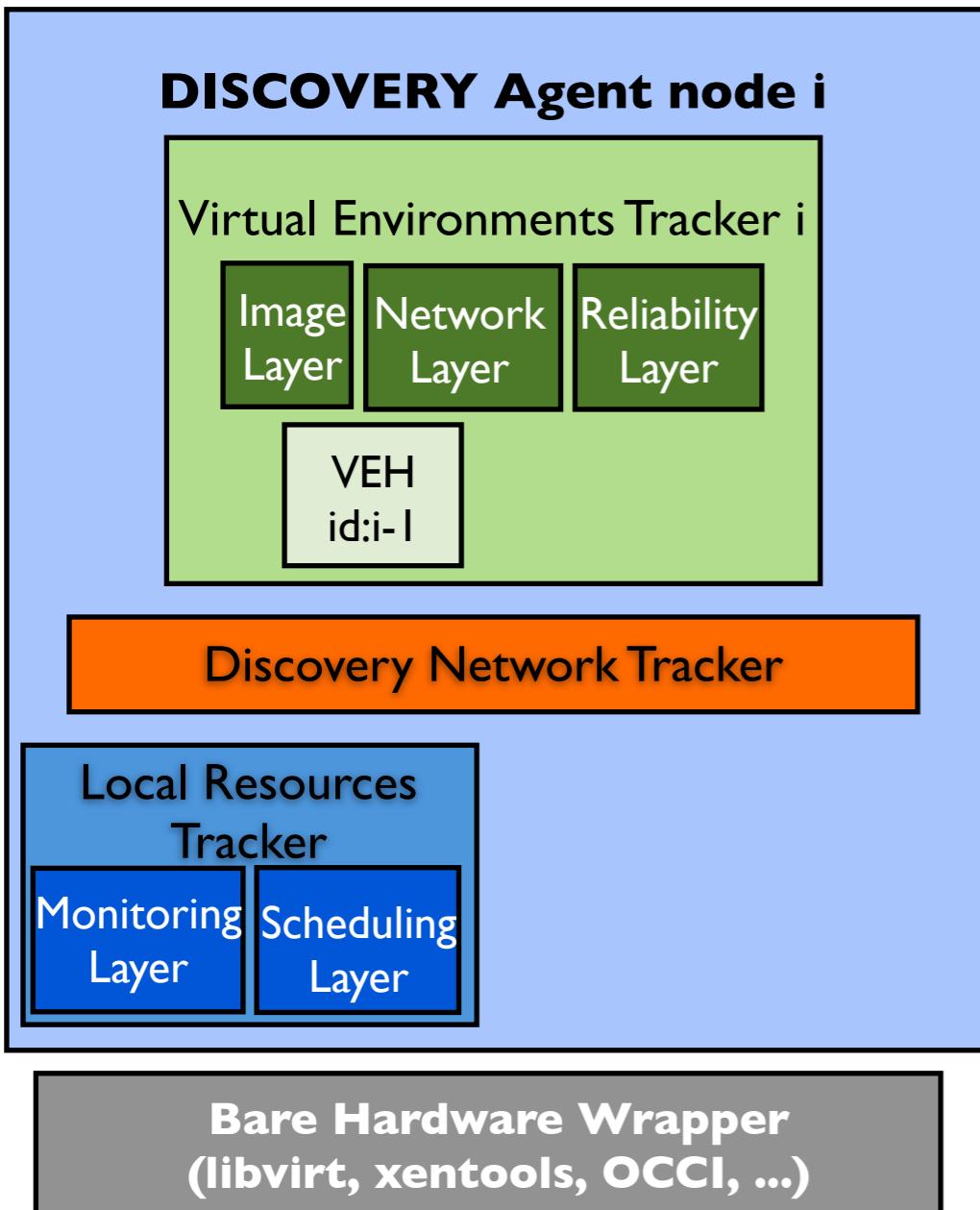


53/67

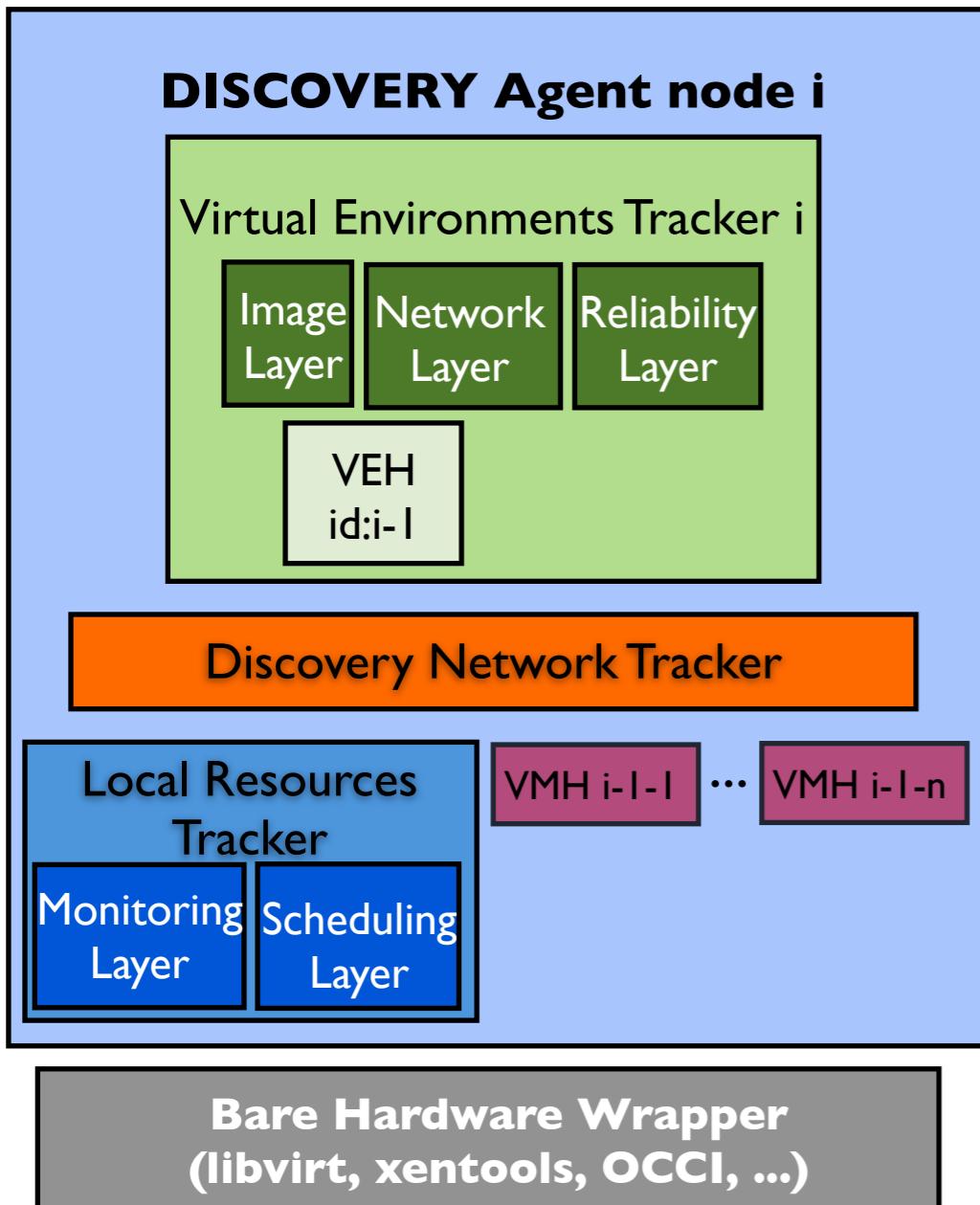
DISCOVERY - Basic Usage



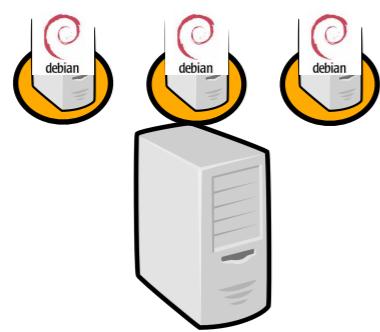
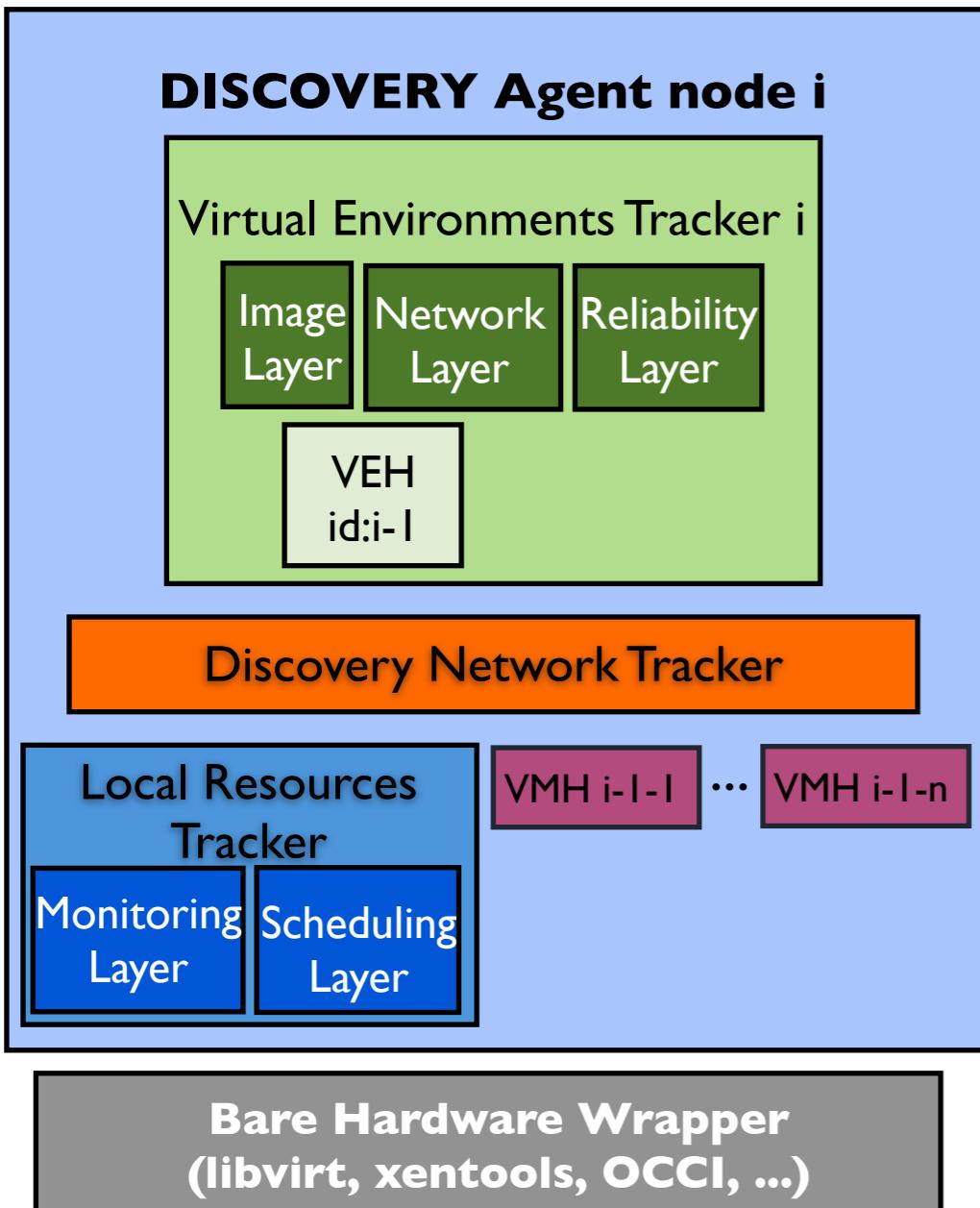
DISCOVERY - Basic Usage



DISCOVERY - Basic Usage

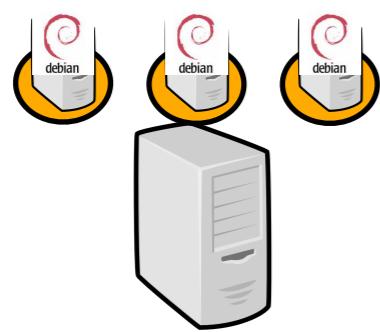
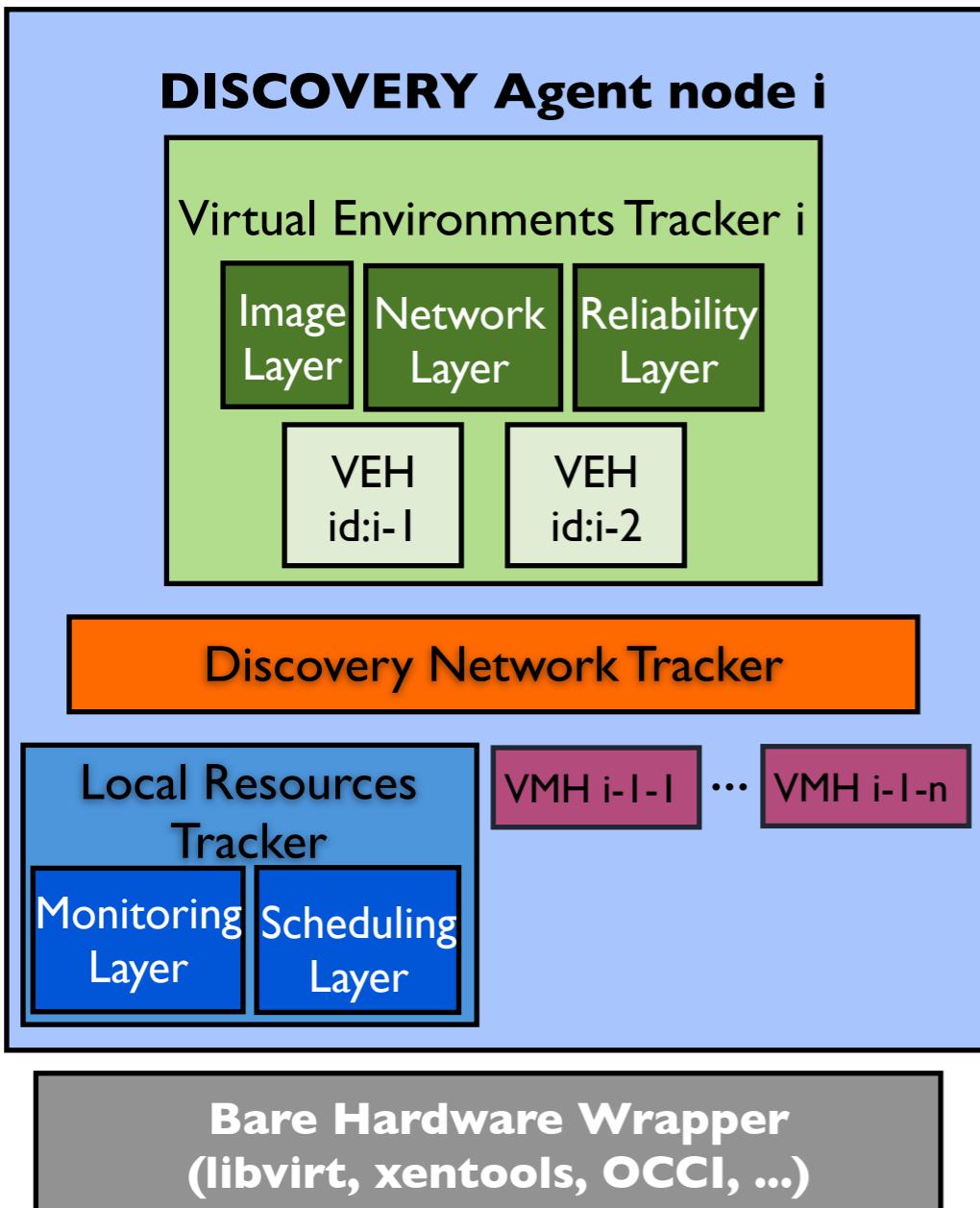


DISCOVERY - Basic Usage



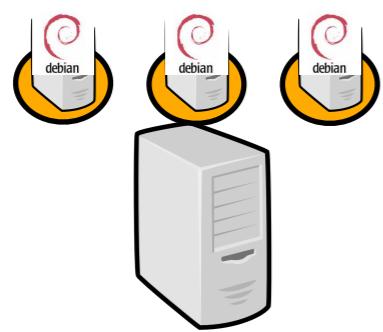
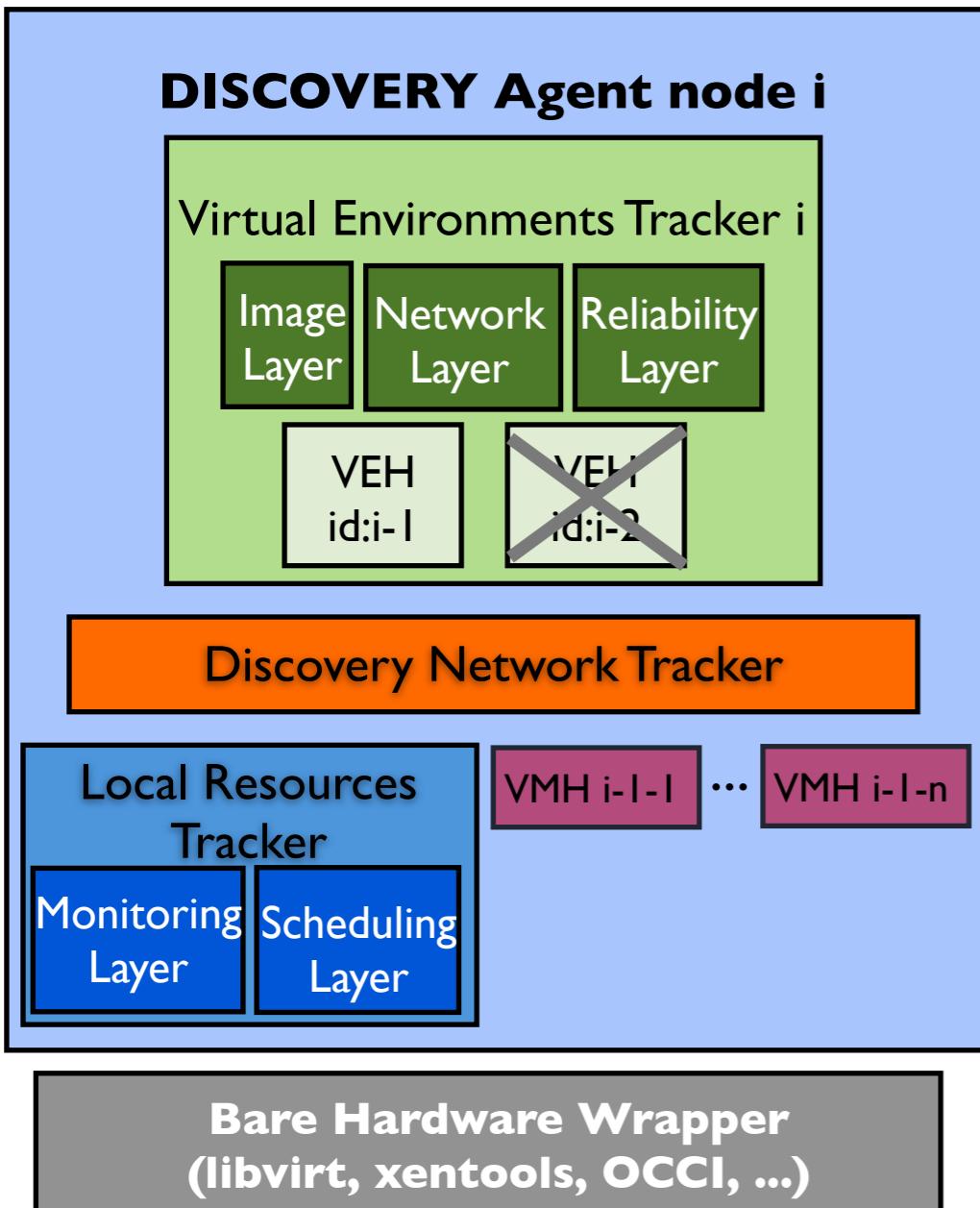
53/67

DISCOVERY - Basic Usage



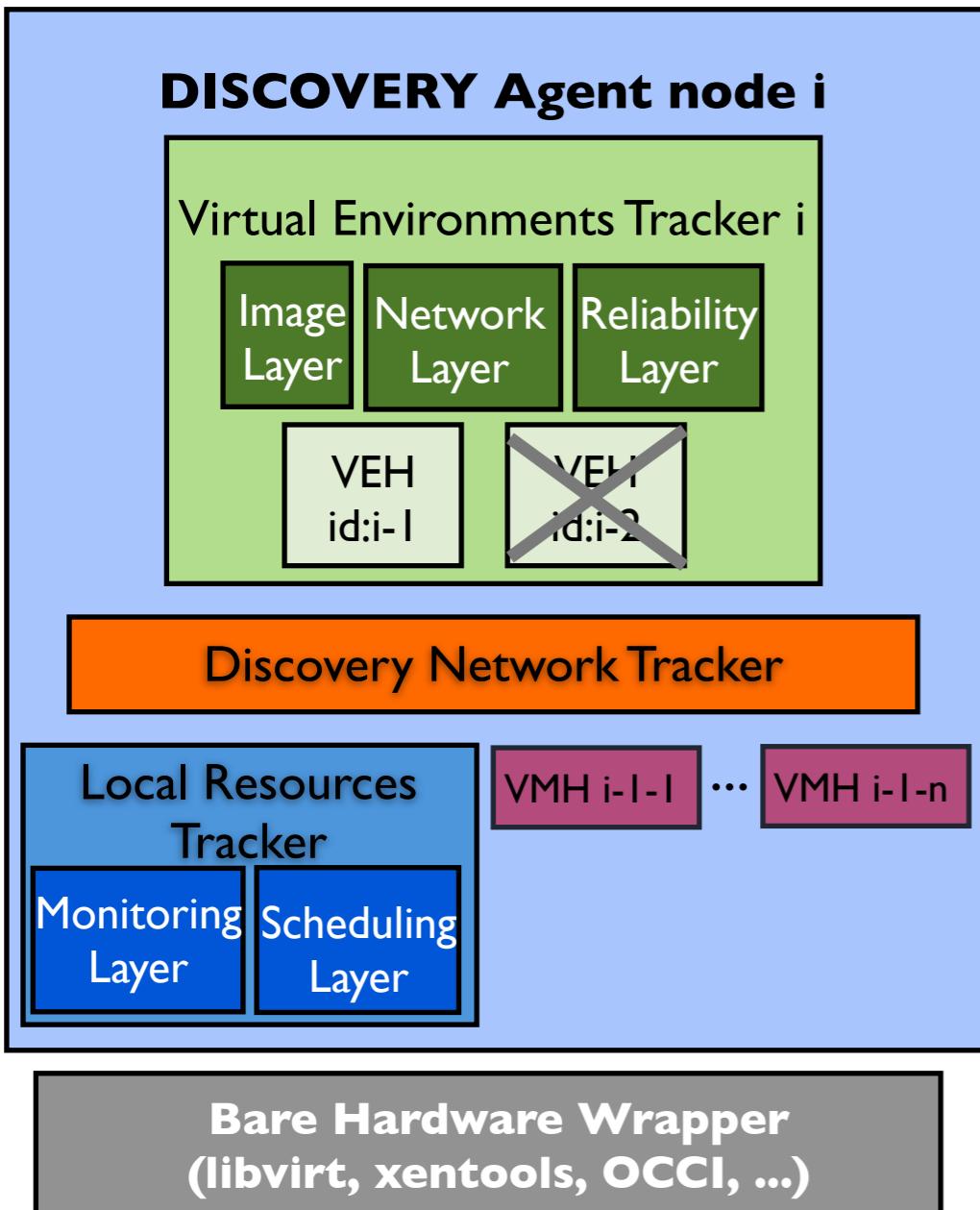
53/67

DISCOVERY - Basic Usage



53/67

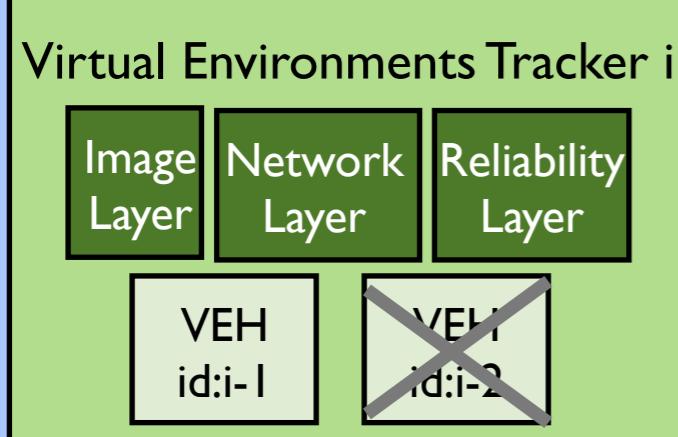
DISCOVERY - Basic Usage



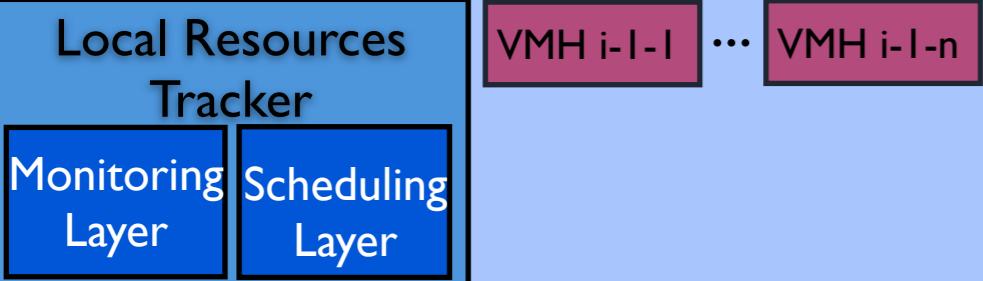
53/67

DISCOVERY - Basic Usage

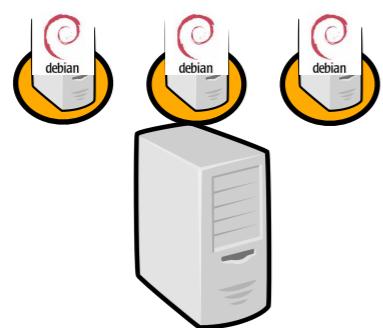
DISCOVERY Agent node i



Discovery Network Tracker

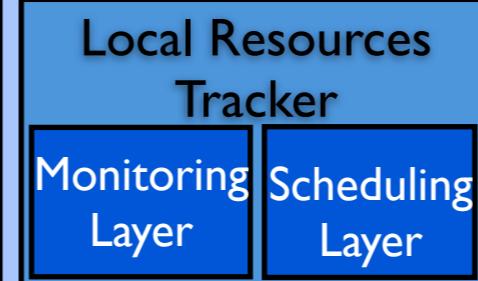


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r

Discovery Network Tracker

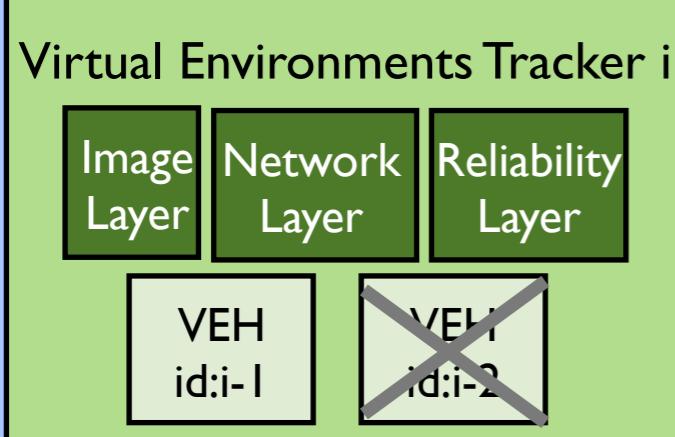


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

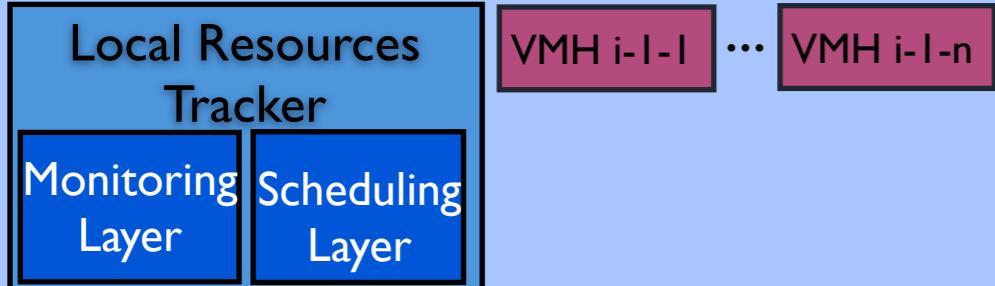


DISCOVERY - Basic Usage

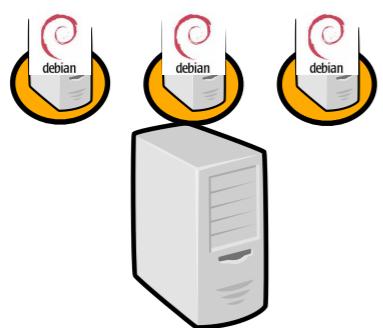
DISCOVERY Agent node i



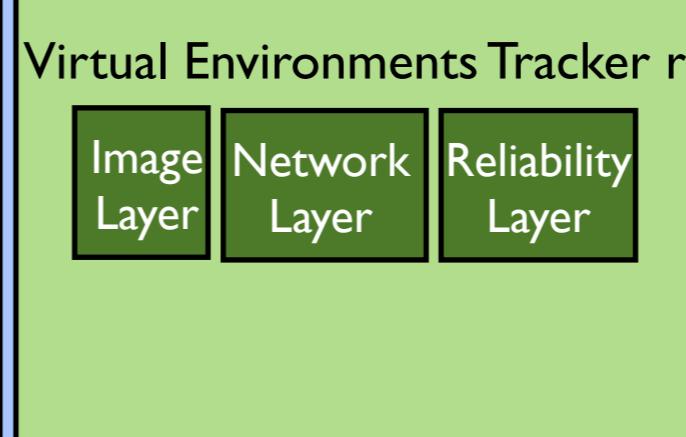
Discovery Network Tracker



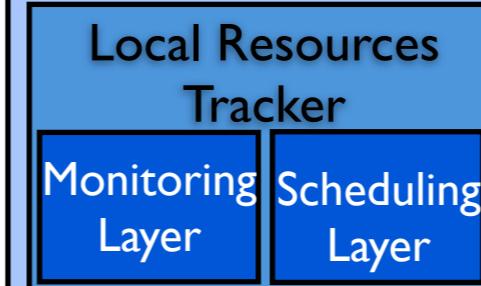
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r



Discovery Network Tracker

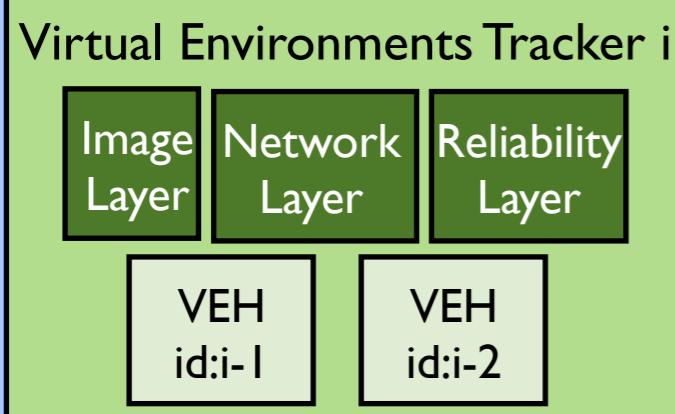


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

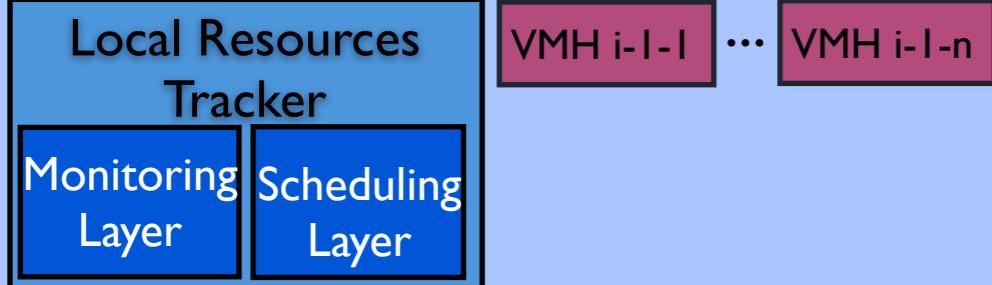


DISCOVERY - Basic Usage

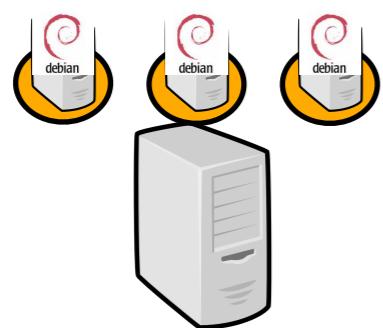
DISCOVERY Agent node i



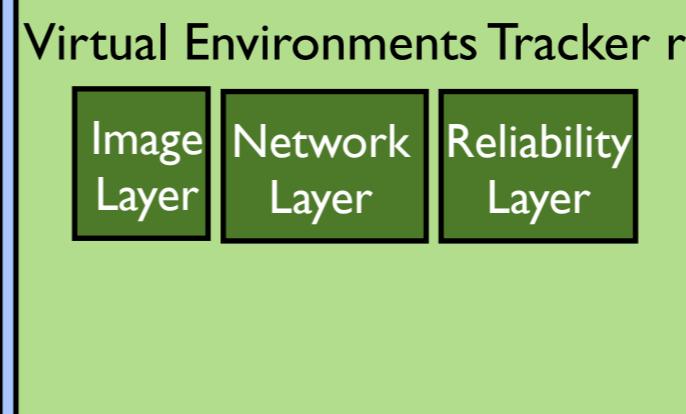
Discovery Network Tracker



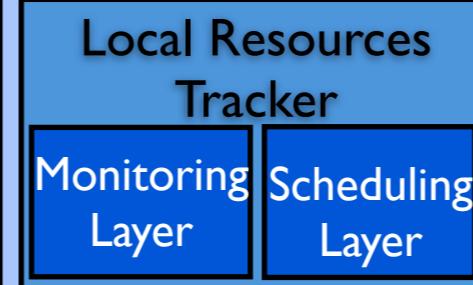
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r



Discovery Network Tracker

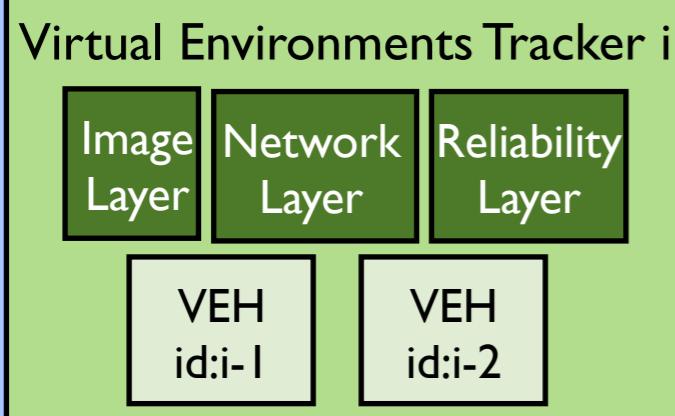


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

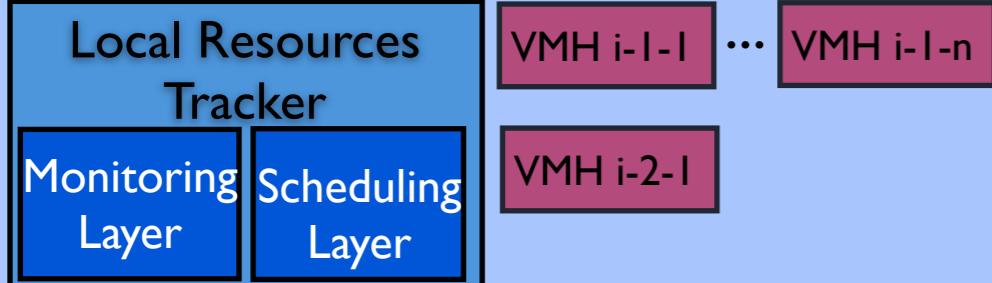


DISCOVERY - Basic Usage

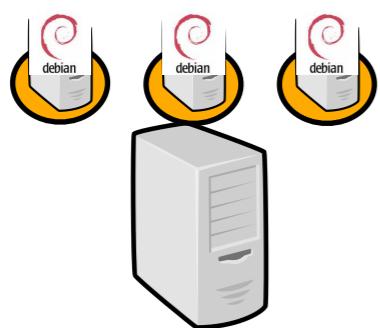
DISCOVERY Agent node i



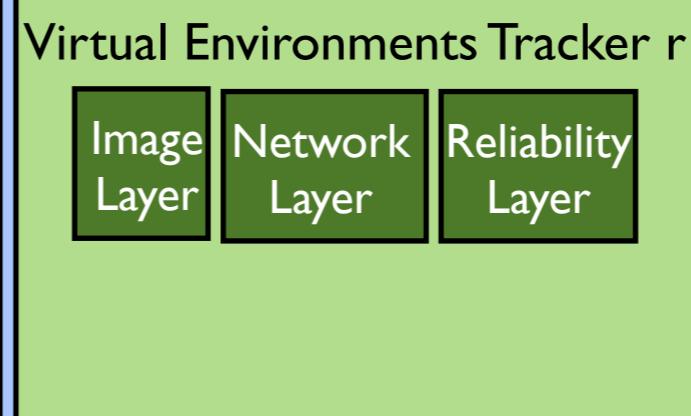
Discovery Network Tracker



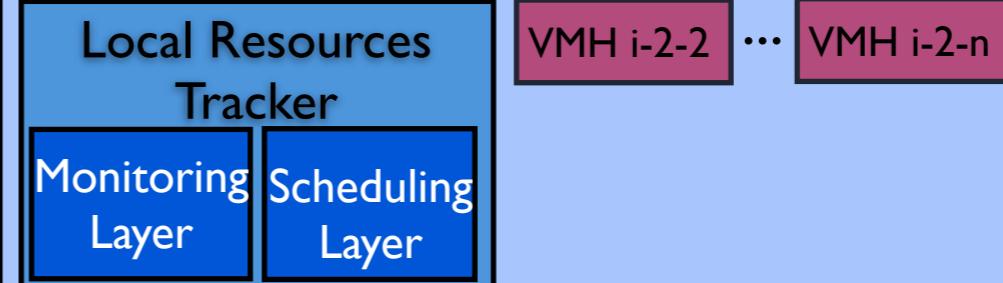
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r



Discovery Network Tracker

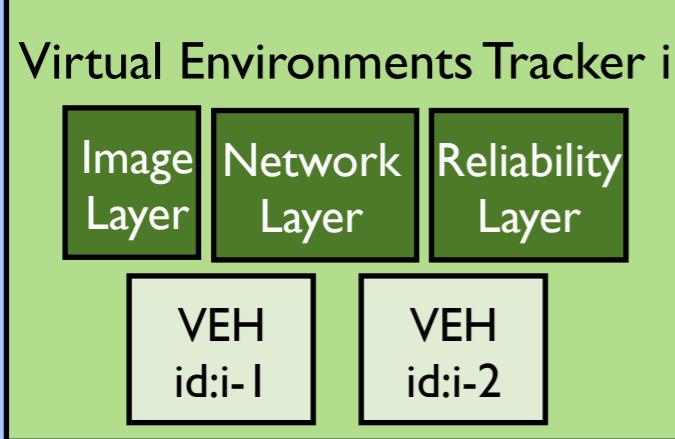


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

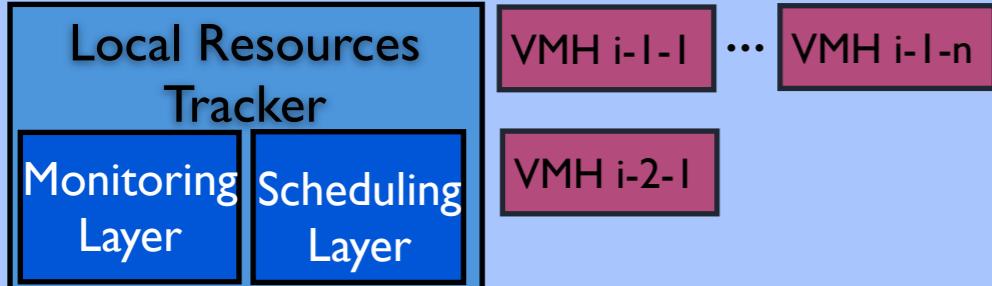


DISCOVERY - Basic Usage

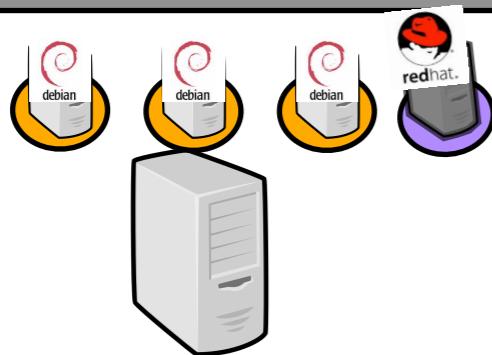
DISCOVERY Agent node i



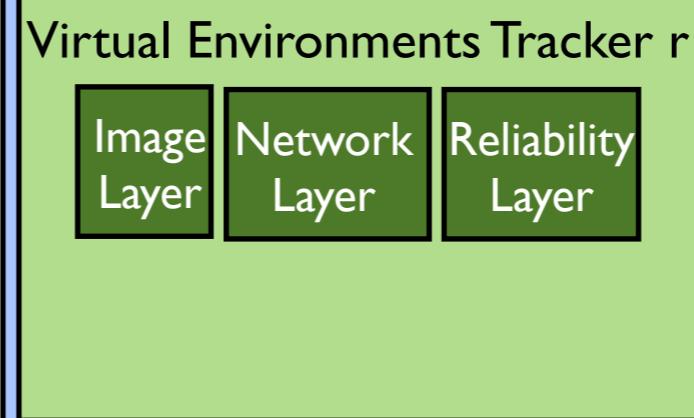
Discovery Network Tracker



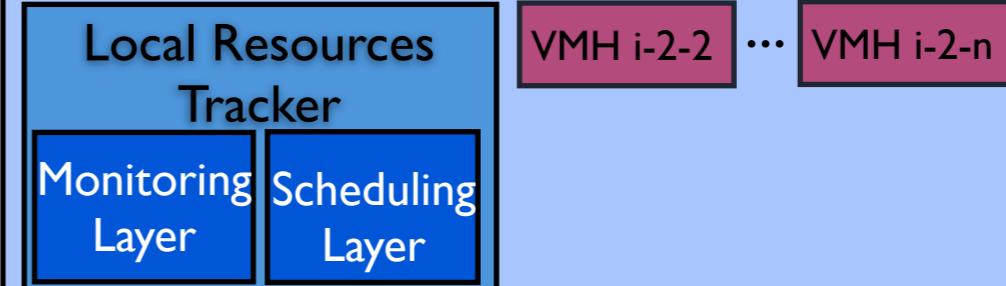
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



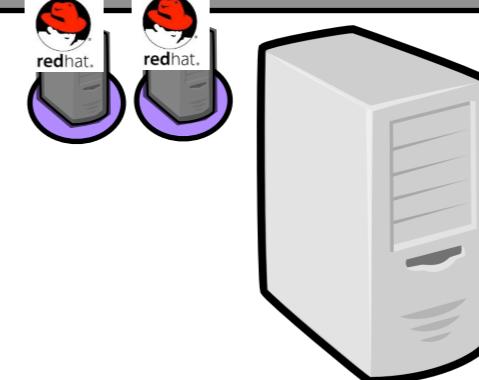
DISCOVERY Agent node r



Discovery Network Tracker

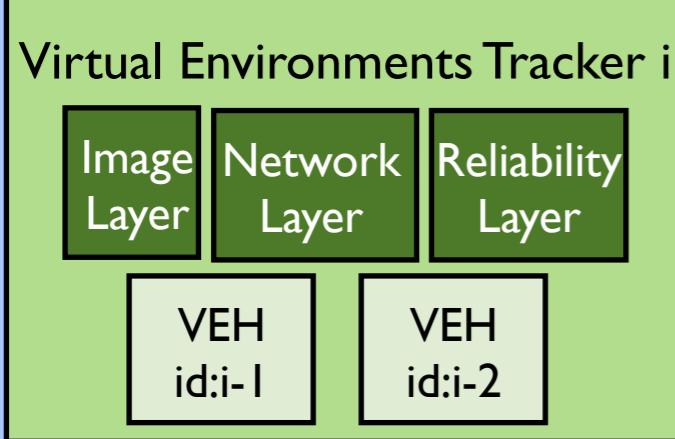


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

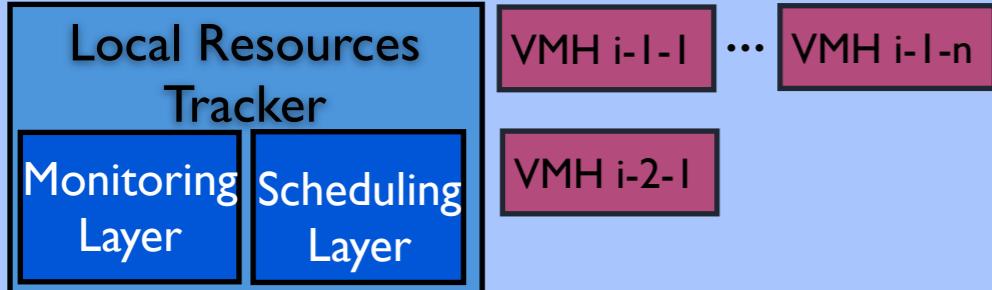


DISCOVERY - Basic Usage

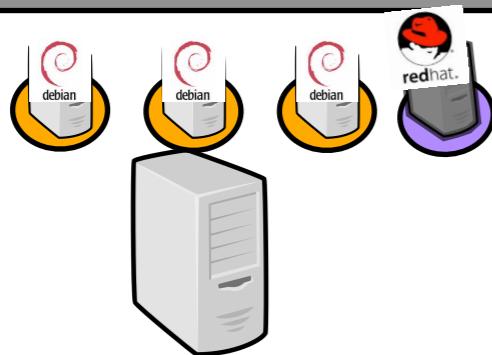
DISCOVERY Agent node i



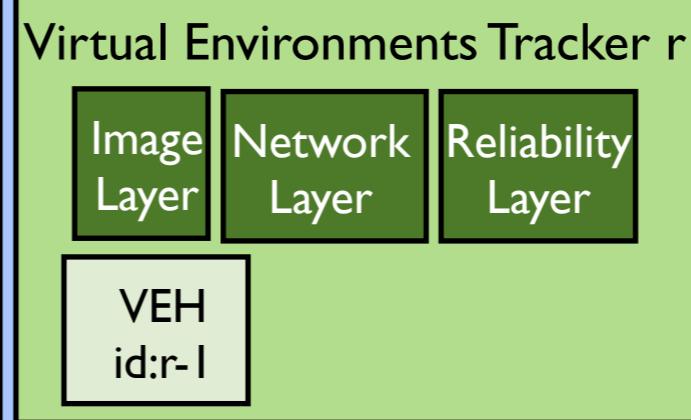
Discovery Network Tracker



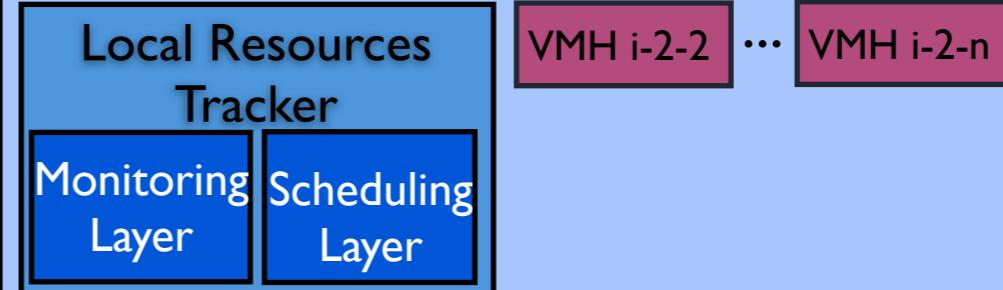
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



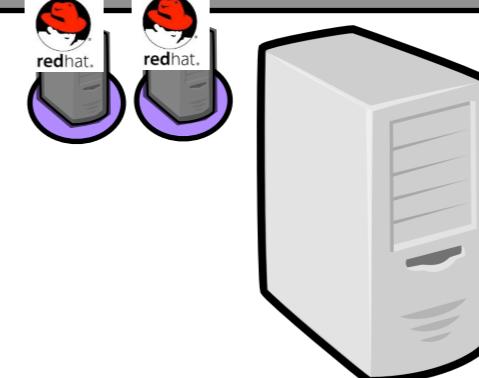
DISCOVERY Agent node r



Discovery Network Tracker

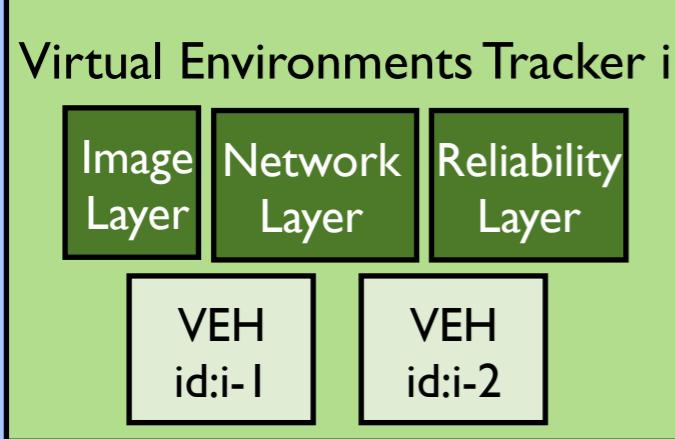


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

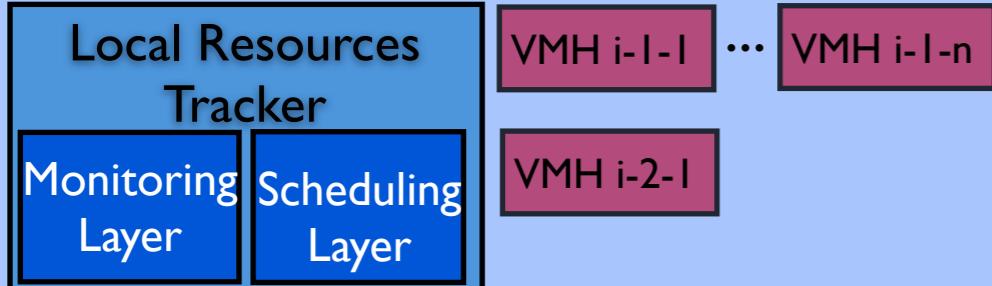


DISCOVERY - Basic Usage

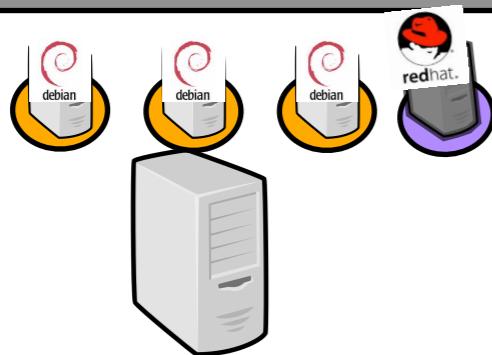
DISCOVERY Agent node i



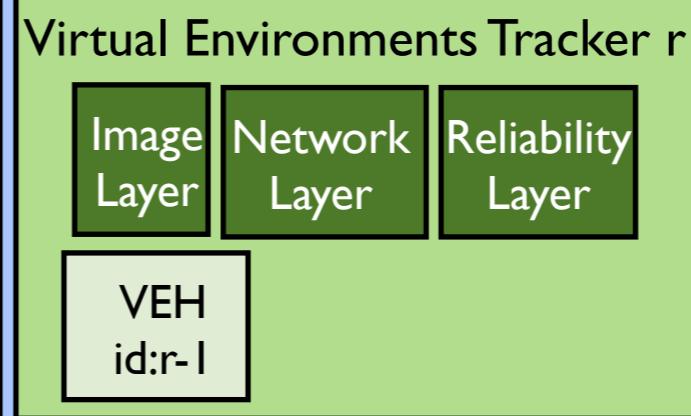
Discovery Network Tracker



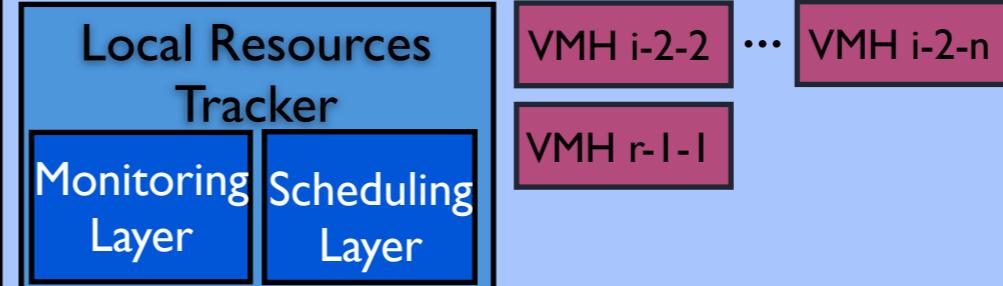
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



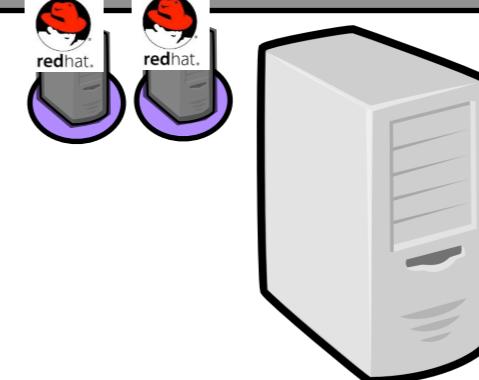
DISCOVERY Agent node r



Discovery Network Tracker

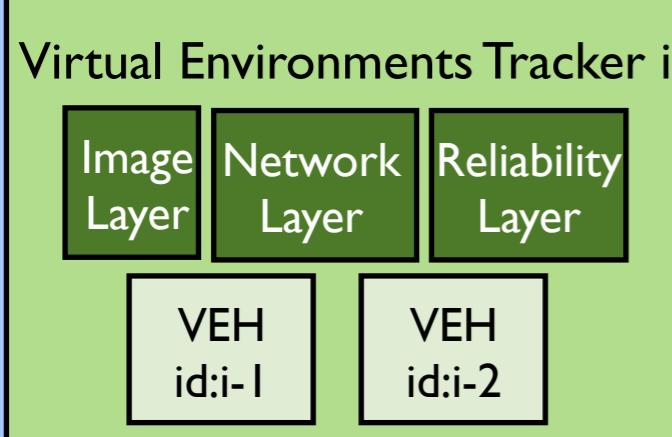


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

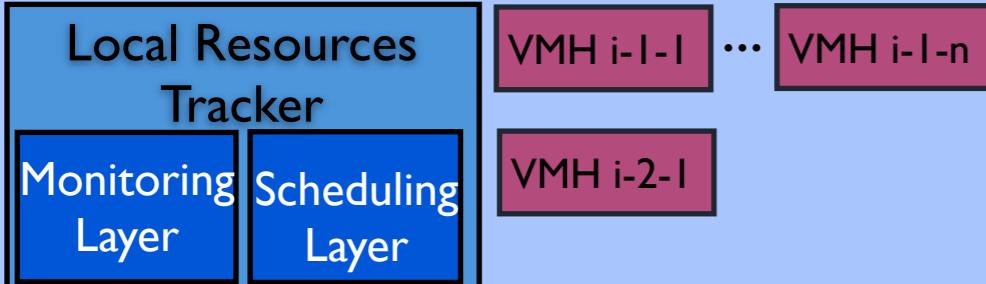


DISCOVERY - Basic Usage

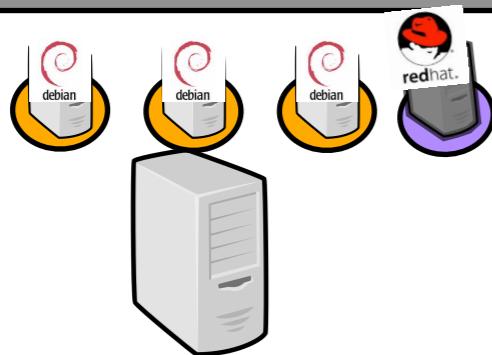
DISCOVERY Agent node i



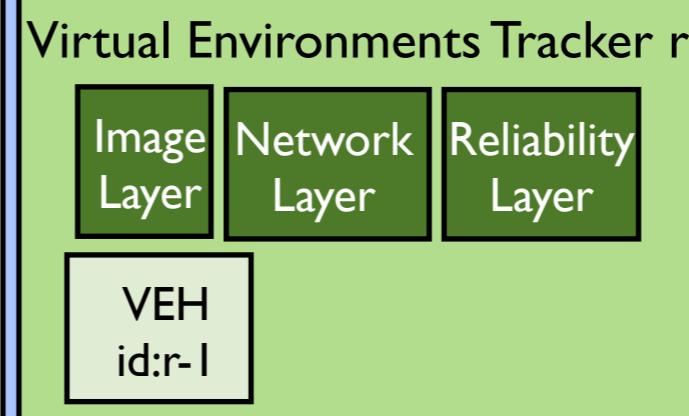
Discovery Network Tracker



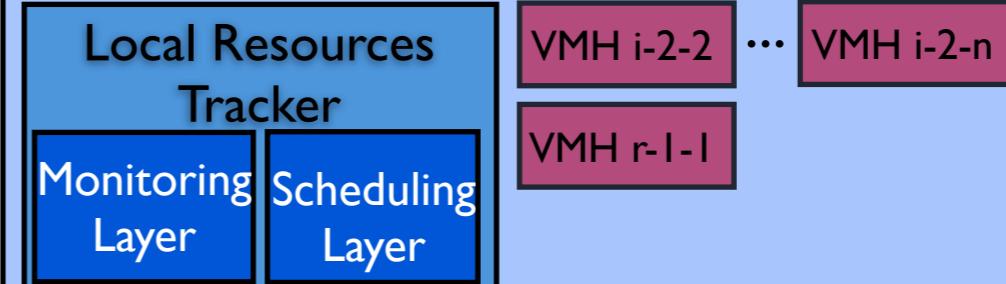
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



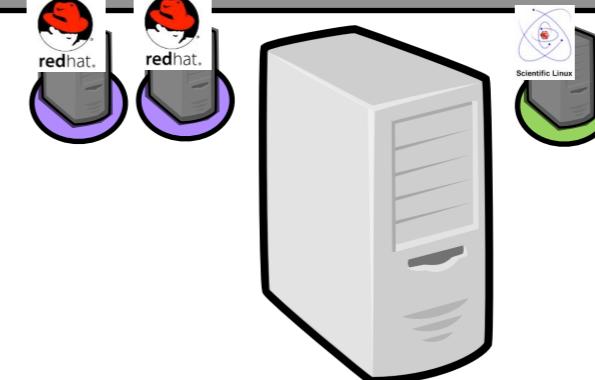
DISCOVERY Agent node r



Discovery Network Tracker



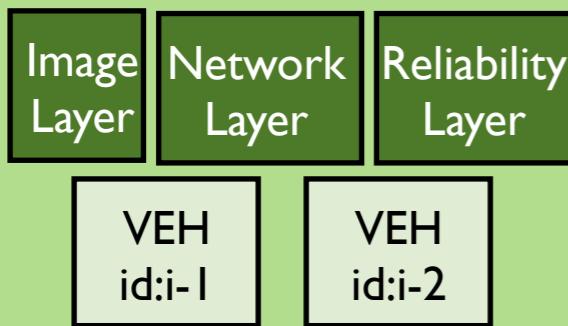
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY - Human removals

DISCOVERY Agent node i

Virtual Environments Tracker i



Discovery Network Tracker

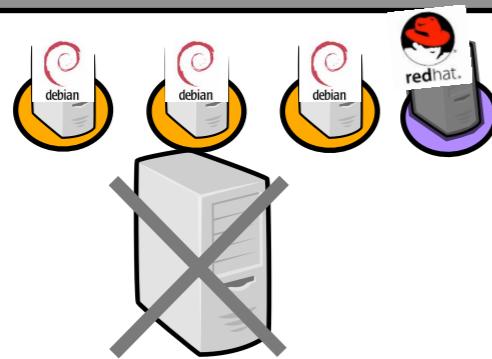
Local Resources Tracker
Monitoring Layer

Scheduling Layer

VMH i-1-1 ... VMH i-1-n

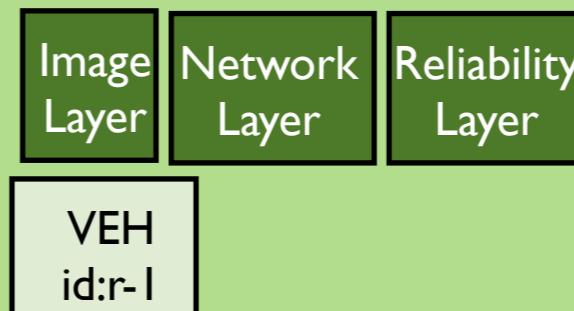
VMH i-2-1

Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r

Virtual Environments Tracker r



Discovery Network Tracker

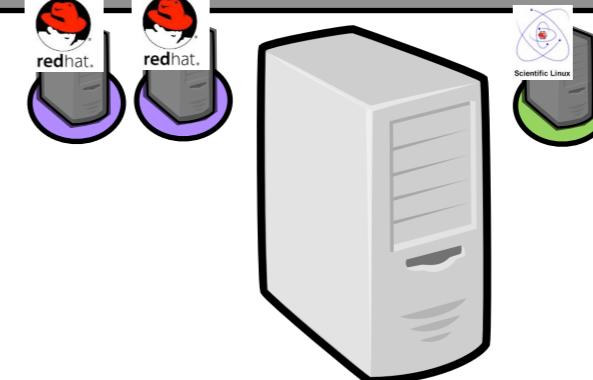
Local Resources Tracker
Monitoring Layer

Scheduling Layer

VMH i-2-2 ... VMH i-2-n

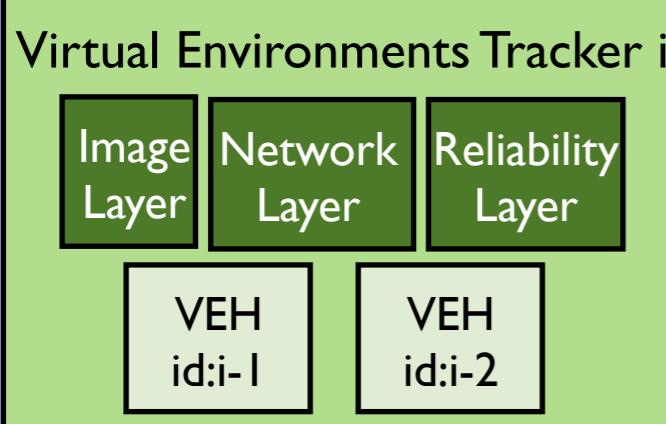
VMH r-1-1

Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

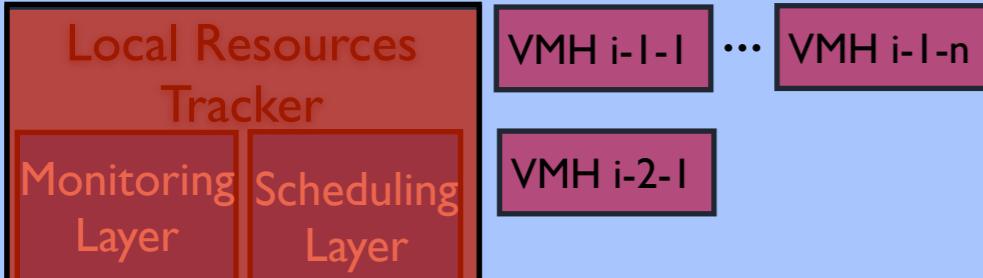


DISCOVERY - Human removals

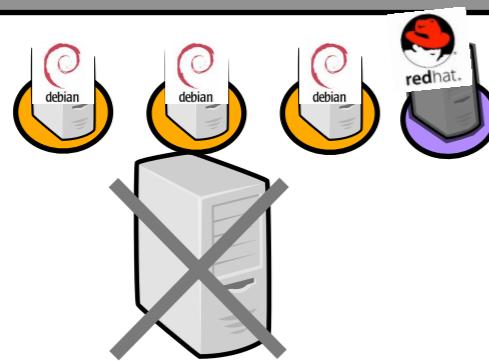
DISCOVERY Agent node i



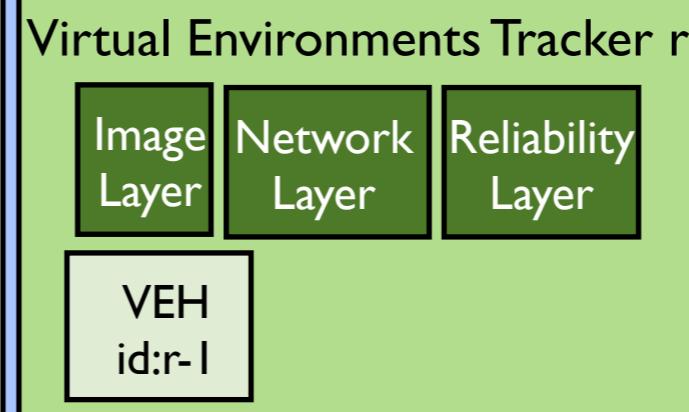
Discovery Network Tracker



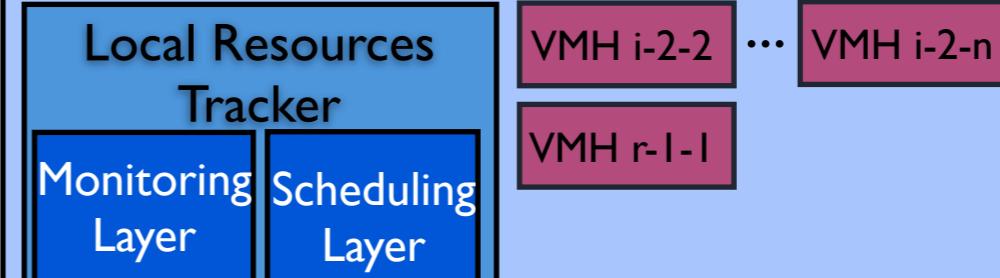
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



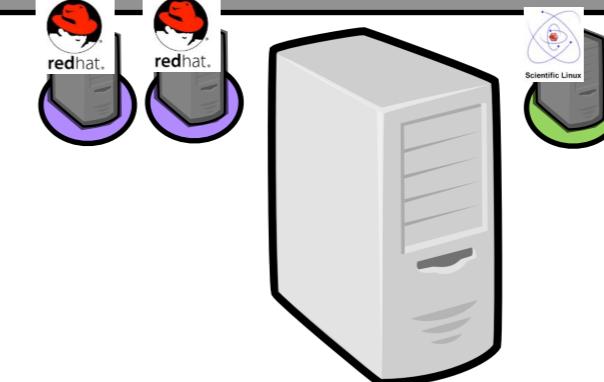
DISCOVERY Agent node r



Discovery Network Tracker

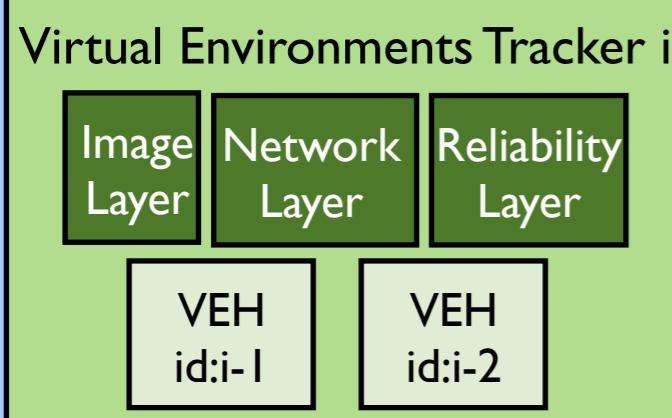


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY - Human removals

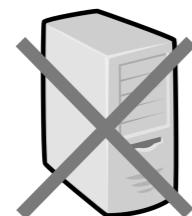
DISCOVERY Agent node i



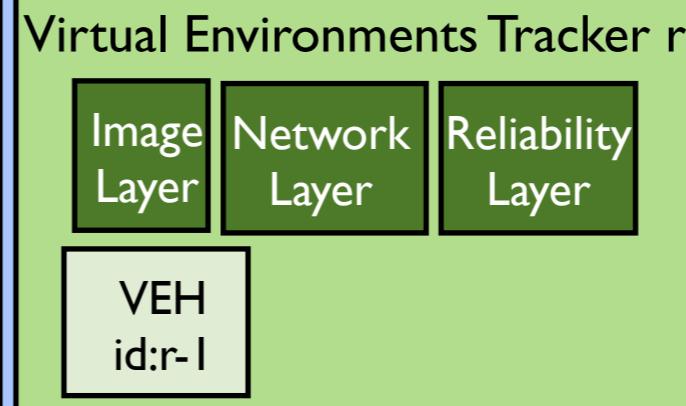
Discovery Network Tracker



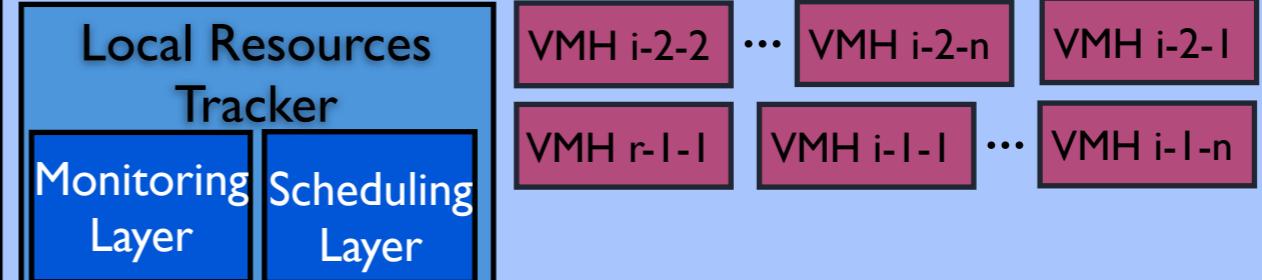
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



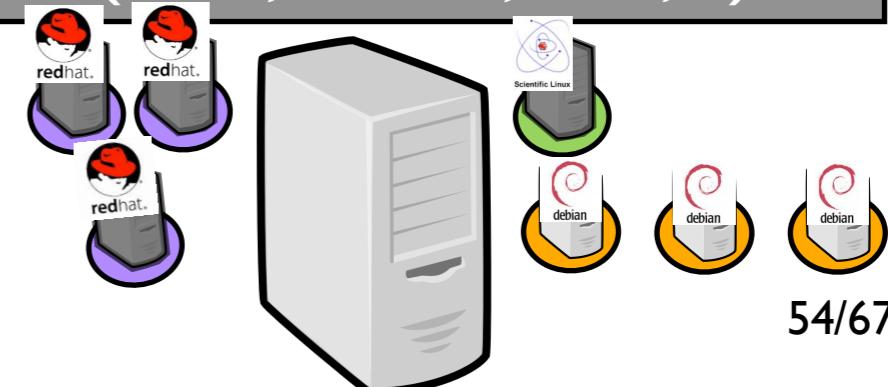
DISCOVERY Agent node r



Discovery Network Tracker



Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY - Human removals

DISCOVERY Agent node i

Discovery Network Tracker

Local Resources Tracker
Monitoring Layer Scheduling Layer

Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r

Virtual Environments Tracker r

Image Layer Network Layer Reliability Layer
VEH id:r-1

Virtual Environments Tracker i

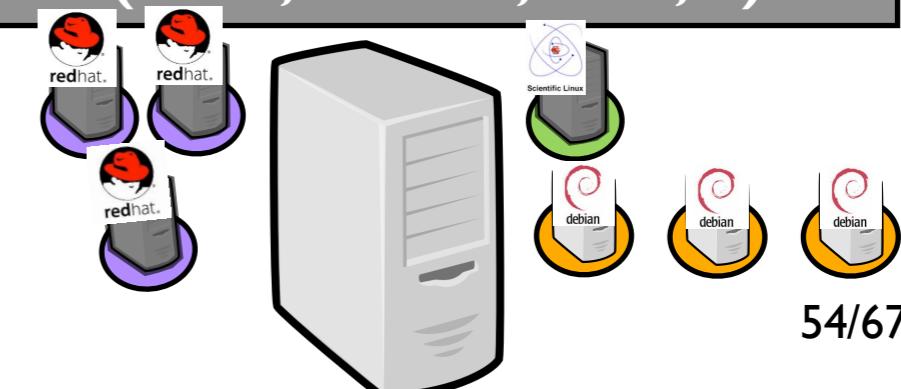
Image Layer Network Layer Reliability Layer
VEH id:i-1 VEH id:i-2

Discovery Network Tracker

Local Resources Tracker
Monitoring Layer Scheduling Layer

VMH i-2-2 ... VMH i-2-n VMH i-2-1
VMH r-1-1 VMH i-1-1 ... VMH i-1-n

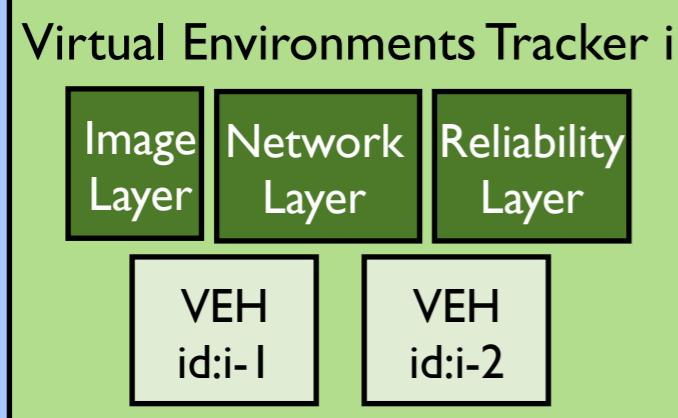
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



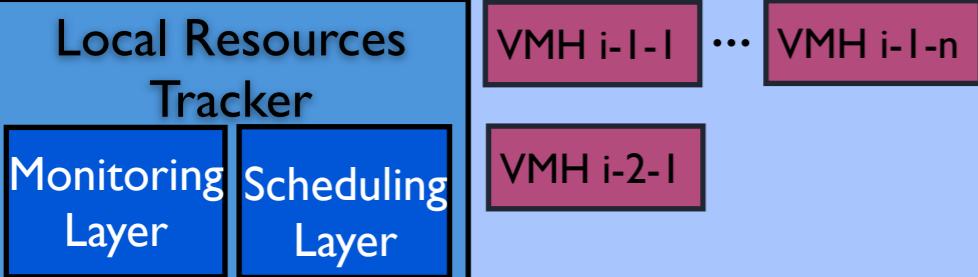
54/67

DISCOVERY - VM Crashes

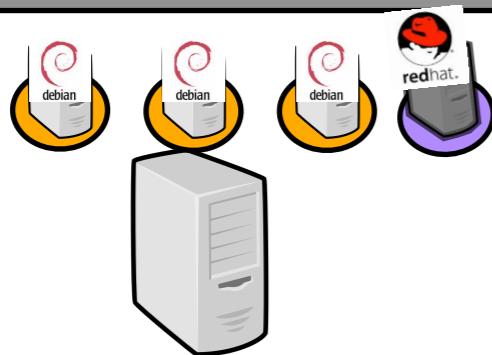
DISCOVERY Agent node i



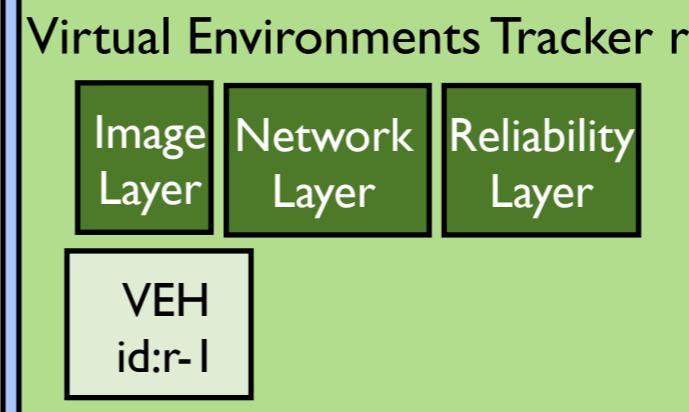
Discovery Network Tracker



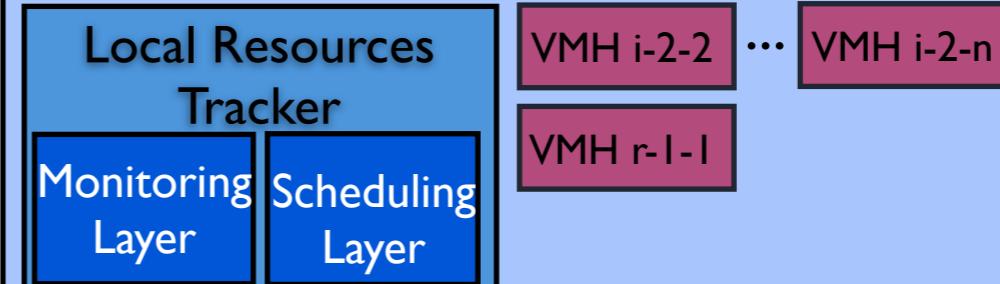
**Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)**



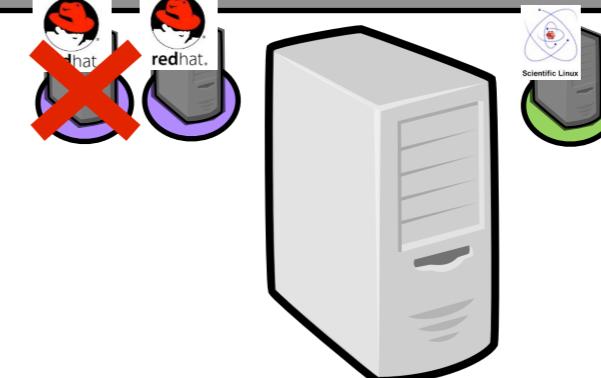
DISCOVERY Agent node r



Discovery Network Tracker

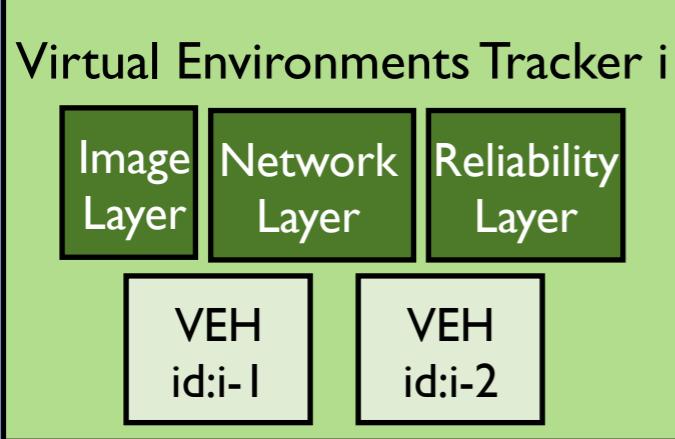


**Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)**

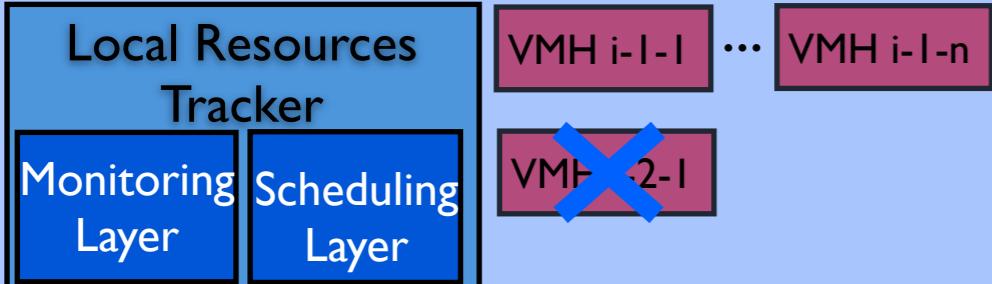


DISCOVERY - VM Crashes

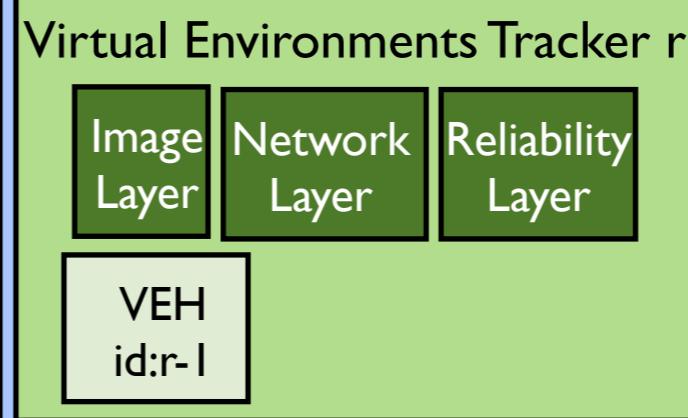
DISCOVERY Agent node i



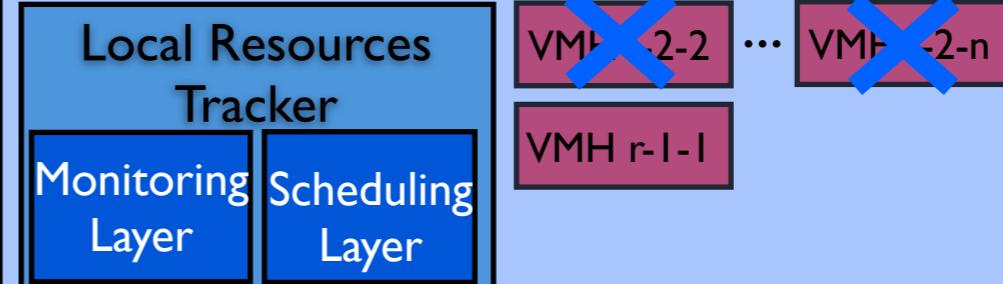
Discovery Network Tracker



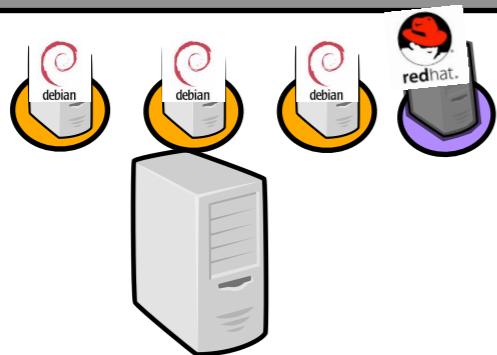
DISCOVERY Agent node r



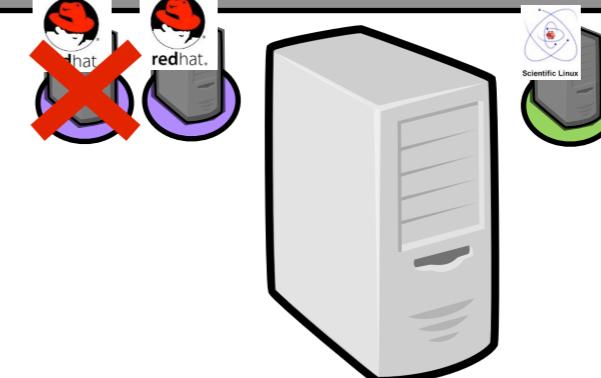
Discovery Network Tracker



Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

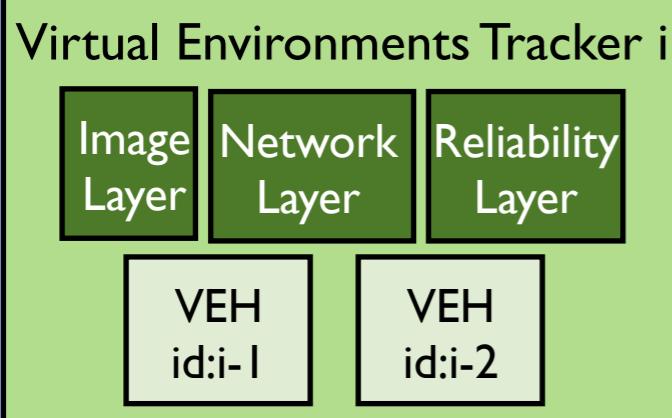


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

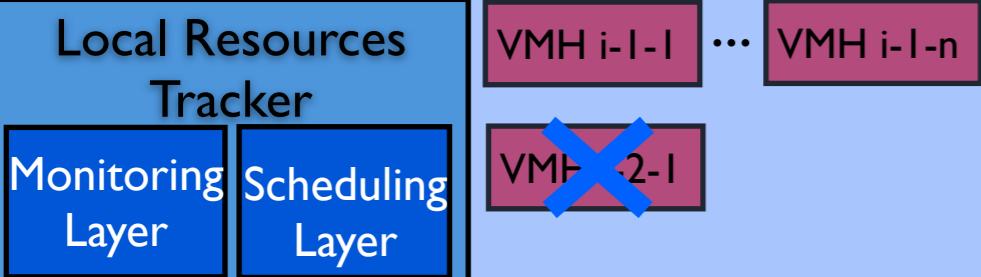


DISCOVERY - VM Crashes

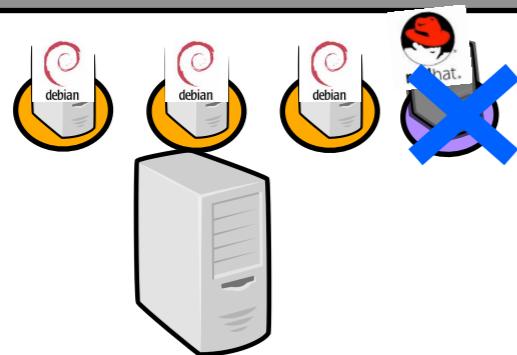
DISCOVERY Agent node i



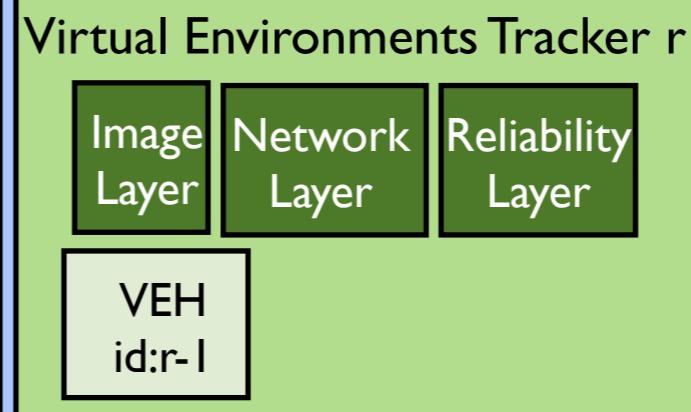
Discovery Network Tracker



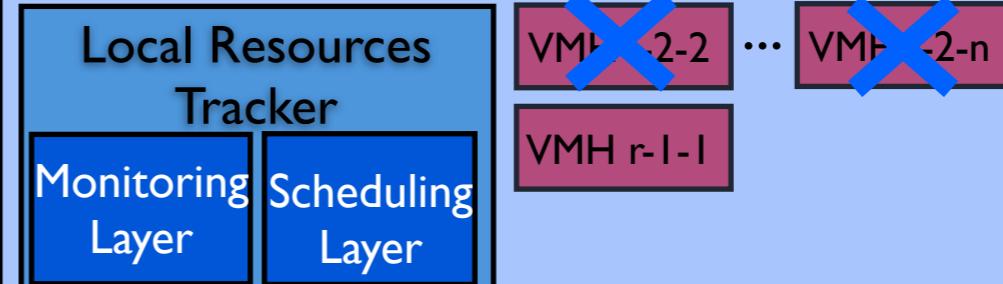
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



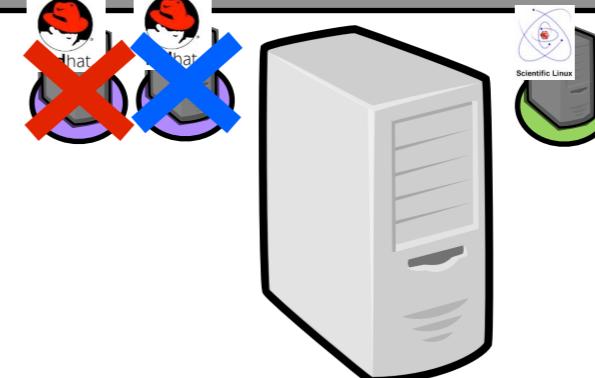
DISCOVERY Agent node r



Discovery Network Tracker

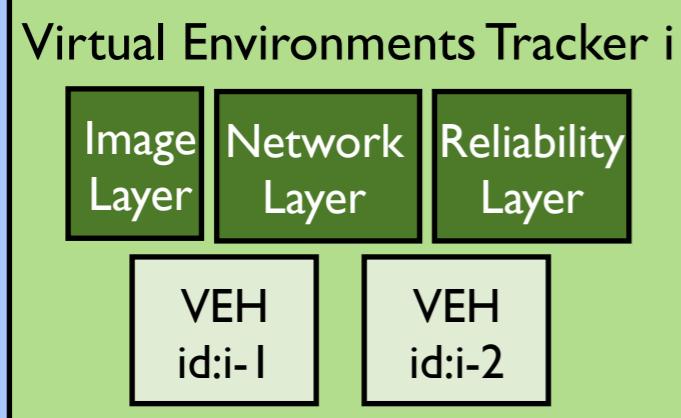


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

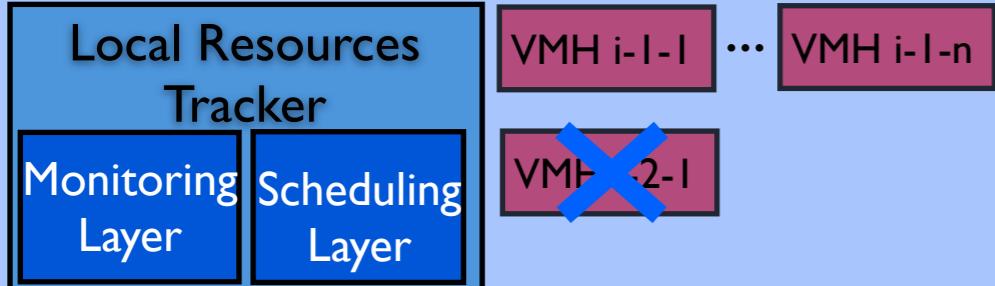


DISCOVERY - VM Crashes

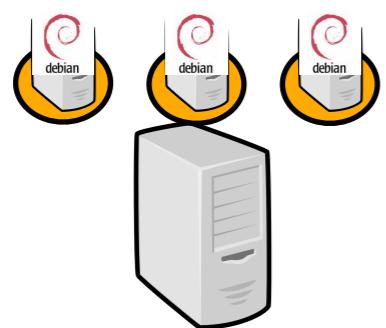
DISCOVERY Agent node i



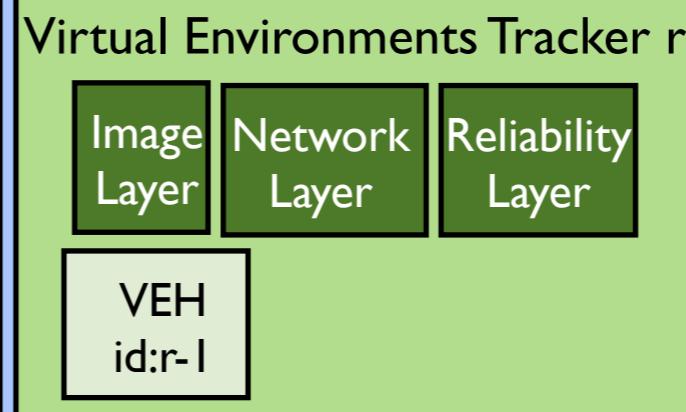
Discovery Network Tracker



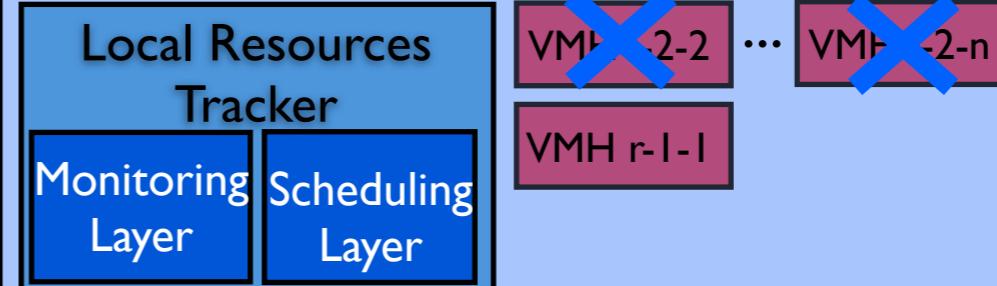
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r



Discovery Network Tracker

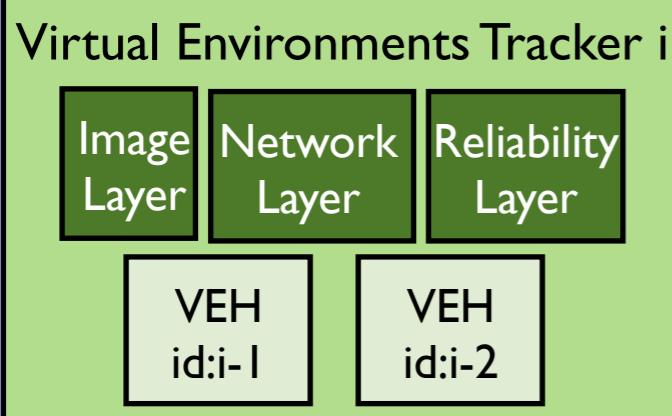


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

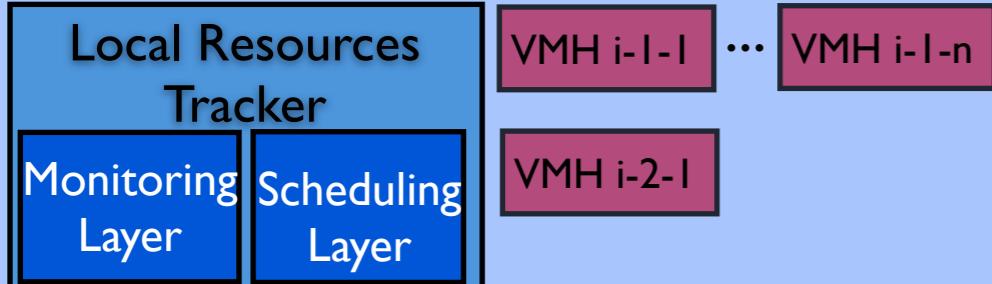


DISCOVERY - VM Crashes

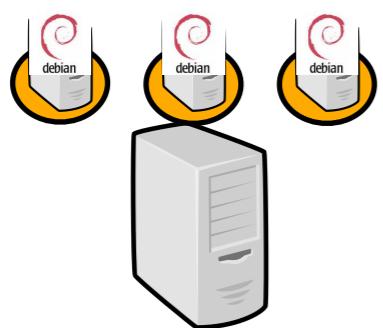
DISCOVERY Agent node i



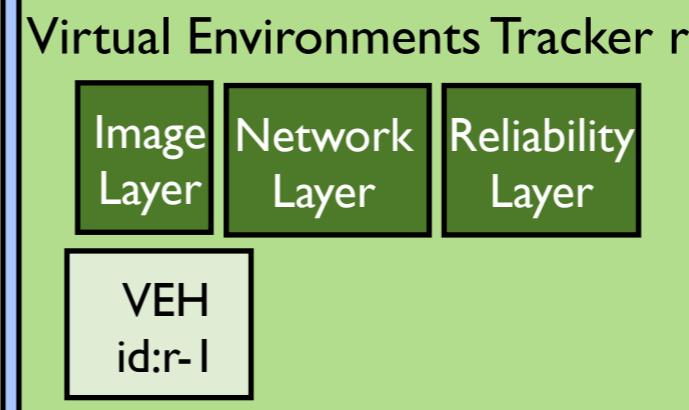
Discovery Network Tracker



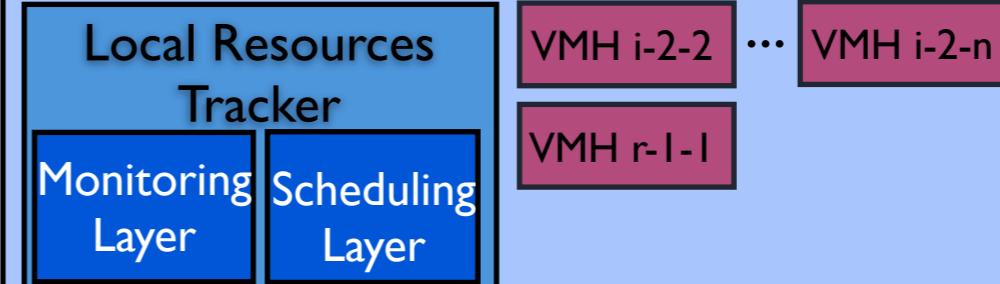
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r



Discovery Network Tracker

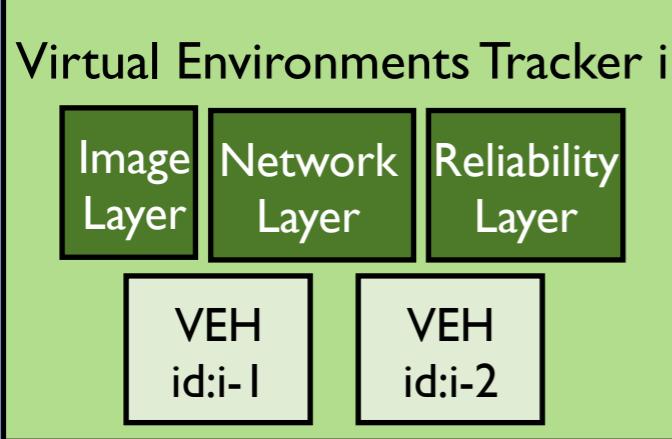


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

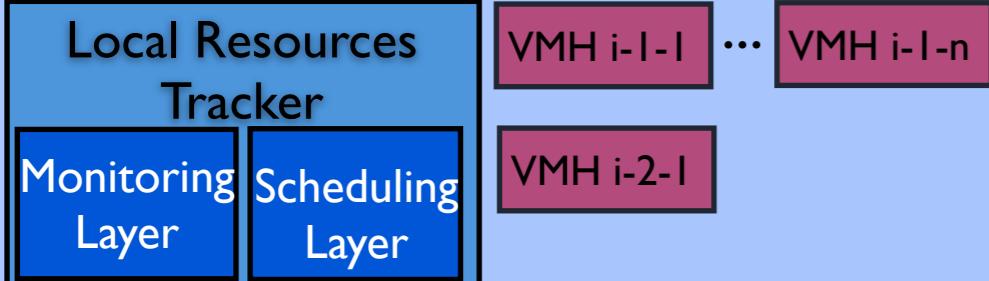


DISCOVERY - VM Crashes

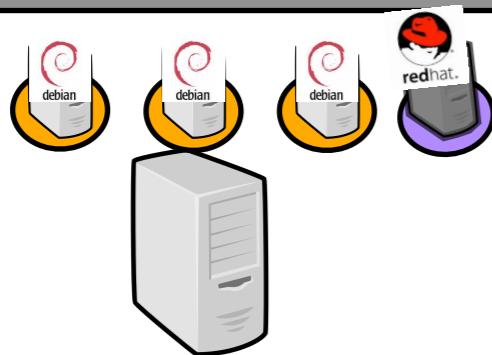
DISCOVERY Agent node i



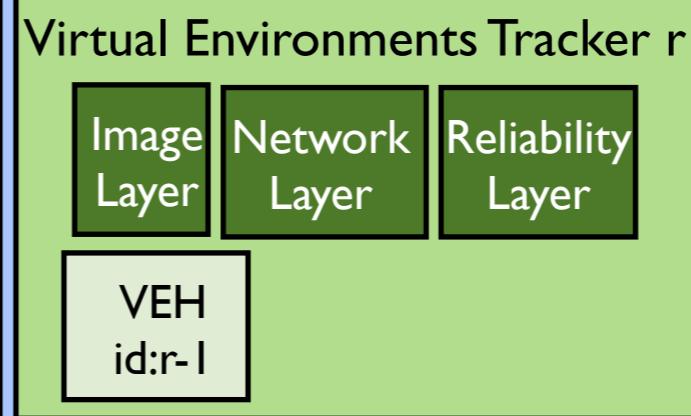
Discovery Network Tracker



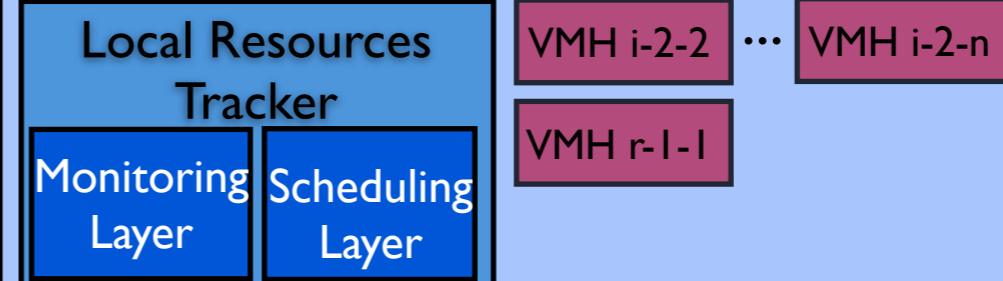
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



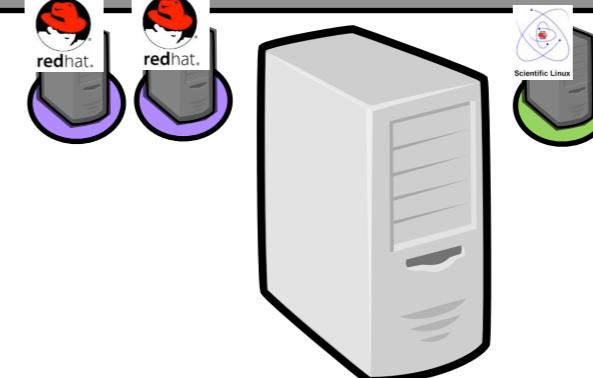
DISCOVERY Agent node r



Discovery Network Tracker

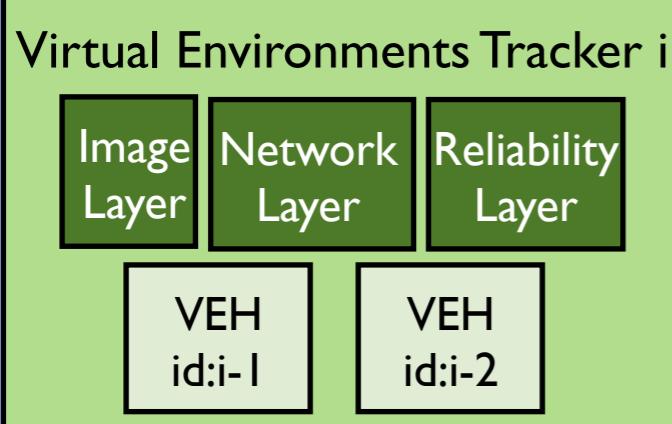


Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)

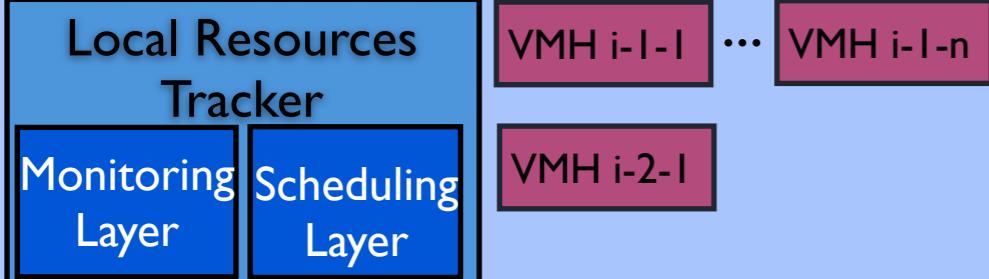


DISCOVERY - Nodes Crashes

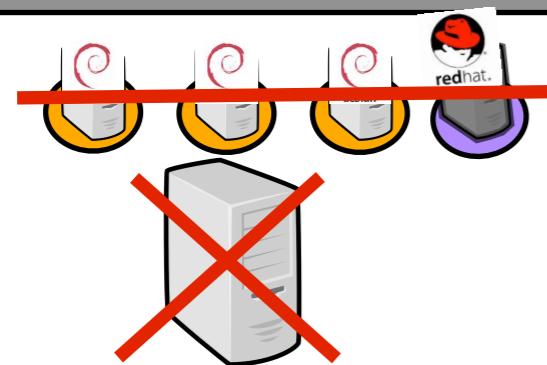
DISCOVERY Agent node i



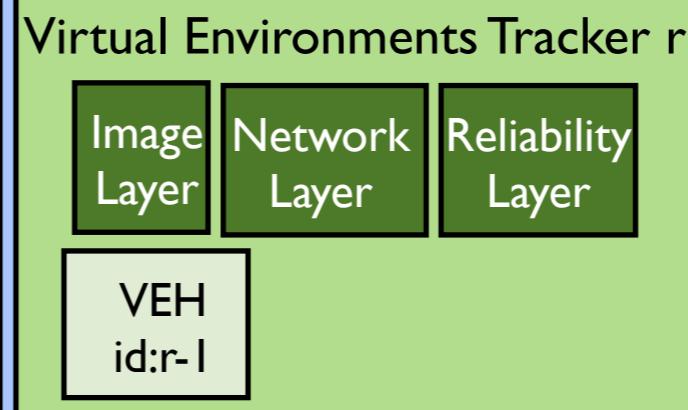
Discovery Network Tracker



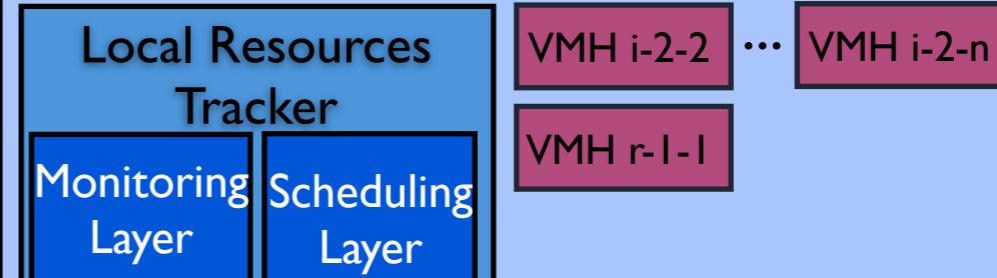
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



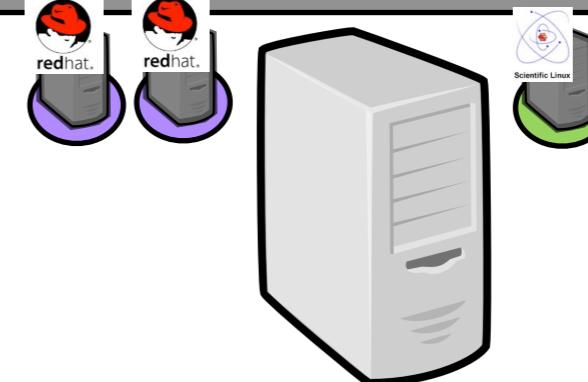
DISCOVERY Agent node r



Discovery Network Tracker



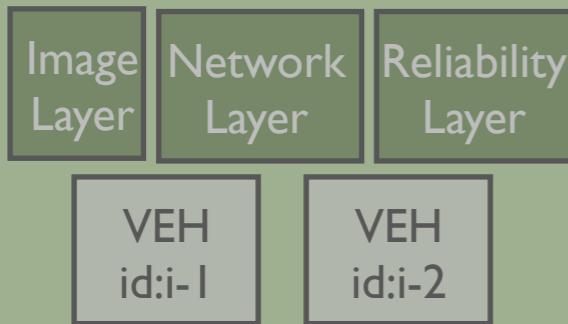
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY - Nodes Crashes

DISCOVERY Agent node i

Virtual Environments Tracker i



Discovery Network Tracker

Local Resources Tracker

Monitoring Layer Scheduling Layer

VMH i-1-1 ... VMH i-1-n

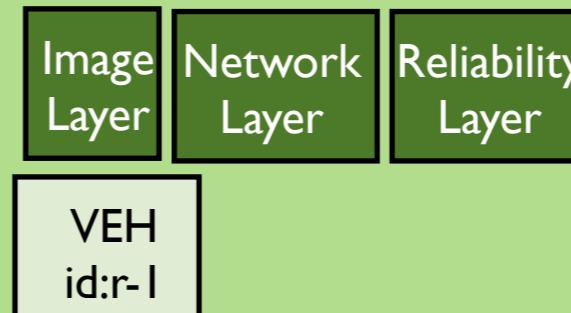
VMH i-2-1

Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r

Virtual Environments Tracker r



Discovery Network Tracker

Local Resources Tracker

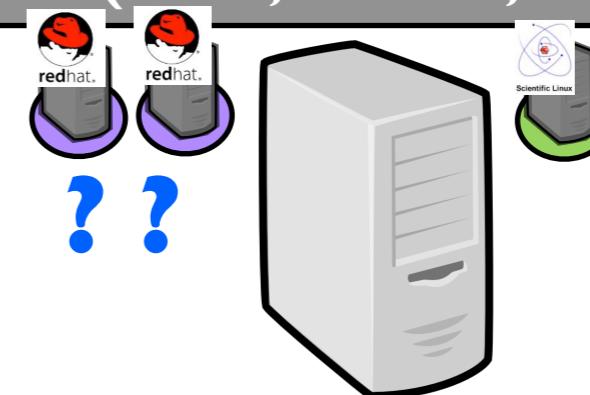
Monitoring Layer Scheduling Layer

VMH i-2-2 ... VMH i-2-n

VMH r-1-1

?

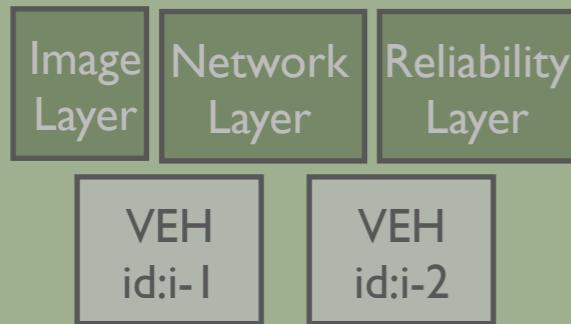
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY - Nodes Crashes

DISCOVERY Agent node i

Virtual Environments Tracker i



Discovery Network Tracker

Local Resources Tracker

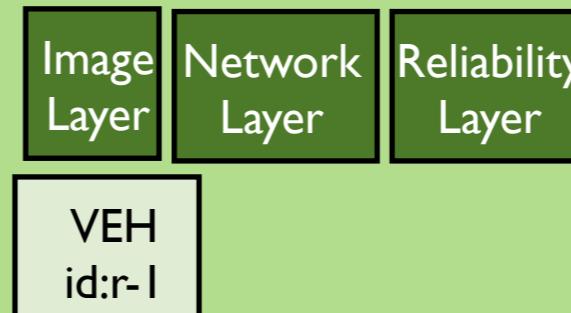
Monitoring Layer Scheduling Layer

Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r

Virtual Environments Tracker r



Discovery Network Tracker

Local Resources Tracker

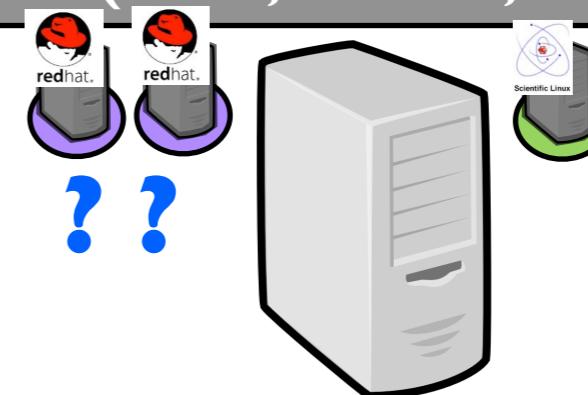
Monitoring Layer Scheduling Layer

VMH i-2-2

... VMH i-2-n

?

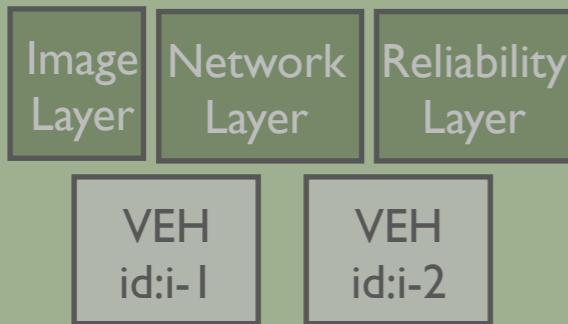
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY - Nodes Crashes

DISCOVERY Agent node i

Virtual Environments Tracker i



Discovery Network Tracker

Local Resources Tracker

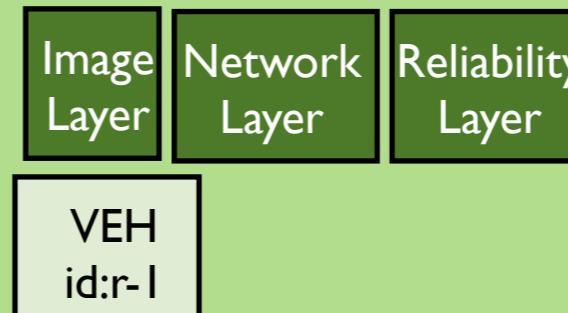
Monitoring Layer Scheduling Layer

Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r

Virtual Environments Tracker r



Discovery Network Tracker

Local Resources Tracker

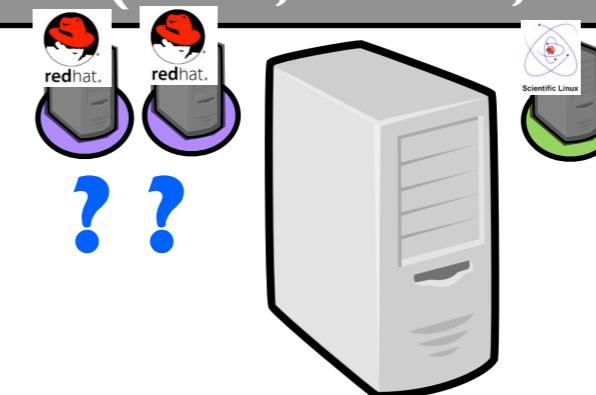
Monitoring Layer Scheduling Layer

VMH i-2-2 ... VMH i-2-n

VMH r-1-1

?

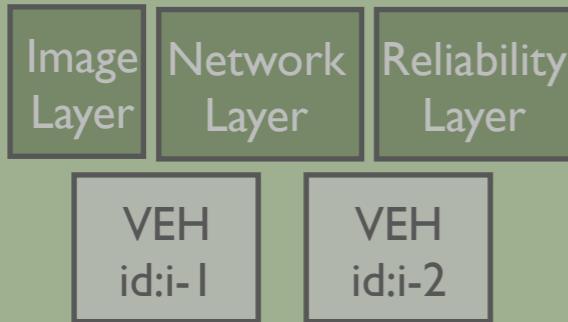
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY - Nodes Crashes

DISCOVERY Agent node i

Virtual Environments Tracker i



Discovery Network Tracker

Local Resources Tracker

Monitoring Layer

VMH i-1-1 ... VMH i-1-n

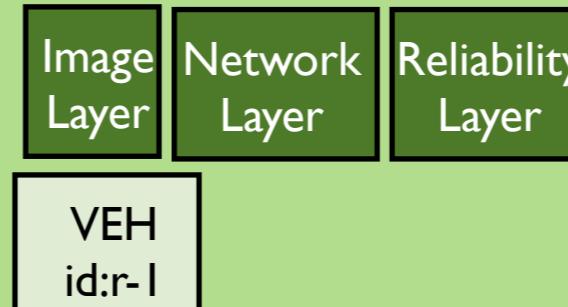
VMH i-2-1

Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r

Virtual Environments Tracker r



Discovery Network Tracker

Local Resources Tracker

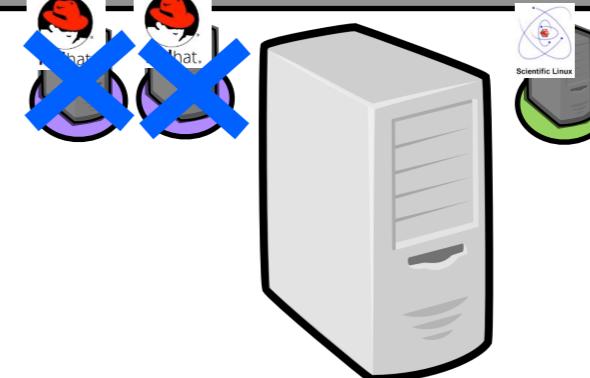
Monitoring Layer

VM i-2-2 ... VM i-2-n

VMH r-1-1

VMH i-1-1 ... VMH i-1-n

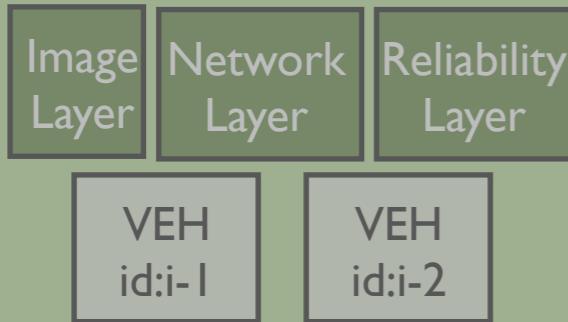
Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY - Nodes Crashes

DISCOVERY Agent node i

Virtual Environments Tracker i



Discovery Network Tracker

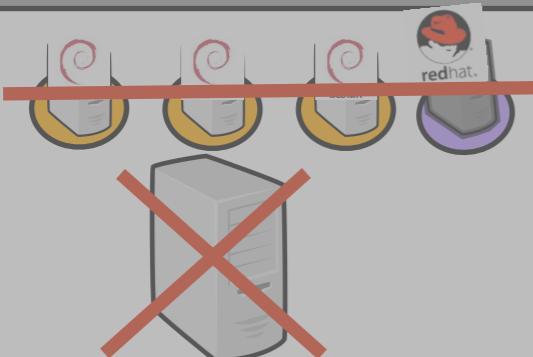
Local Resources Tracker

Monitoring Layer Scheduling Layer

VMH i-1-1 ... VMH i-1-n

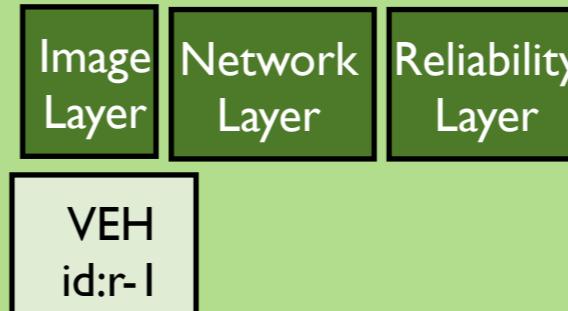
VMH i-2-1

Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



DISCOVERY Agent node r

Virtual Environments Tracker r



Discovery Network Tracker

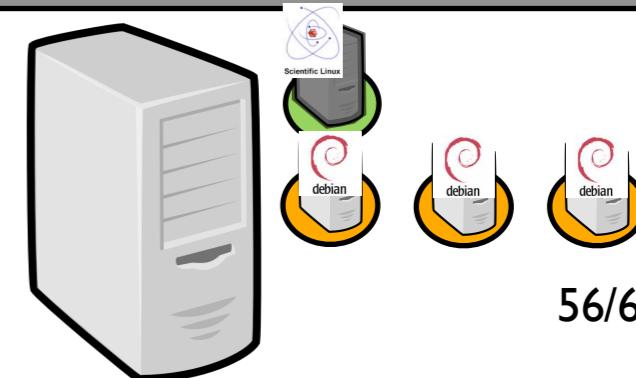
Local Resources Tracker

Monitoring Layer Scheduling Layer

VMH r-1-1

VMH i-1-1 ... VMH i-1-n

Bare Hardware Wrapper
(libvirt, xentools, OCCI, ...)



Discovery from the Software Programming Point of View

Background - Actor Model

- Model for concurrent computation
- Actors are the primitive for parallel computing
- Actors communicate with messages
- Actors process only one message at a time
- No shared state between actors

No lock for data (no barriers)
scalability



- “Let it crash” pattern

Erlang
Fault tolerance



Supervisor and Peer concept

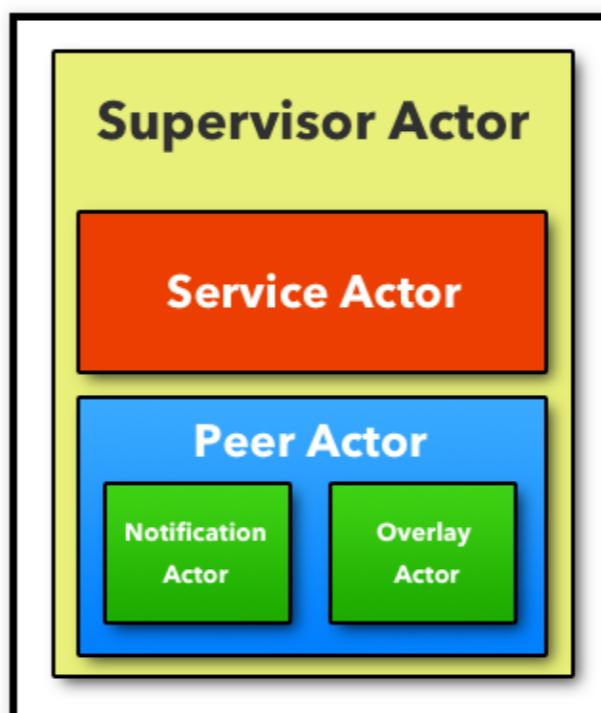
- **PeerActor**: primitive for developing an application over an overlay (abstraction of the network concerns)

OverlayActor: Implement the overlay

NotificationActor: Event bus (overlay modifications)

- **SupervisorActor**: monitoring of sub actors

Analyze failures causes, decides what to do, ...

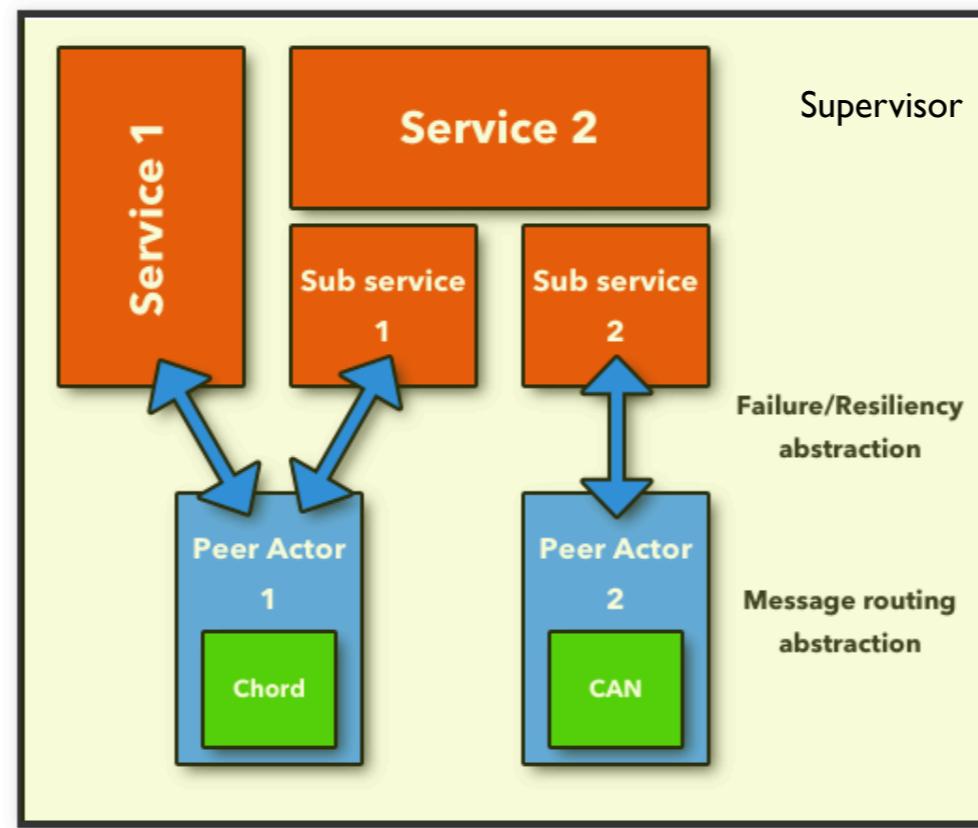


Credits: J. Pastor - The PeerActor Model, 2013

59/67

The Discovery Software Programming Model

- Discovery system will be composed of several services
- Each service will use one or more PeerActor
- Each service will be monitored by a SupervisorActor
- Resiliency will be handled by SupervisorActor



60/67

The Discovery Software Programming

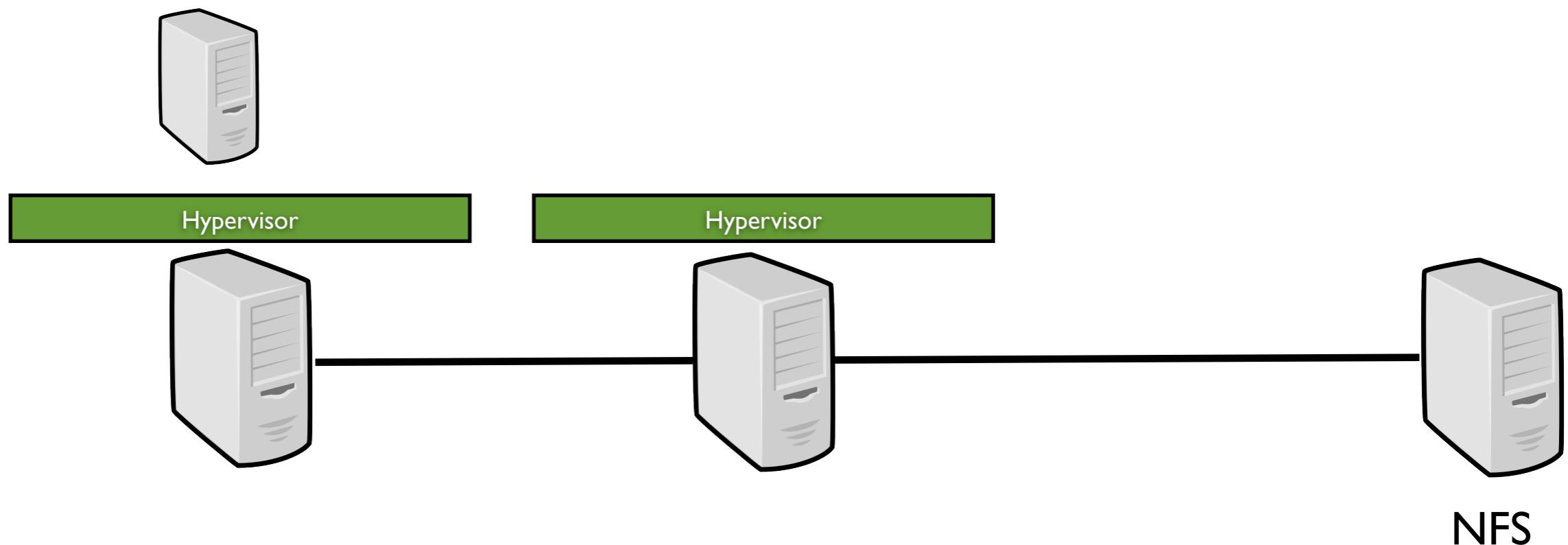
- AkkaArc
 - PeerActor + SupervisorActor definitions (Scala/Akka)
 - A Chord Implementation
- Reimplementation of the DVMS proposal with fault tolerant mechanisms in less than 3 months 
- Implementation of distributed IPOP like system (on-going work)

Conclusion

- System Virtualization provides mature “[techniques for organizing computer systems resources to provide extraordinary system flexibility](#)” Golberg, 1974
- Leveraging previous works is mandatory
- System Virtualization leads to new concerns
(which have not been addressed in this talk)
 - Storage (management of VM images/VM migrations through distinct sites)
 - Network management (IP assignment, VLAN configurations, suspend/resume)
 - Coordination between users and administrators expectations

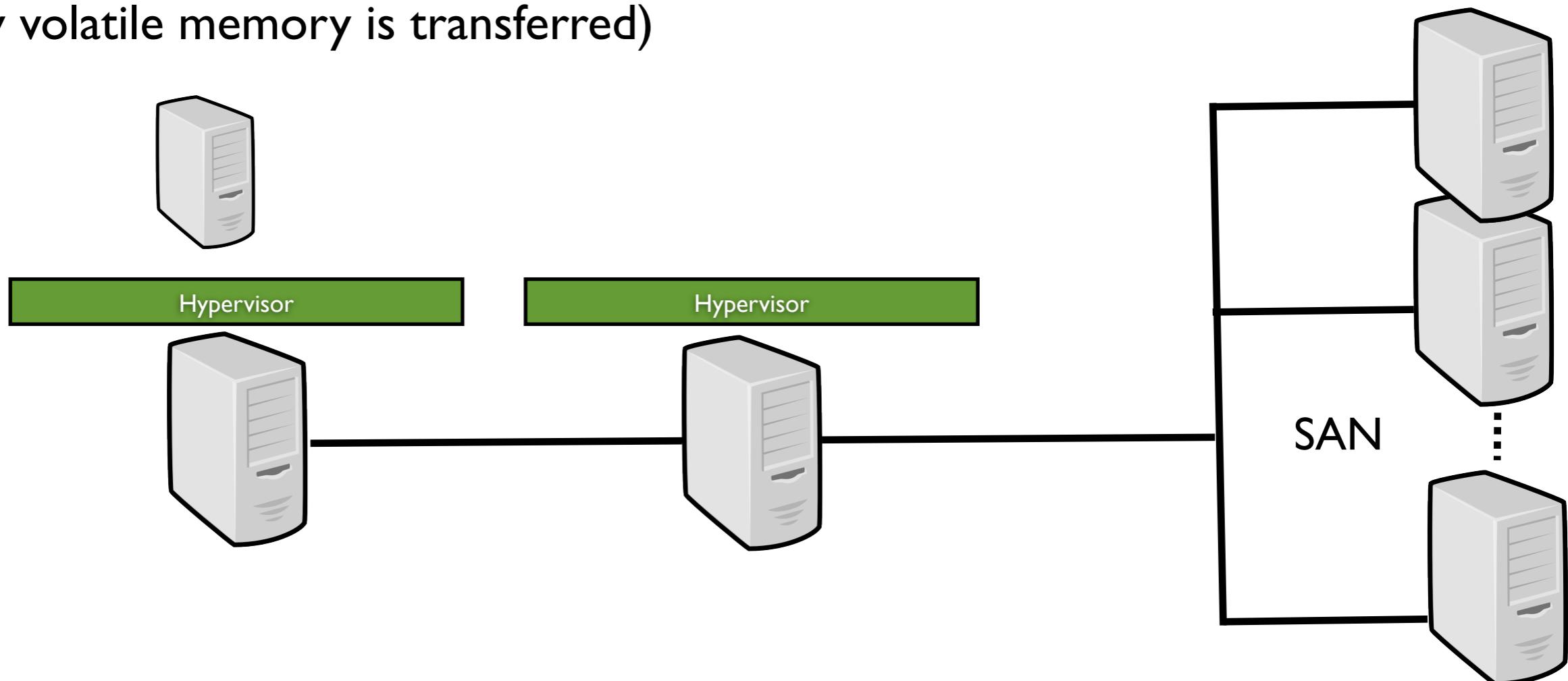
Conclusion

- VM : volatile states vs persistent ones
 - VM images (AMI Amazon Machine Image)
- VM migration requires efficient storage mechanisms
 - Exploit a distributed file system (NAS/SAN)
(only volatile memory is transferred)



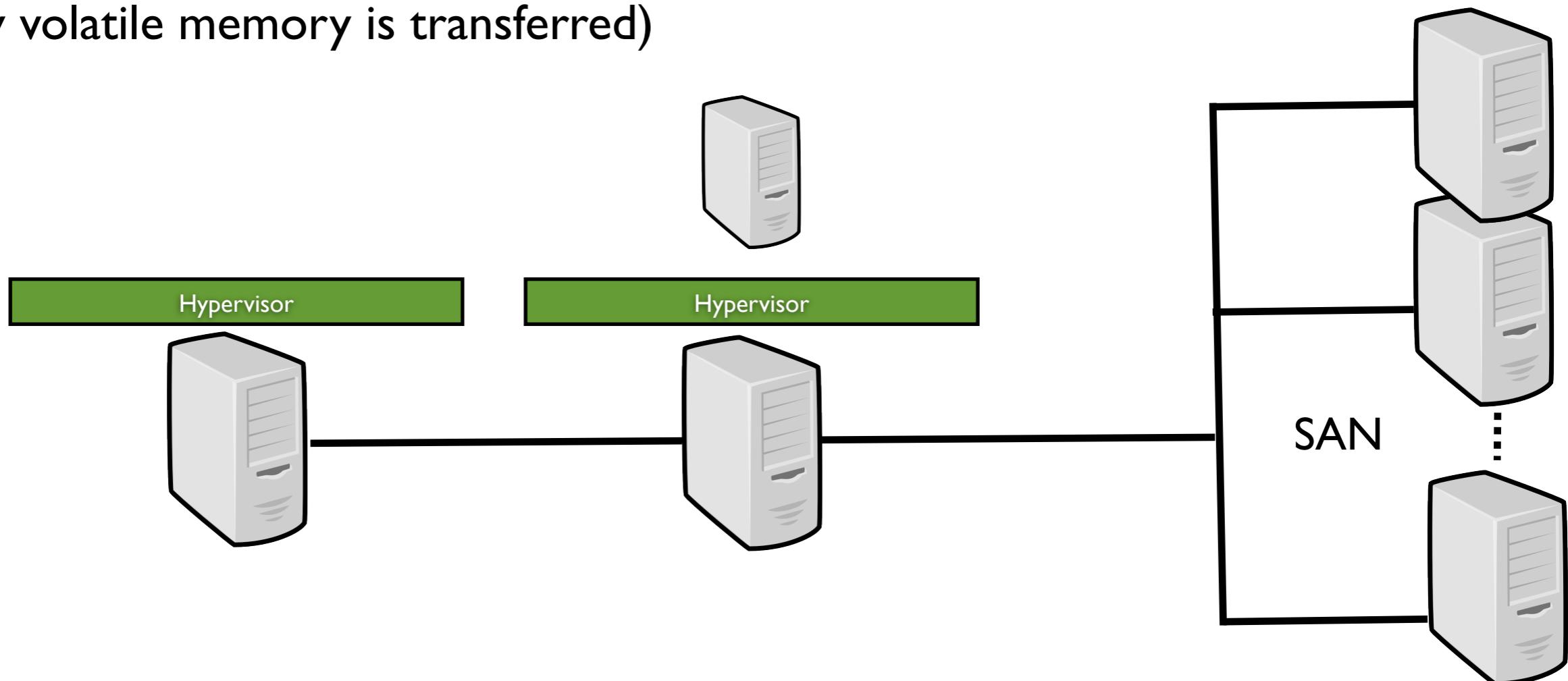
Conclusion

- VM : volatile states vs persistent ones
 - VM images (AMI Amazon Machine Image)
- VM migration requires efficient storage mechanisms
 - Exploit a distributed file system (NAS/SAN)
(only volatile memory is transferred)



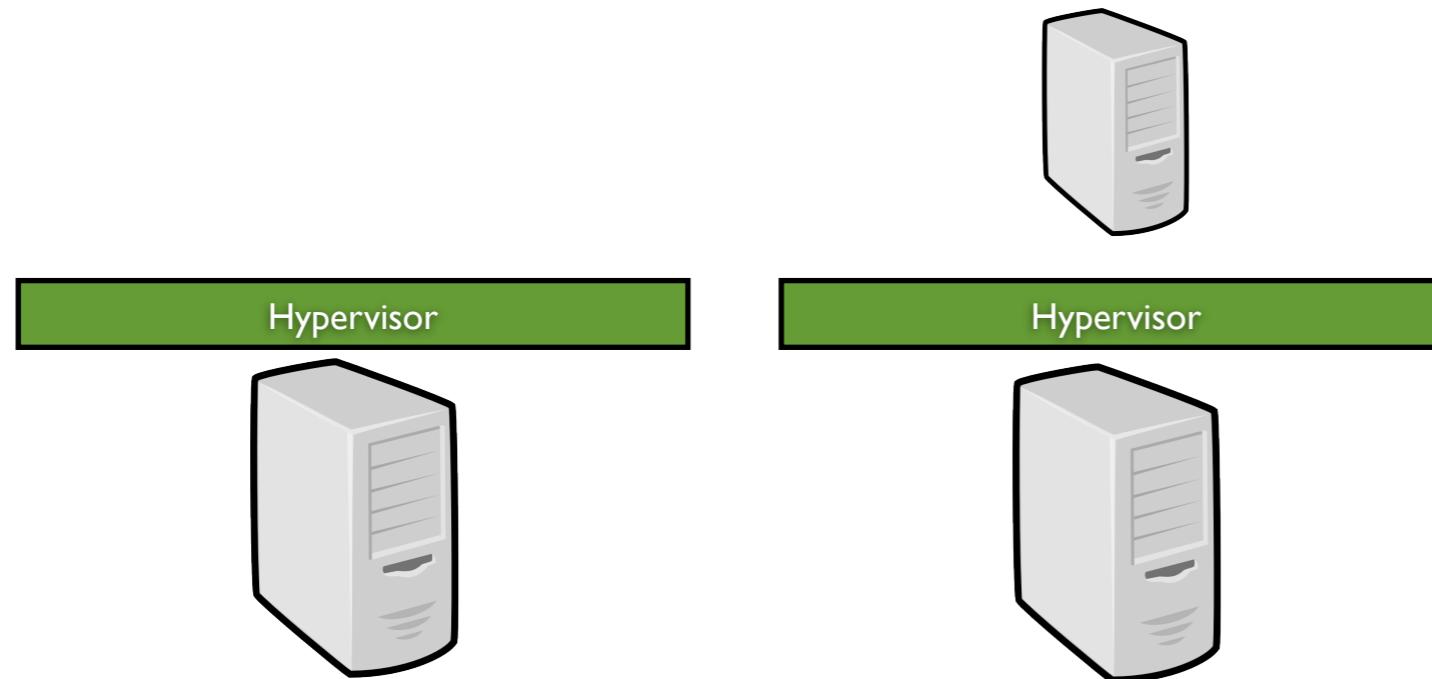
Conclusion

- VM : volatile states vs persistent ones
 - VM images (AMI Amazon Machine Image)
- VM migration requires efficient storage mechanisms
 - Exploit a distributed file system (NAS/SAN)
(only volatile memory is transferred)



Conclusion

- VM : volatile states vs persistent ones
 - VM images (AMI Amazon Machine Image)
- VM migration requires efficient storage mechanisms
 - Exploit a distributed file system (NAS/SAN)
(only volatile memory is transferred)



Copy HDD image from the source to the destination node

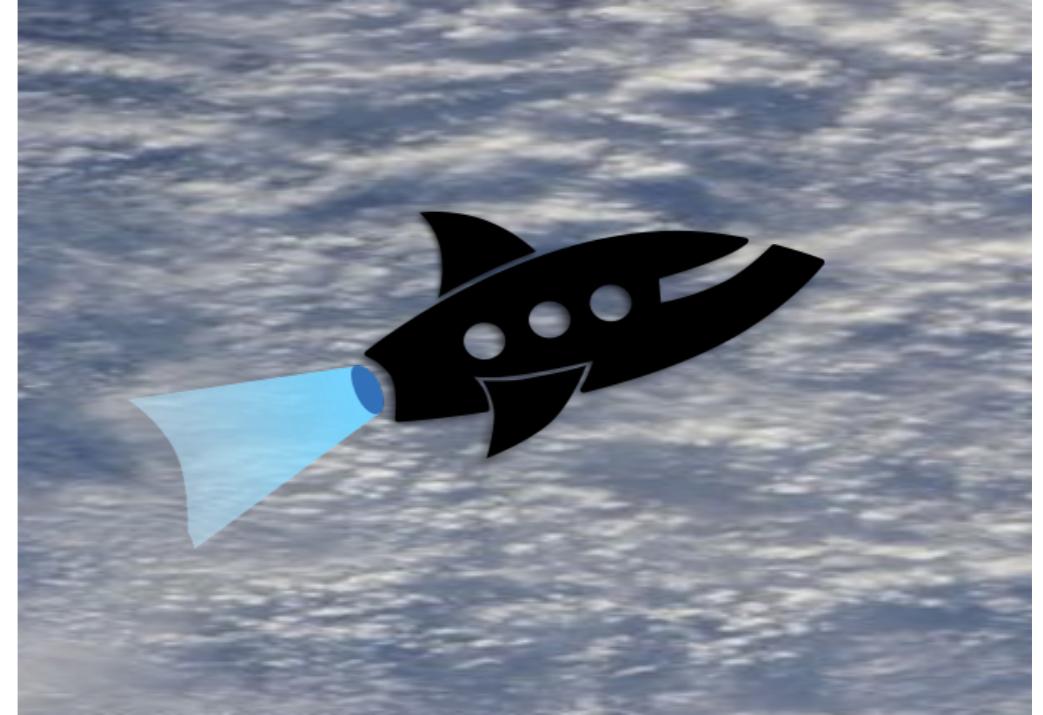
Portable but expensive (some contributions such as qcow/backing file solutions, background pre-copy,..)

Conclusion

- Cloud Computing technology is changing every day
 - New features, new requirements

The main challenge of the Discovery Initiative is to ensure that such new features/mechanisms can run in a distributed manner.
- But Distributed Cloud Computing is happening !
 - Dist. CC workshop (collocated with IEEE/ACM UCC 2013)
 - FOG Computing workshop (collocated with IEEE ICC 2013)

Thanks

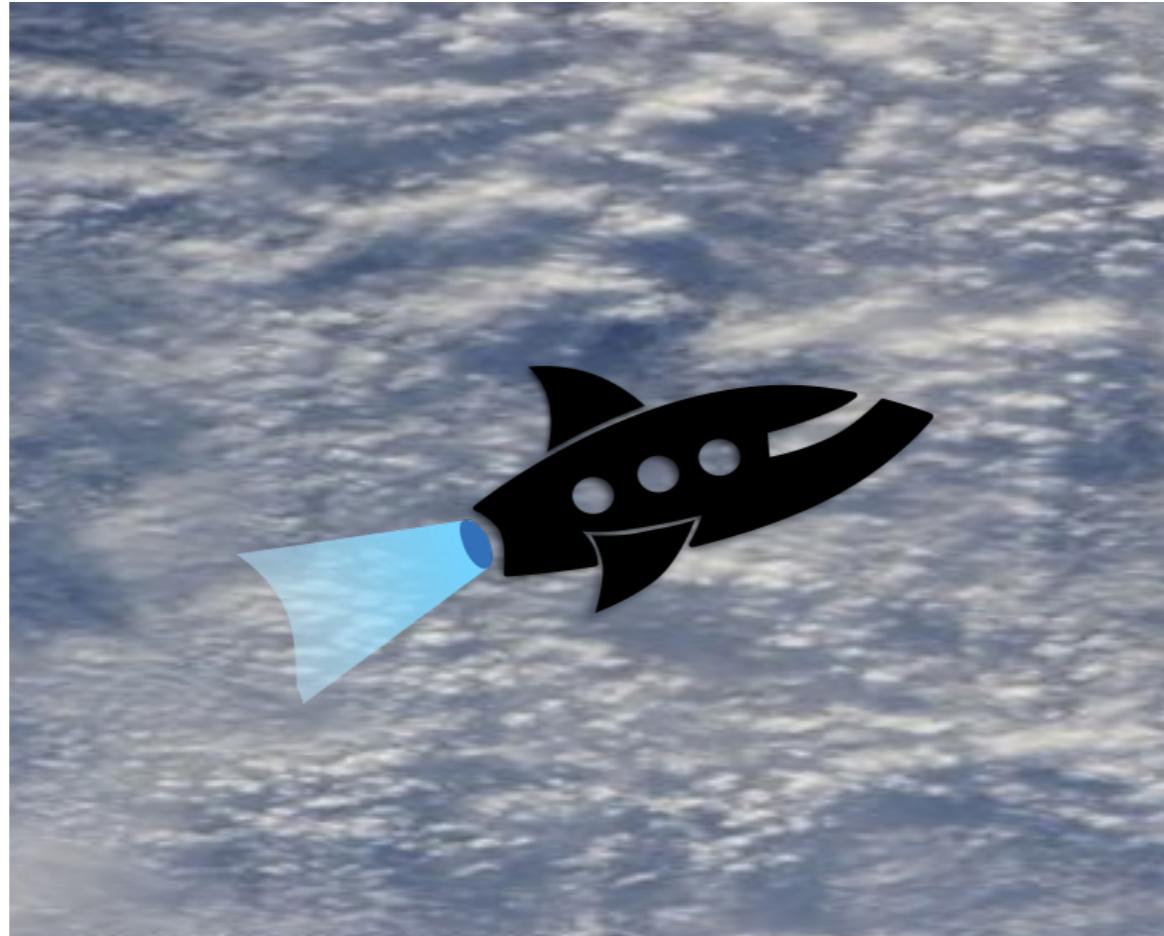


- The Discovery Initiative
Past and on-going contributors

Paolo Anedda, Marin Bertier, Frédéric Desprez, Massimo Gaggero, Fabien Hermenier, Flavien Quesnel, Jean-Marc Menaud, Jonathan Pastor, Rémi Pottier, Etienne Riviere, Thomas Ropars, Jonathan Rouzaud-Cornabas, Cédric Tedeschi, Gianluigi Zanetti...

<http://beyondtheclouds.github.io/>

Beyond the Clouds, the DISCOVERY Initiative



Localization is a key element to deliver
efficient as well as sustainable Utility Computing Solutions



INSTITUT
Mines-Télécom

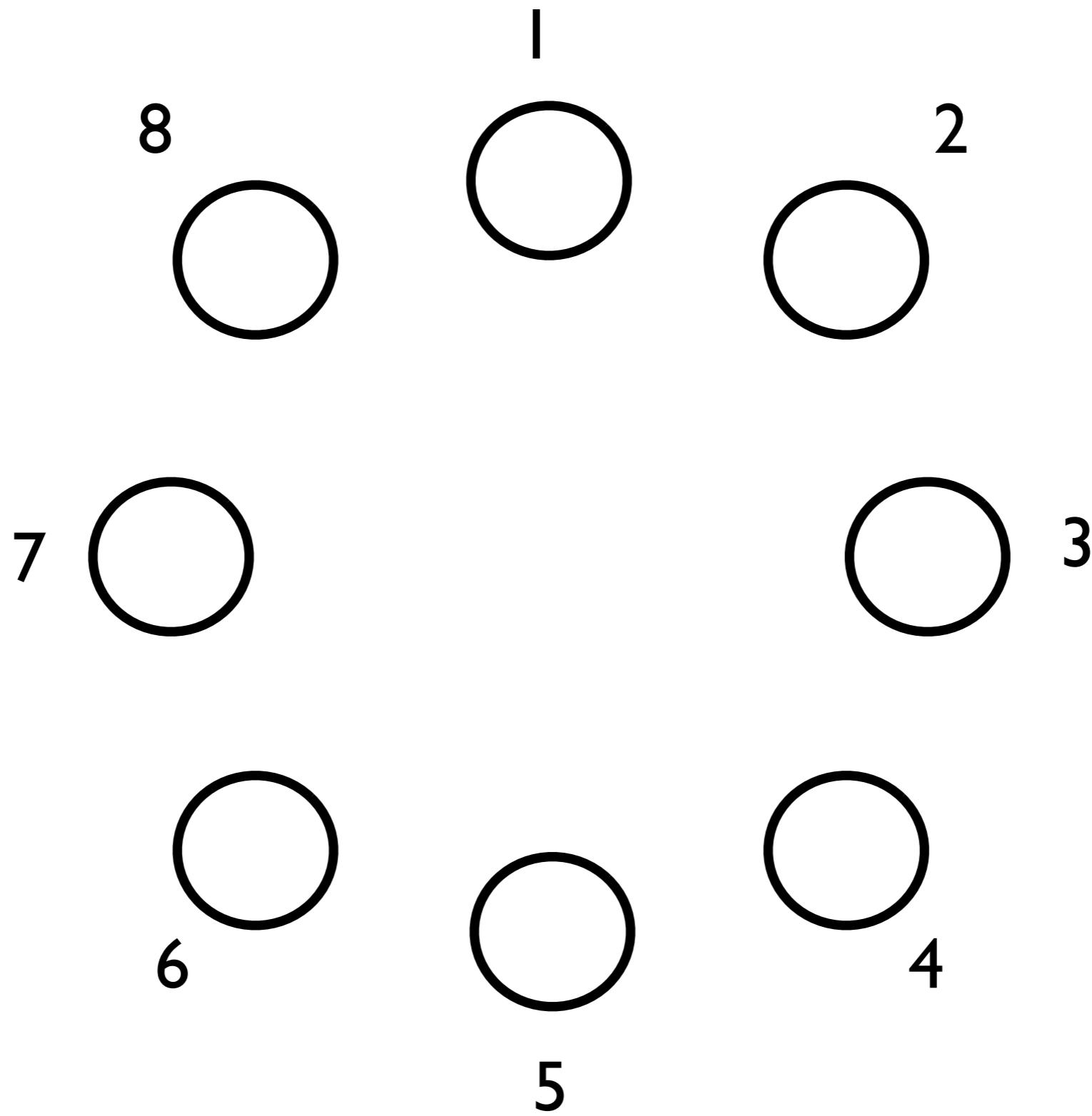


Adrien Lèbre / Ascola Project Team
August, 2013

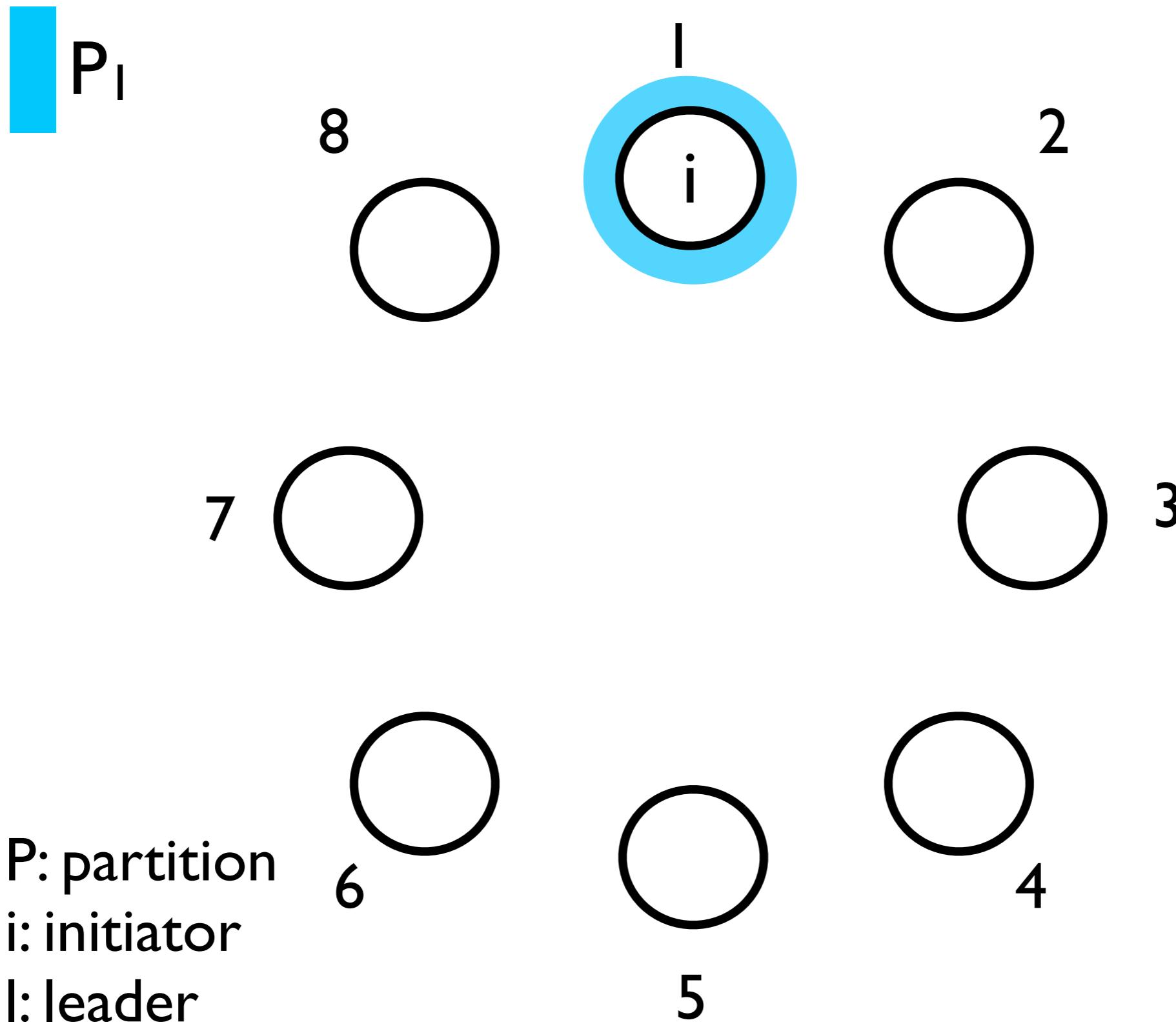


Backup / Advanced Informations

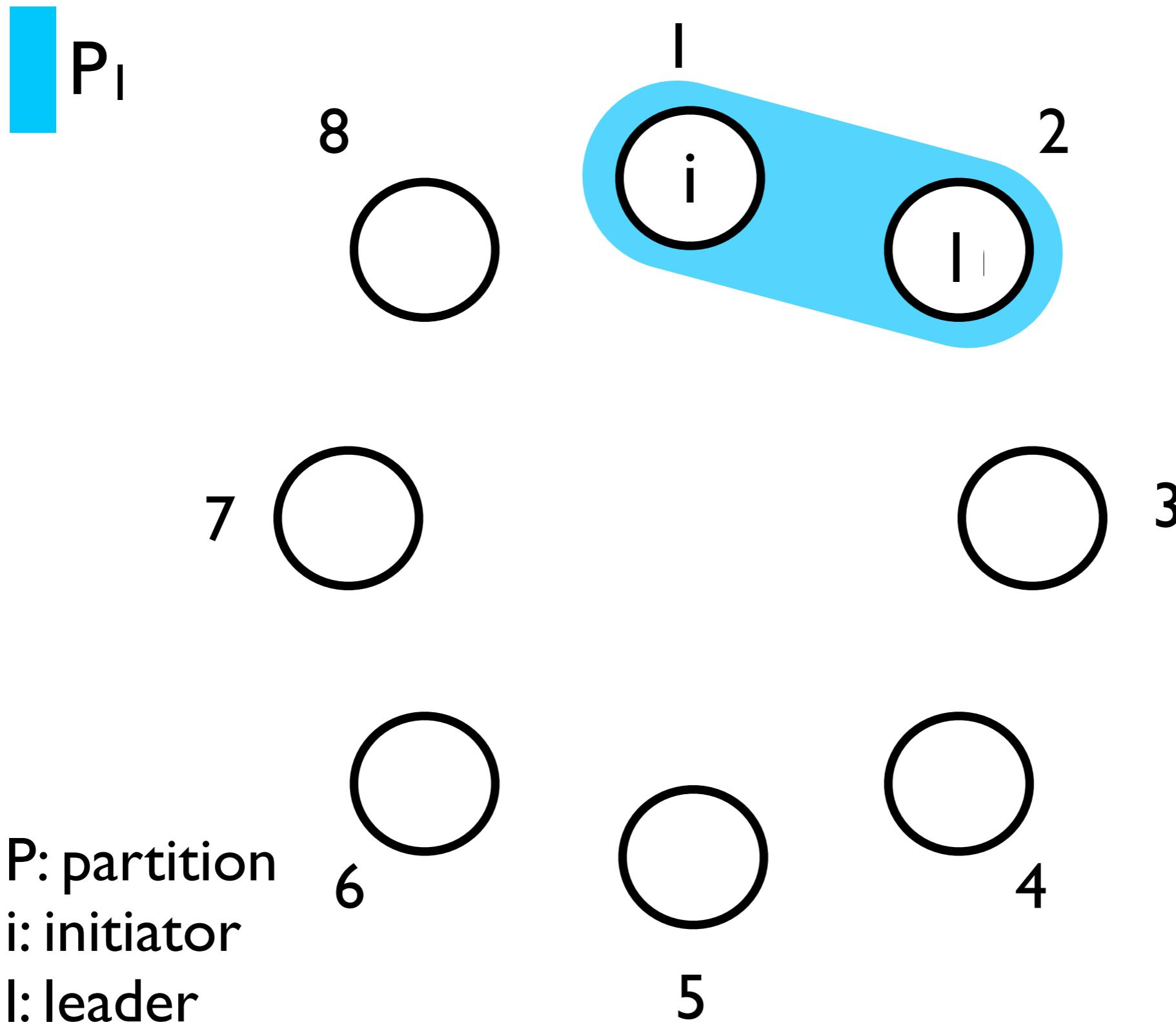
DVMS Proposal - Main Algorithm



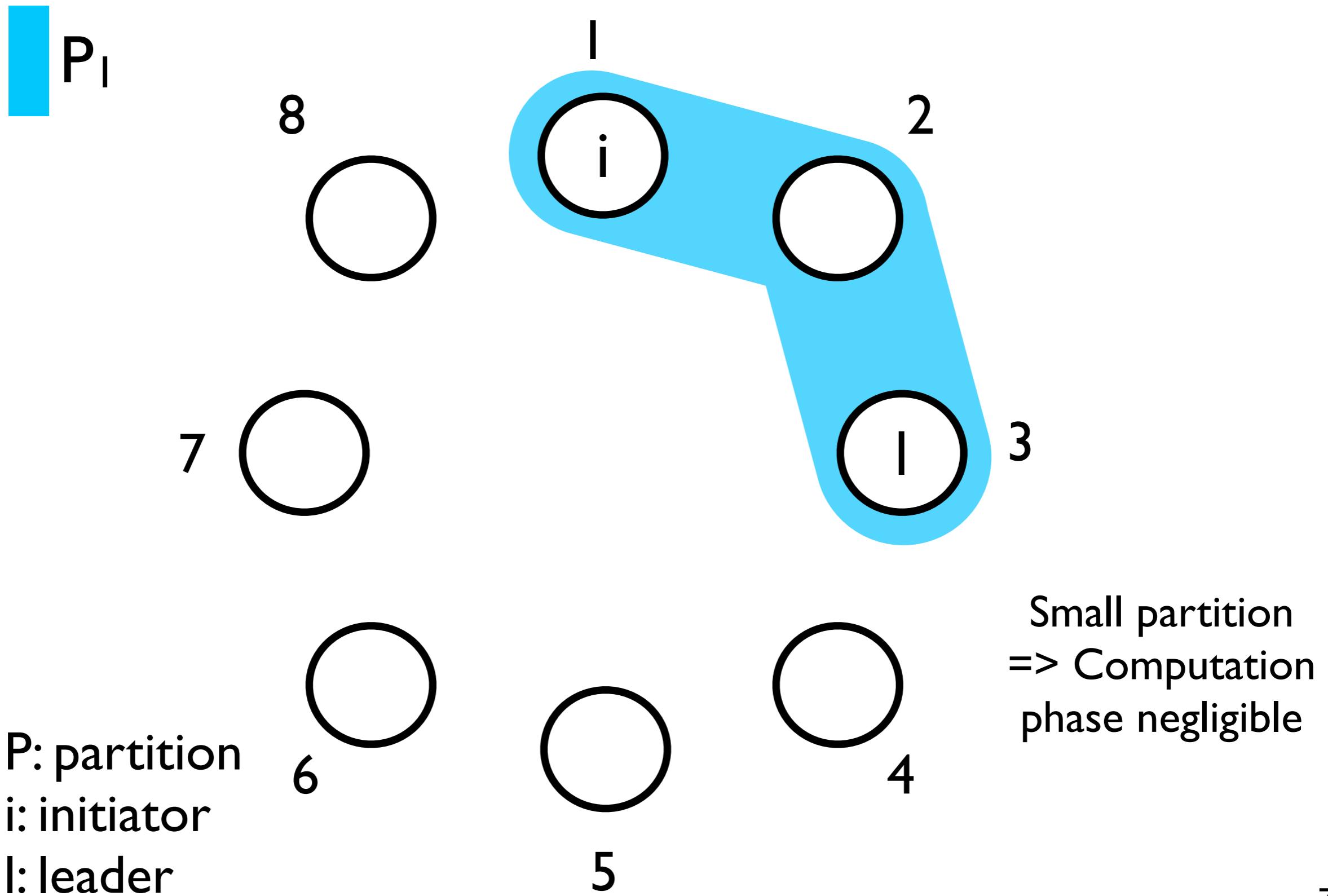
DVMS Proposal - Main Algorithm



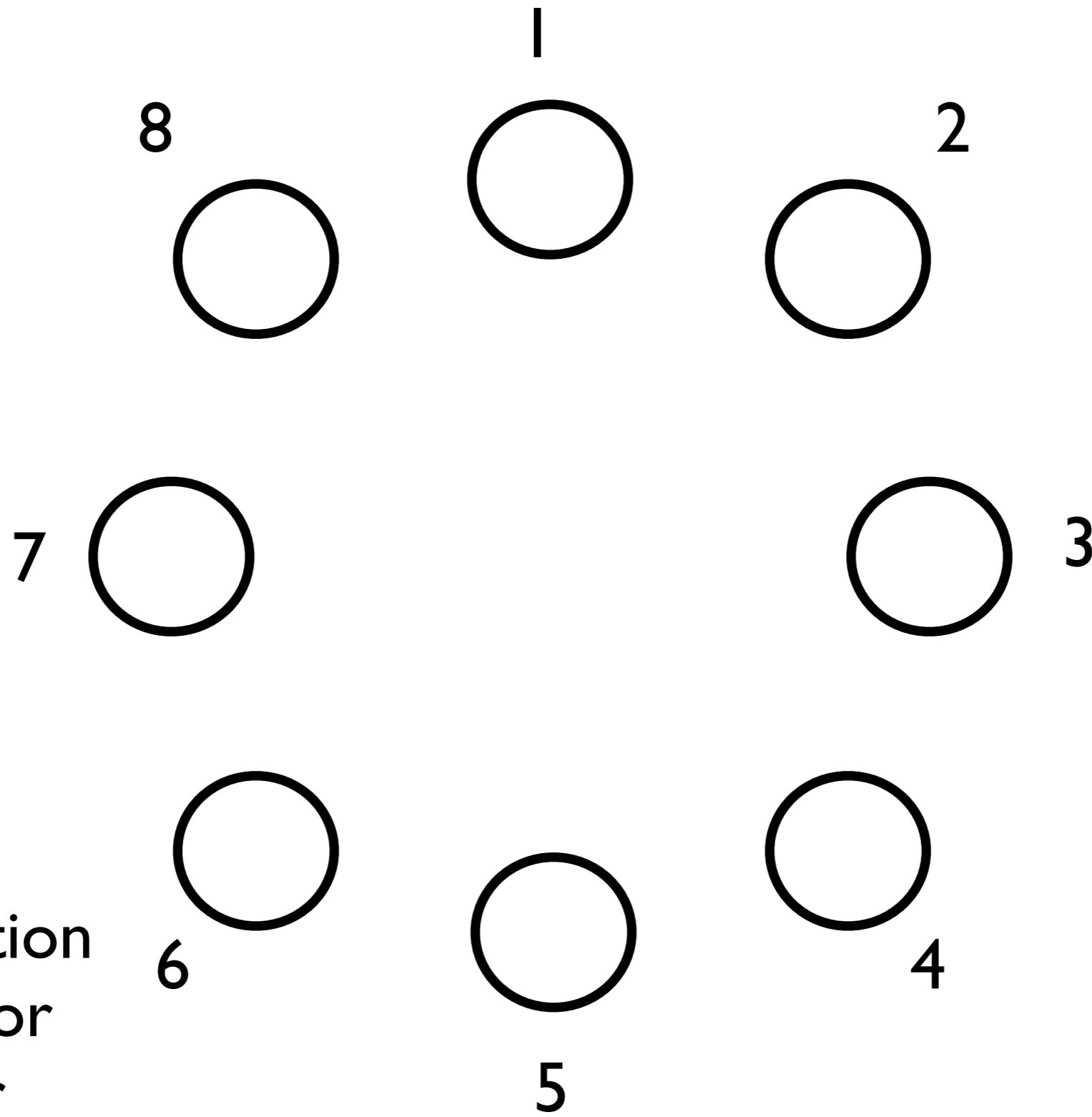
DVMS Proposal - Main Algorithm



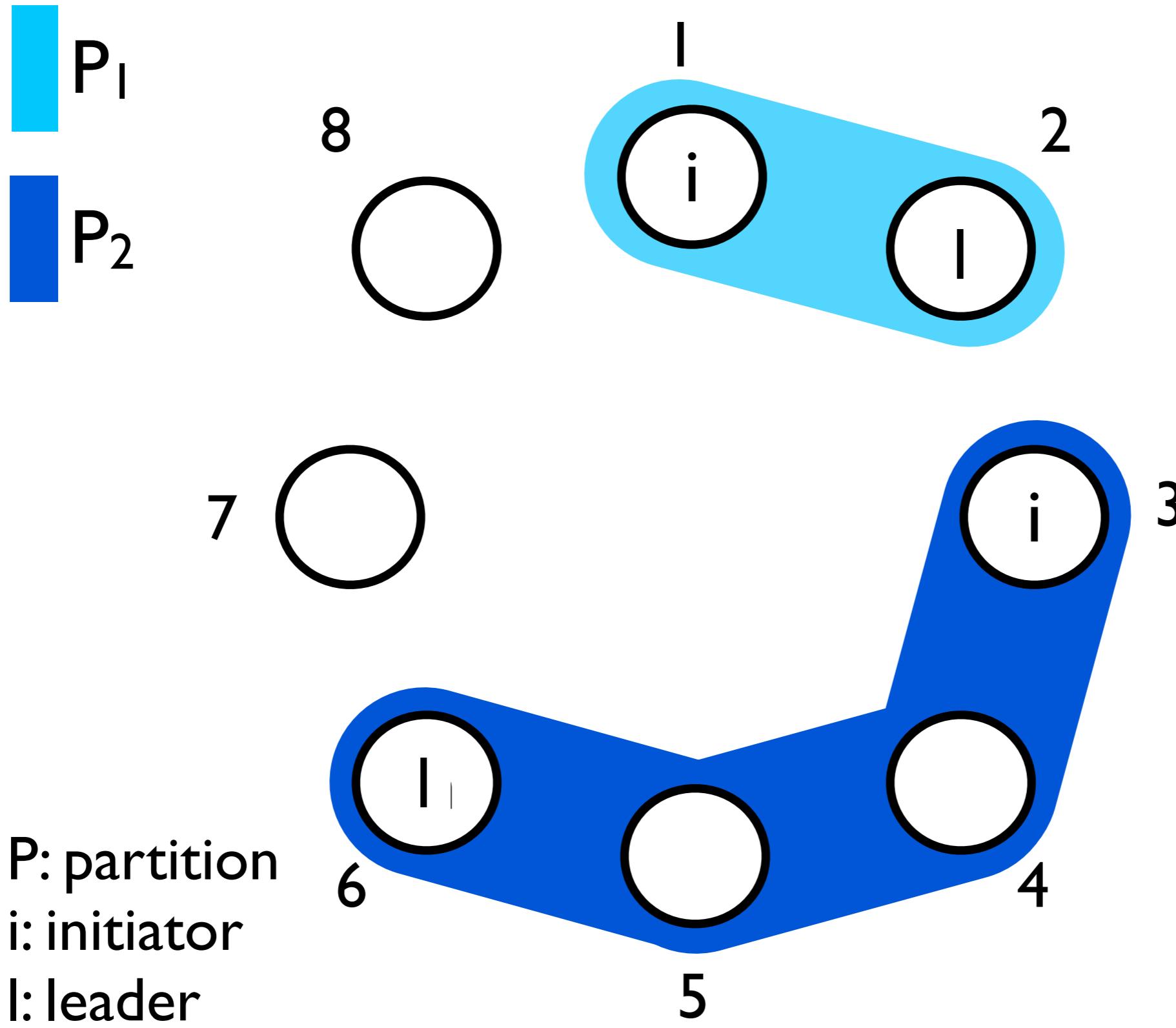
DVMS Proposal - Main Algorithm



DVMS Proposal - Main Algorithm



DVMS Proposal - Shortcuts

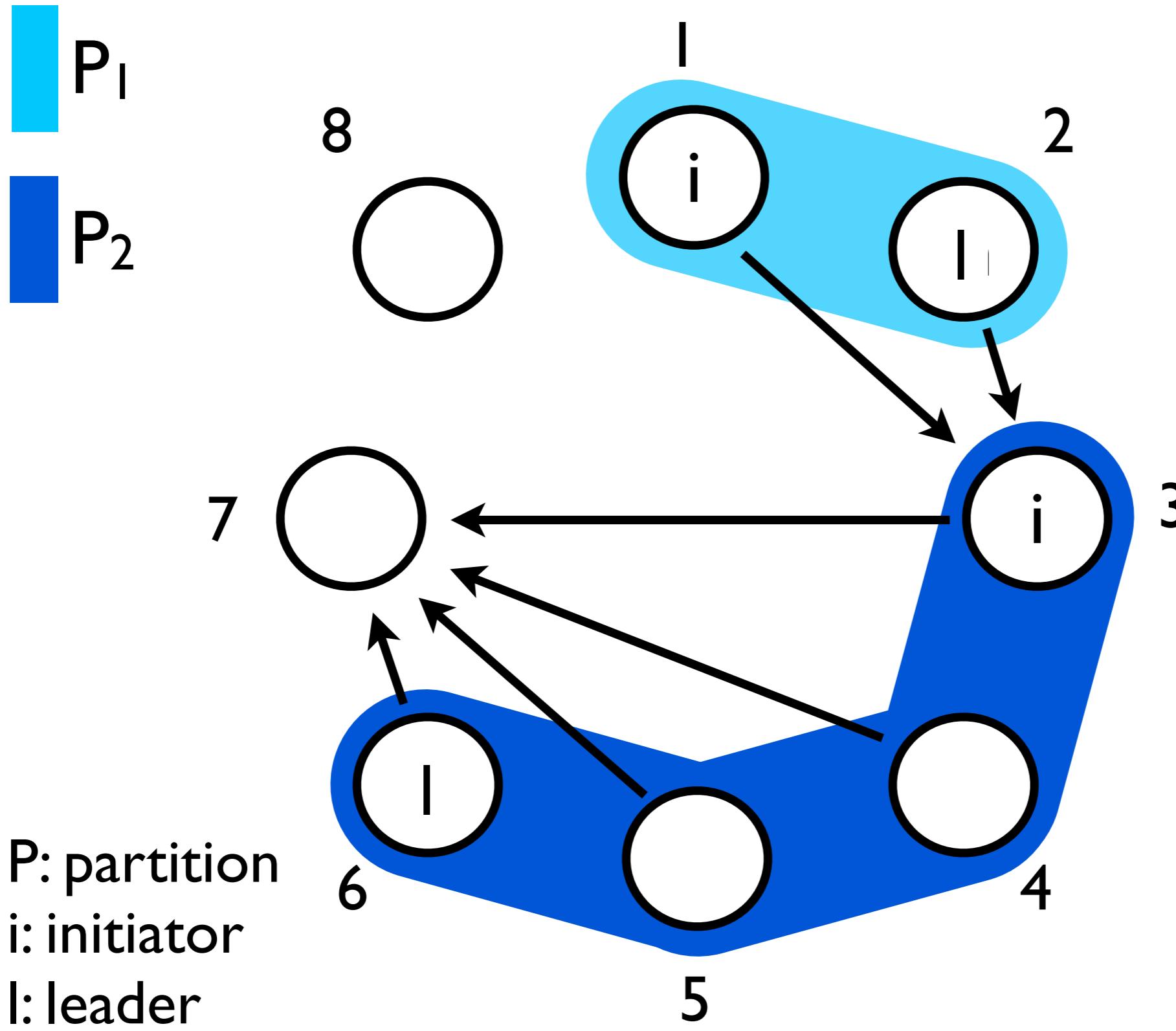


P: partition

i: initiator

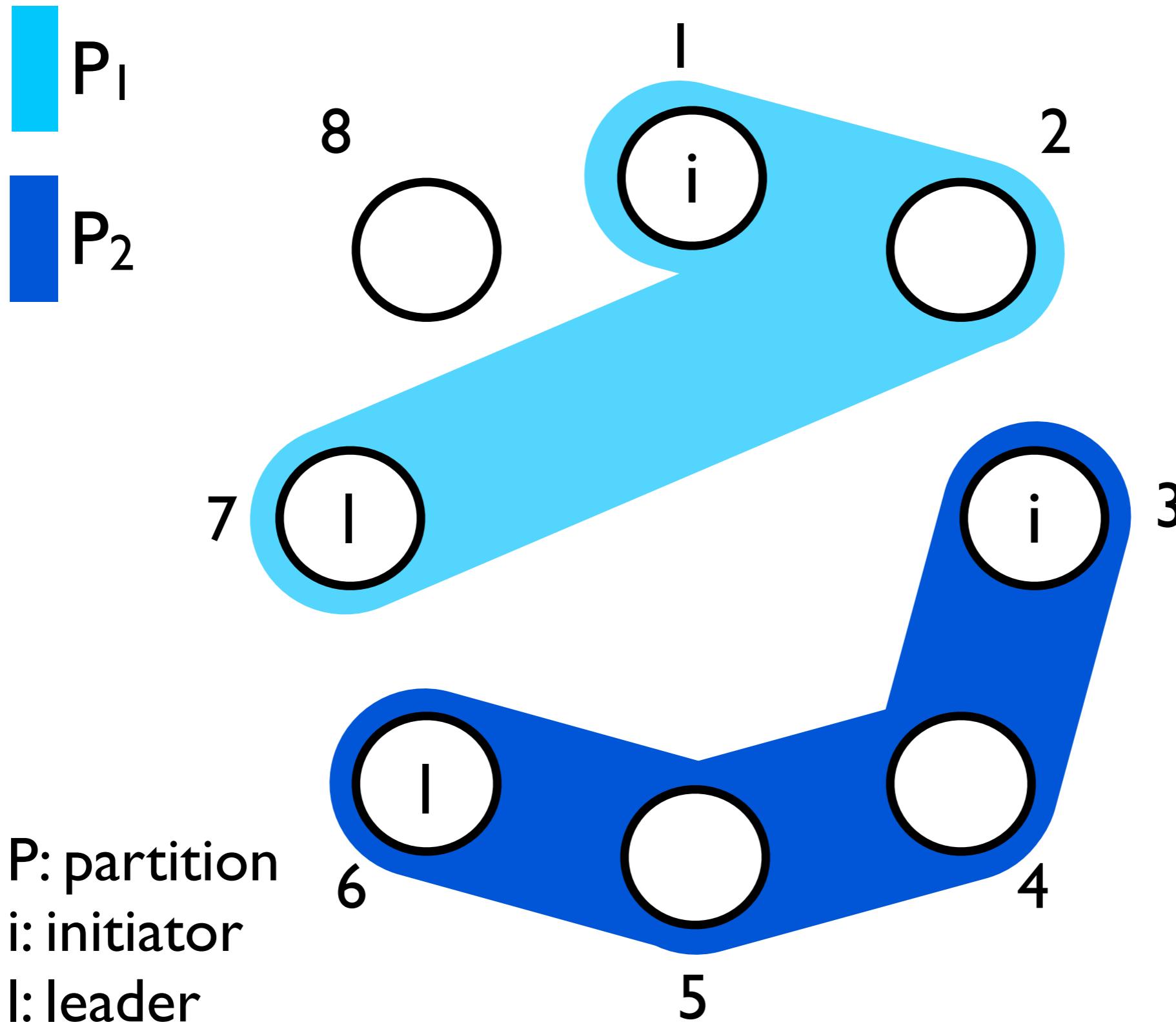
l: leader

DVMS Proposal - Shortcuts



P: partition
i: initiator
l: leader

DVMS Proposal - Shortcuts



P: partition

i: initiator

l: leader

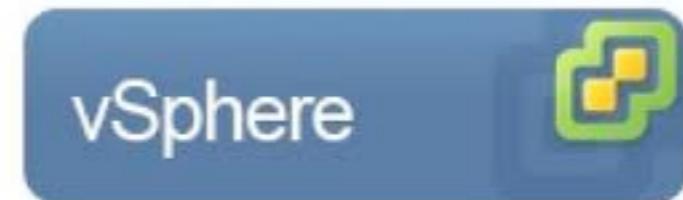
Operating IaaS Platforms

A state of the art (in 5 slides)



Operating IaaS - CloudKit (*Cloud OS ?*)

- Proprietary proposals
- vCloud/vSphere (vmware) 60%



ESXi

- XenServer/Xen Cloud platform (20%)

Xen

CITRIX® XenServer

- Microsoft System Center VM (20%)

Hypervisors agnostic



Credits: <http://www.v-index.com>
Virtualization Industry Quarterly Survey

Operating IaaS - CloudKit (Cloud OS ?)

- Academic proposals

Nimbus (Freeman and Keahey, University of Chicago)

Based on GT4 and the Globus Virtual Workspace Service

Target: cloud for science

Tutorials and documentation in “grid space”



Open Nebula (Montero & Llorente, DSA-Research at UCM)

Support for the Xen, KVM and VMware

Access to Amazon EC2 (cloud bursting)

Probably, the most deployed in EU (2012)



Eucalyptus (Wolsky, University of Santa Barbara)

Web services based implementation of elastic/utility/cloud computing infrastructure



Operating IaaS - CloudKit (*Cloud OS ?*)

- Community proposals

OpenStack

Supported by several industrials

Successor of OpenNebula for the core of the Ubuntu cloud proposal



CloudStack

Supported by CITRIX

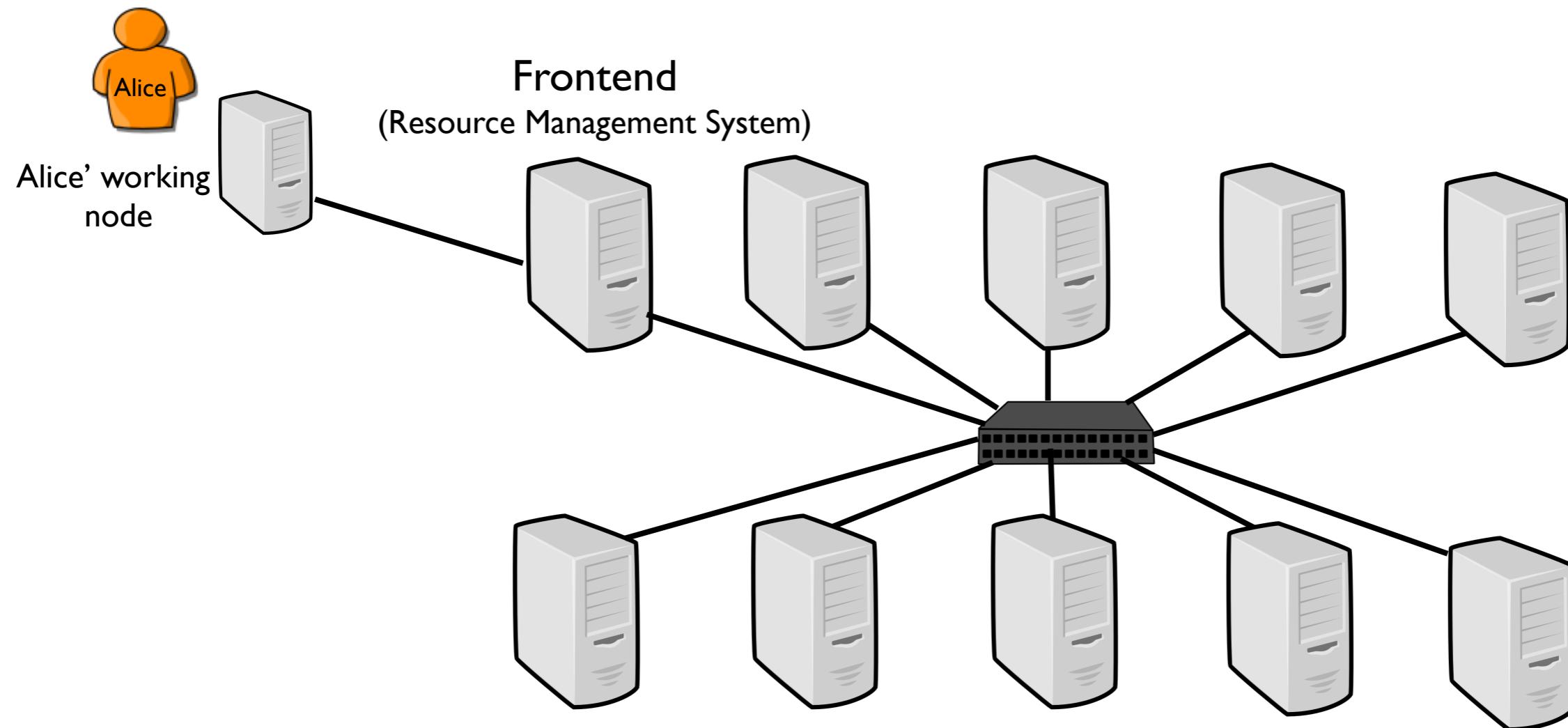
Full JAVA implementation

Apache project



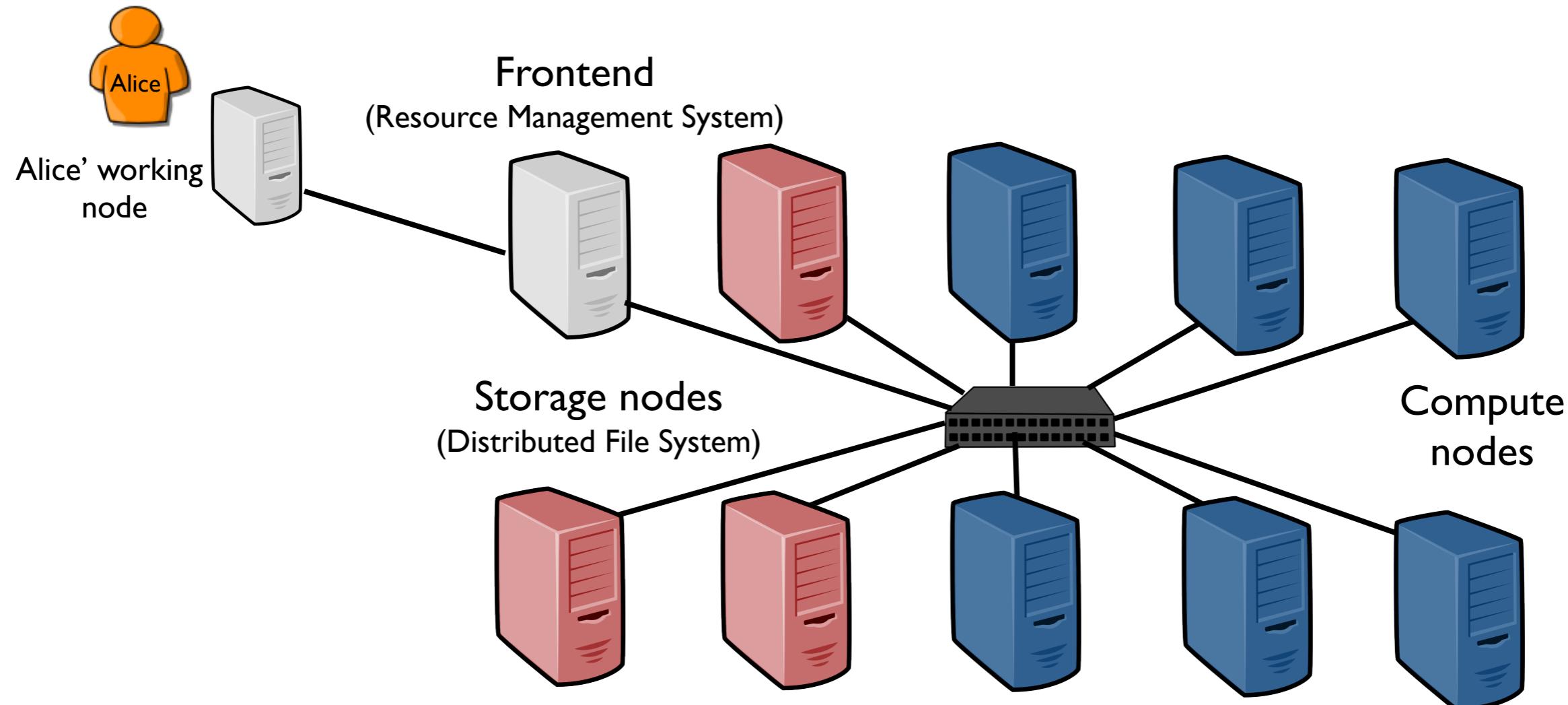
Utility Computing - Successive Generations

- Network of Workstations 1990 / 20**xx**



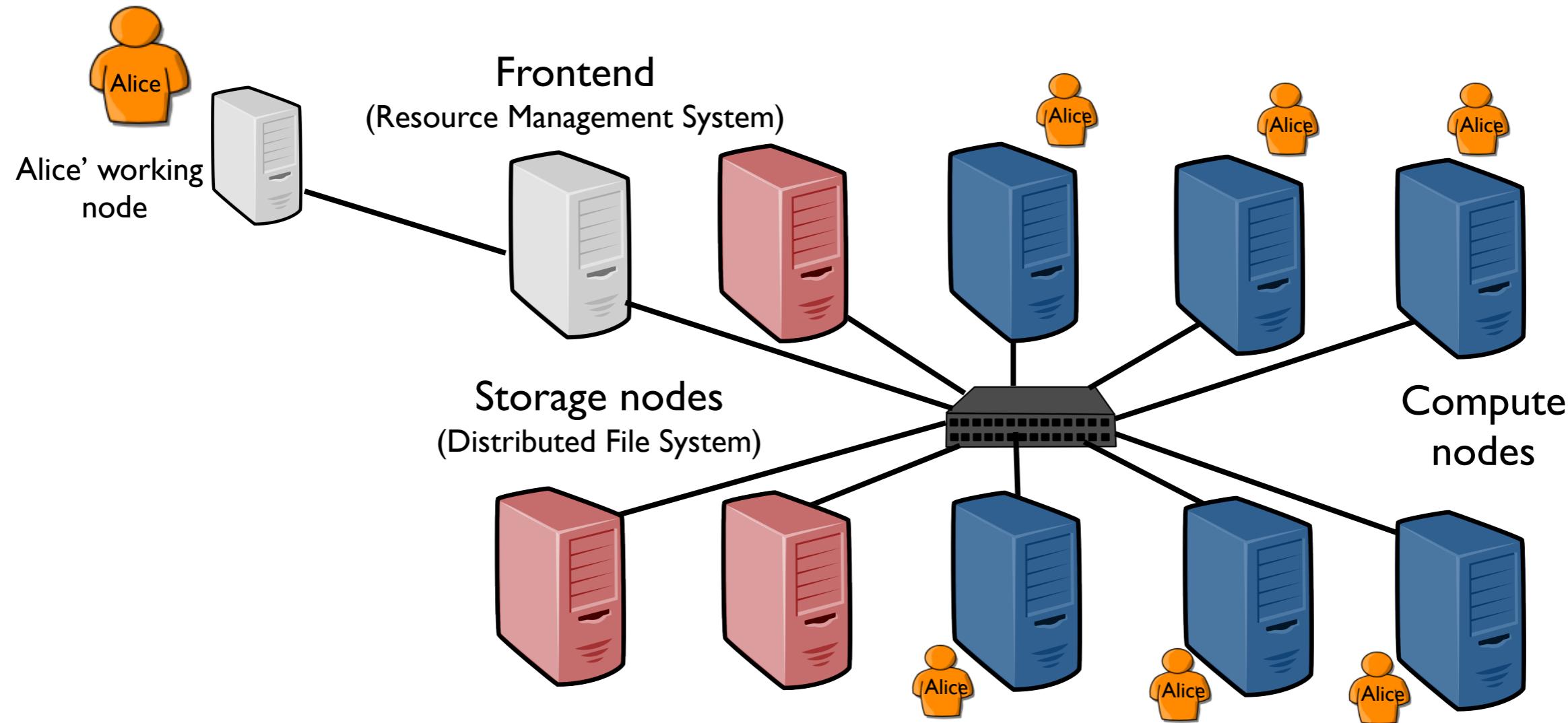
Utility Computing - Successive Generations

- Network of Workstations 1990 / 20XX



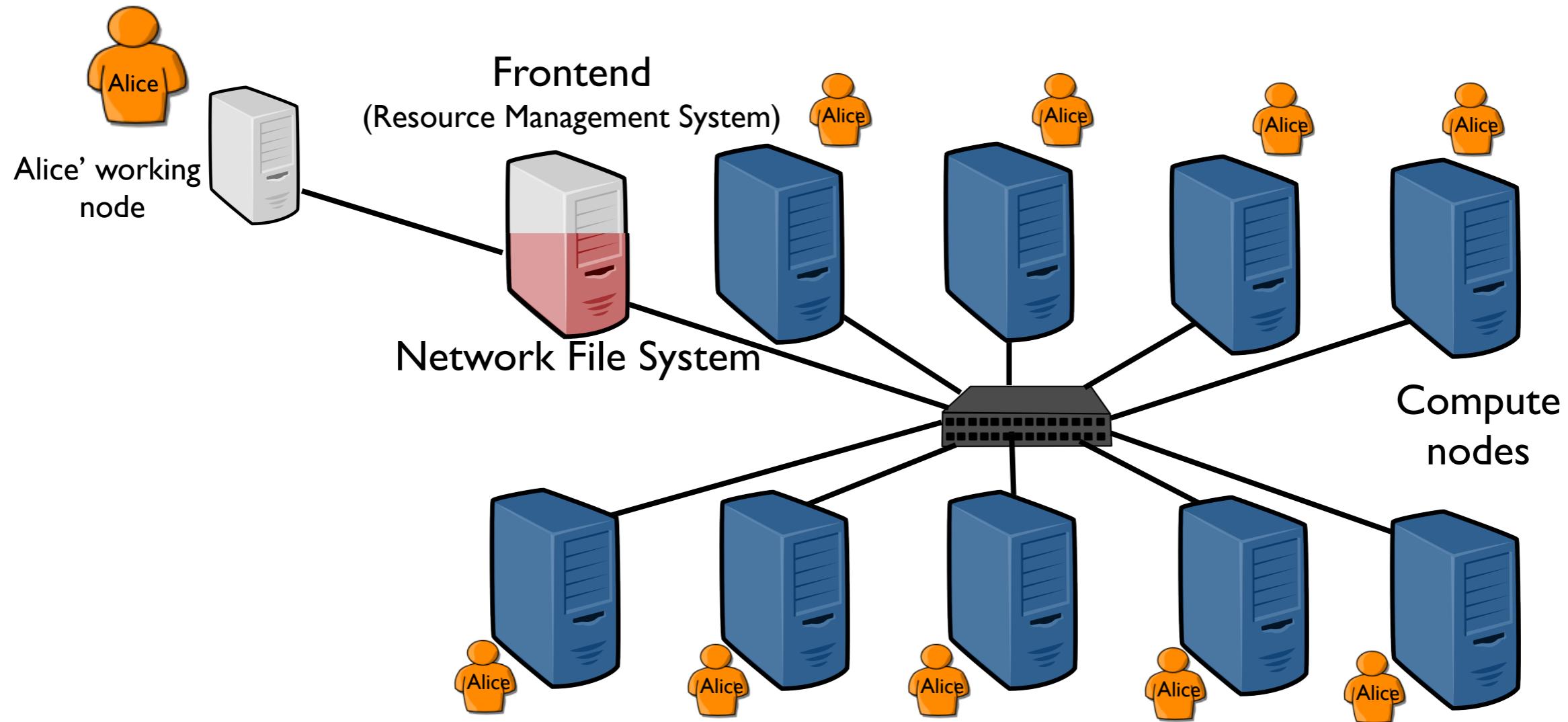
Utility Computing - Successive Generations

- Network of Workstations 1990 / 20XX



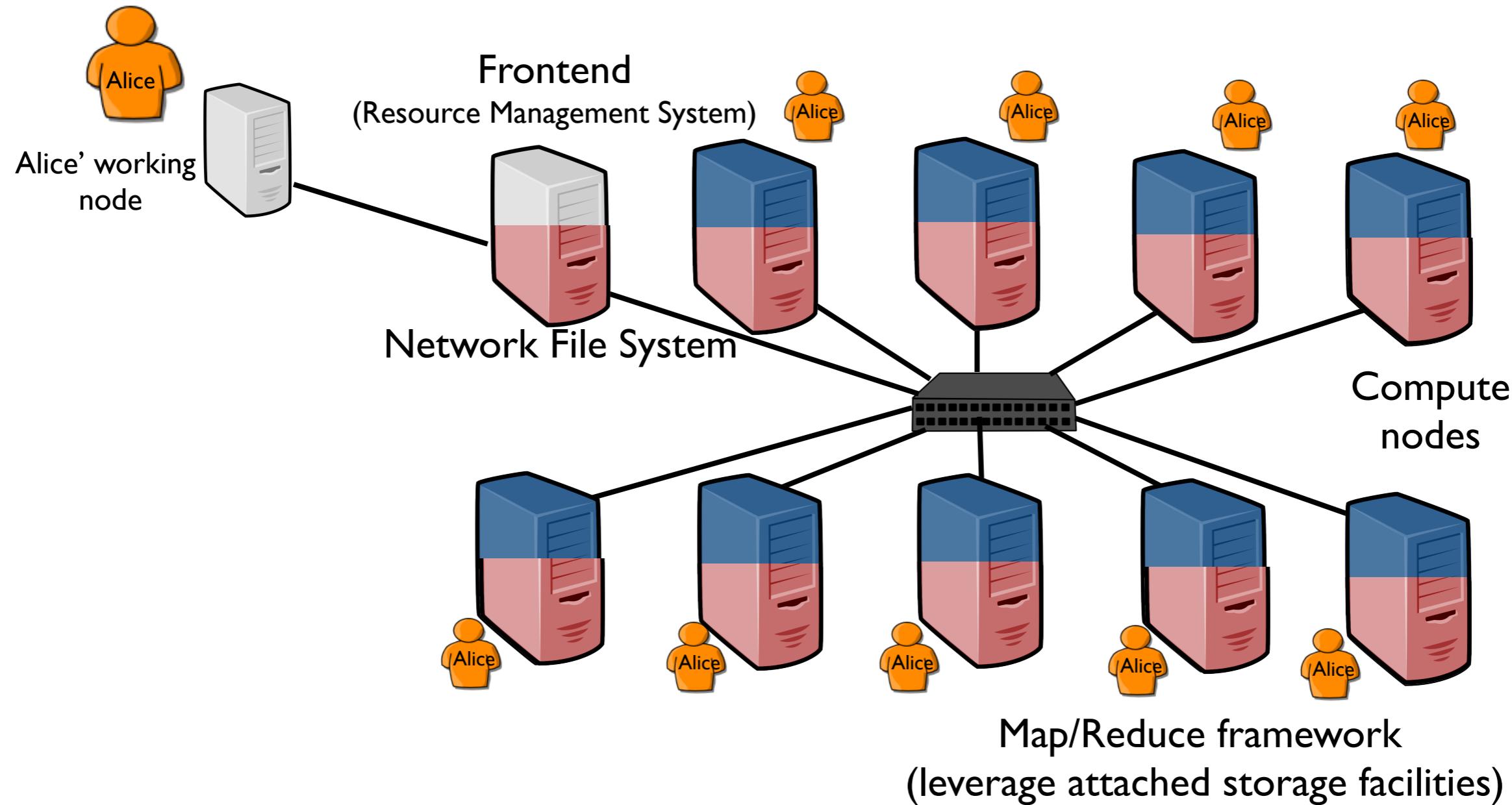
Utility Computing - Successive Generations

- Network of Workstations 1990 / 20XX



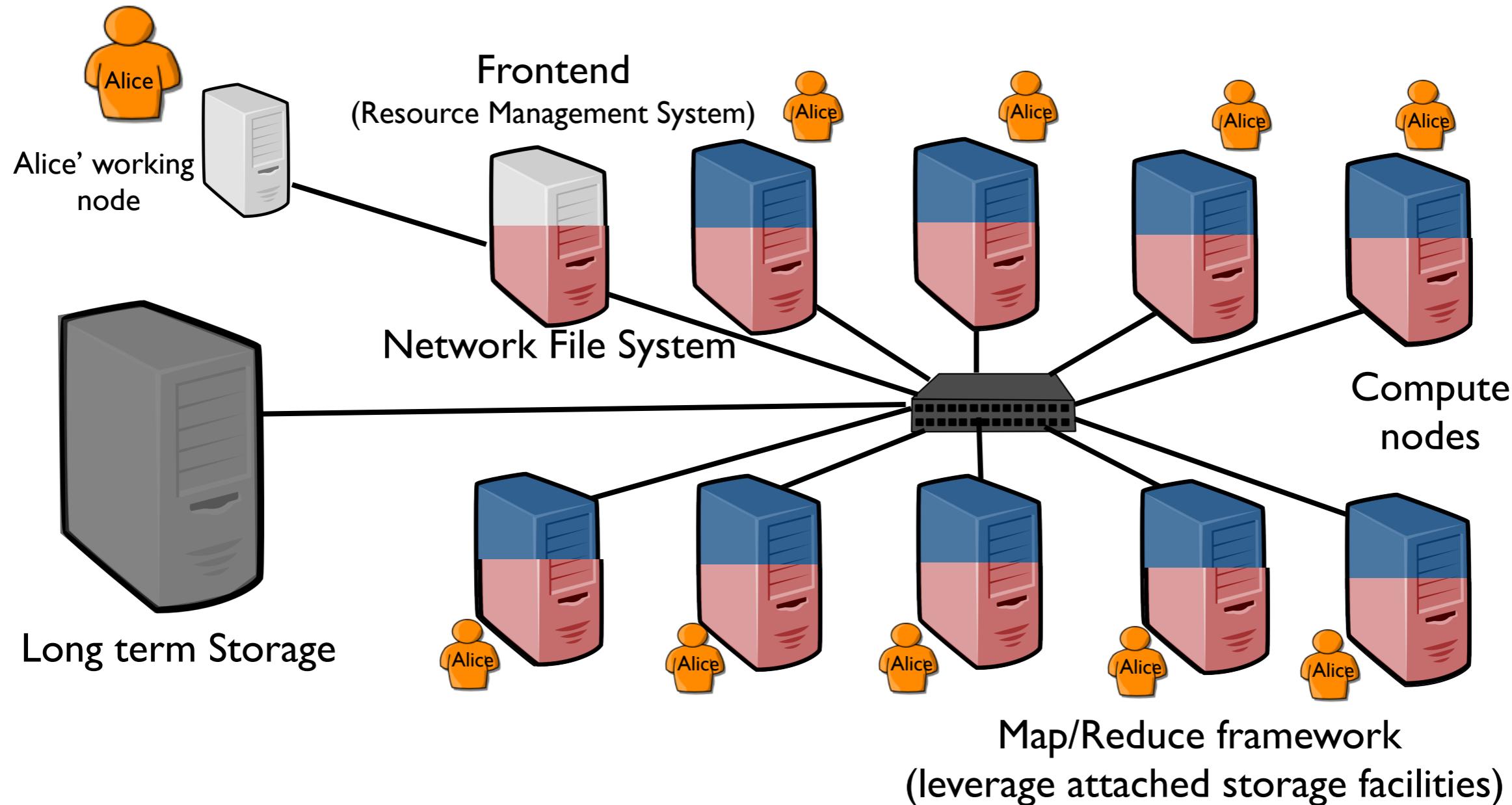
Utility Computing - Successive Generations

- Network of Workstations 1990 / 20XX



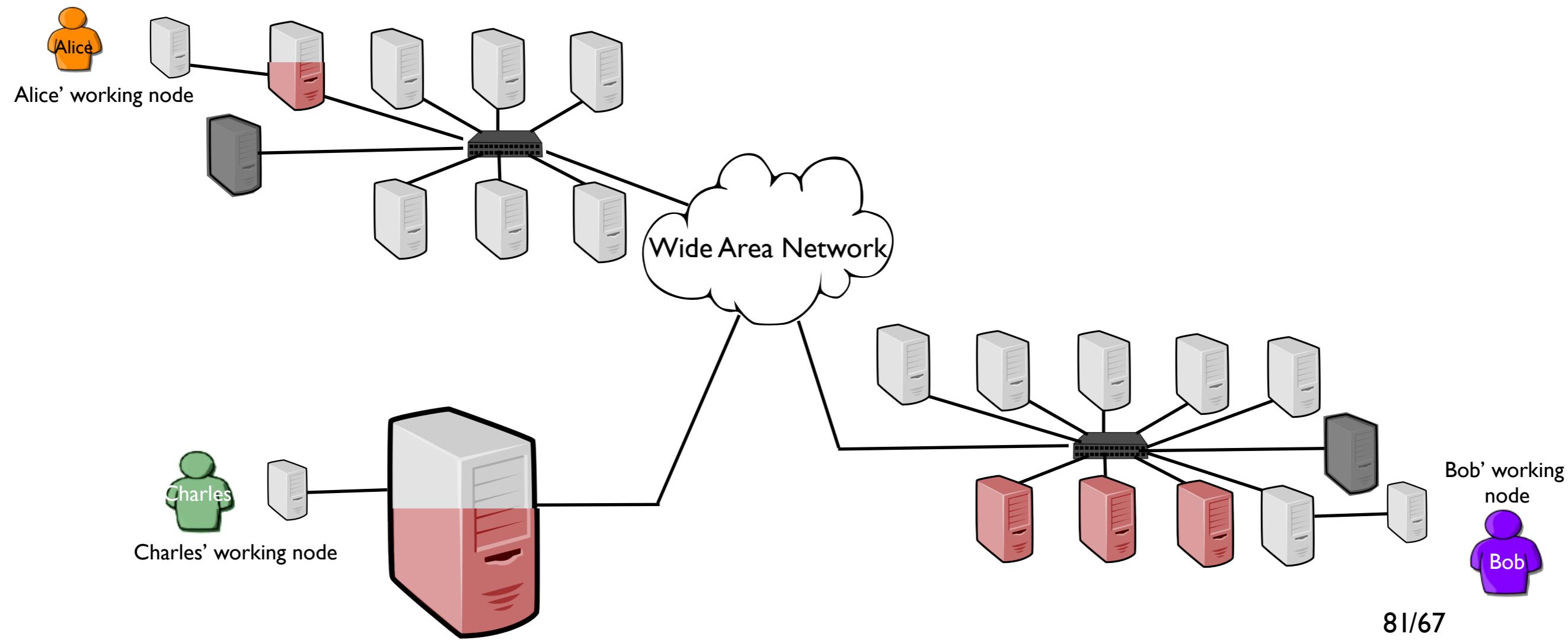
Utility Computing - Successive Generations

- Network of Workstations 1990 / 20XX



Utility Computing - Successive Generations

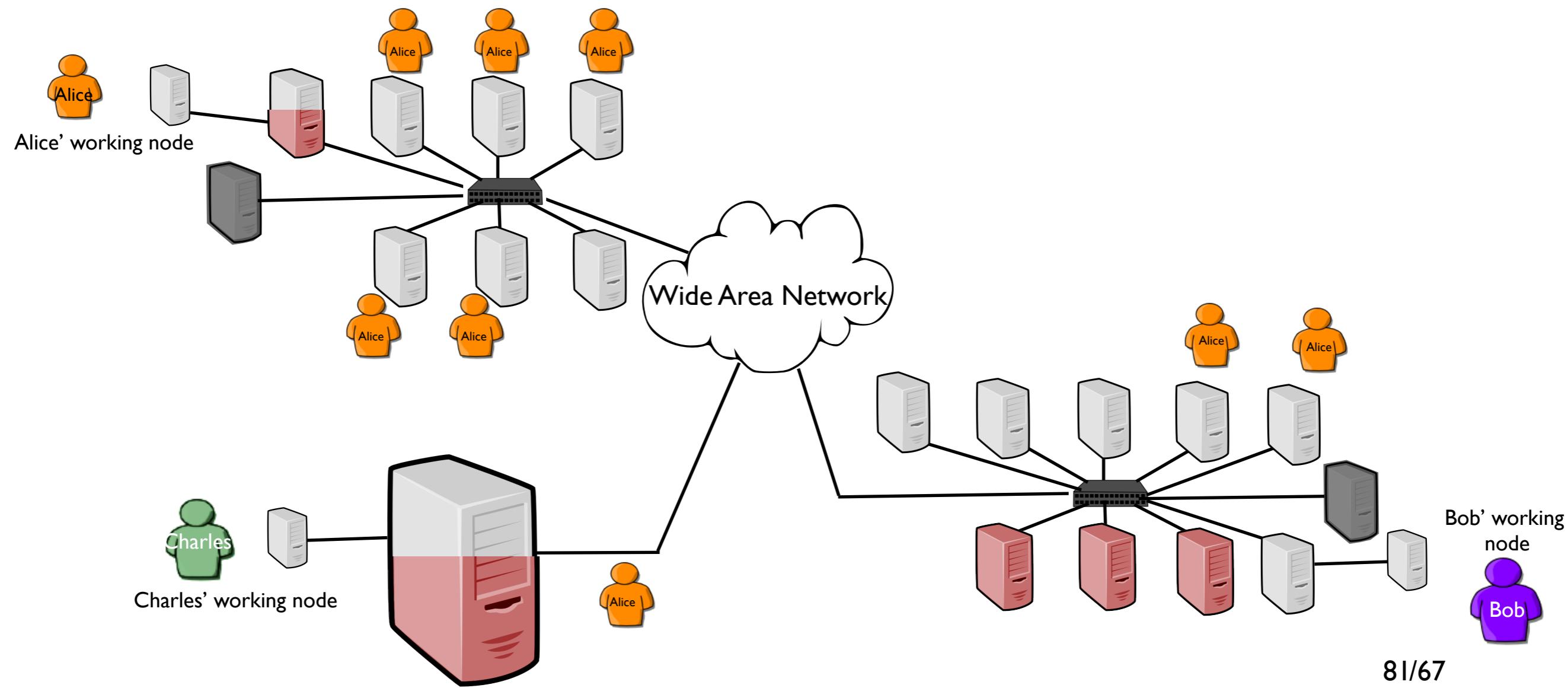
- Network of Workstations 1990 / 20~~xx~~
- The Grid 1997 / 20~~xx~~



81/67

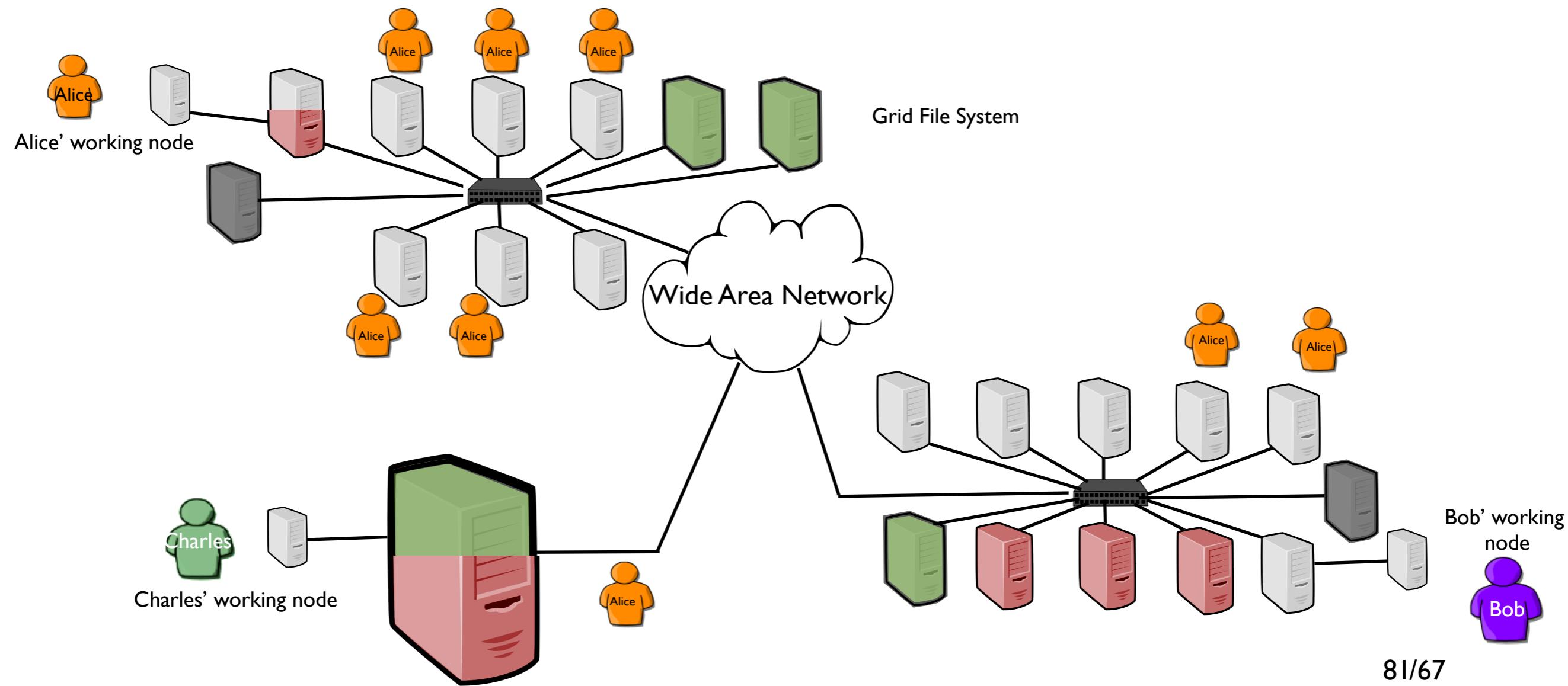
Utility Computing - Successive Generations

- Network of Workstations 1990 / 20~~xx~~
- The Grid 1997 / 20~~xx~~



Utility Computing - Successive Generations

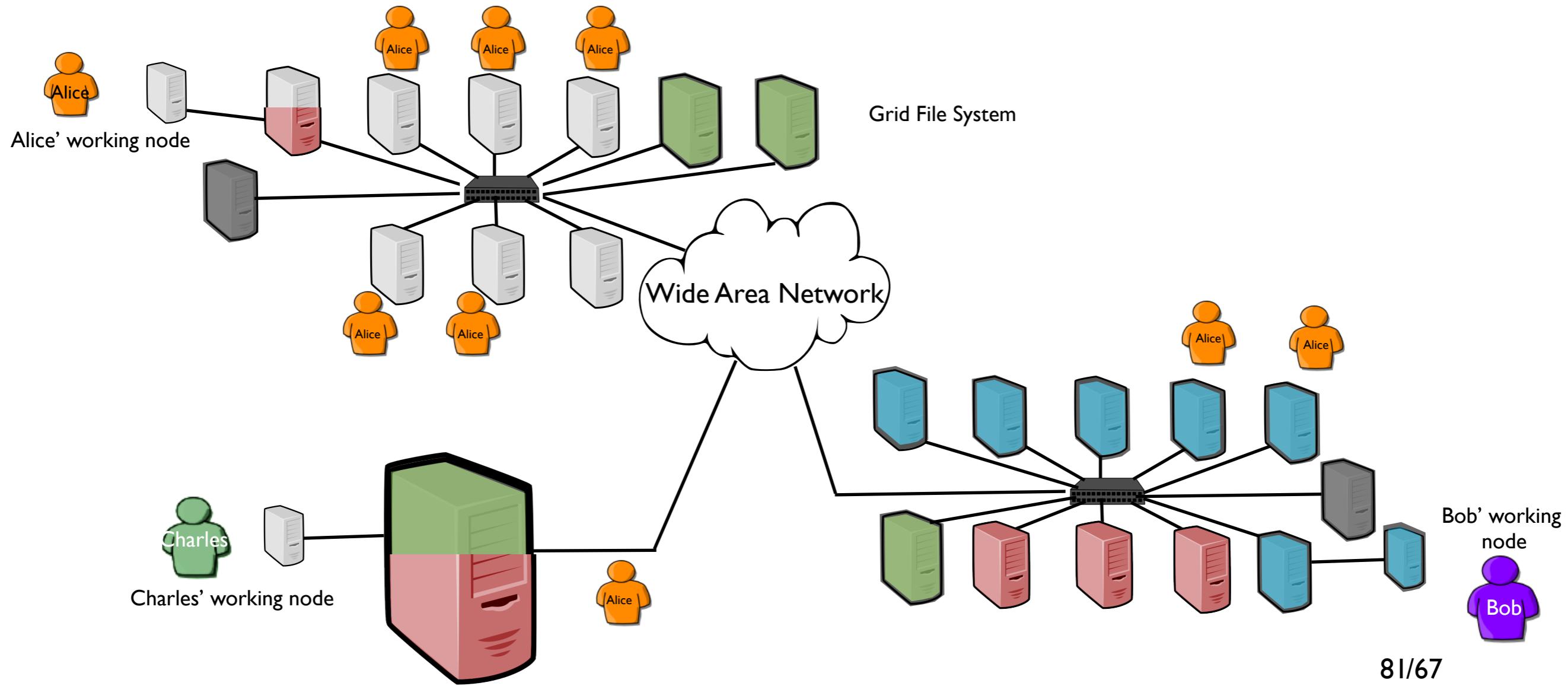
- Network of Workstations 1990 / 20~~xx~~
- The Grid 1997 / 20~~xx~~



81/67

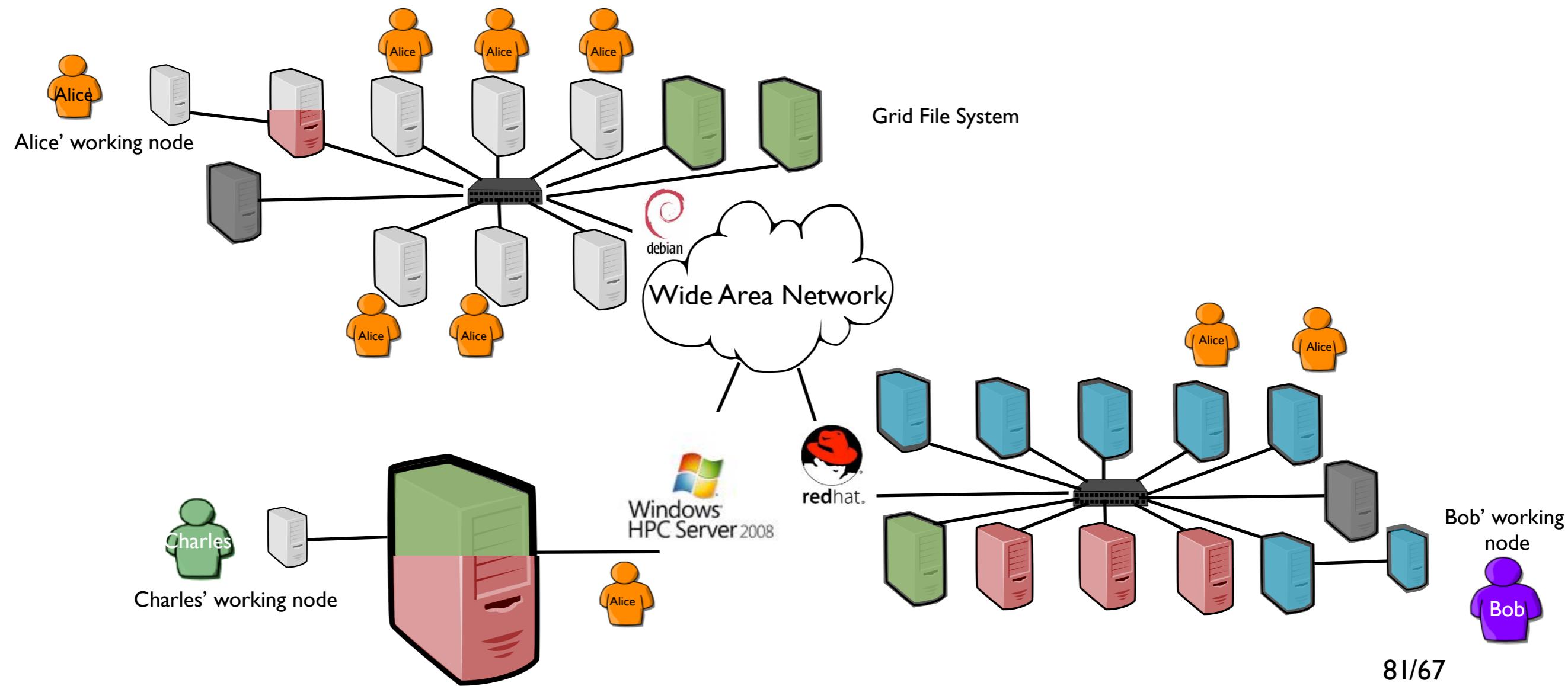
Utility Computing - Successive Generations

- Network of Workstations 1990 / 20xx
 - The Grid 1997 / 201x



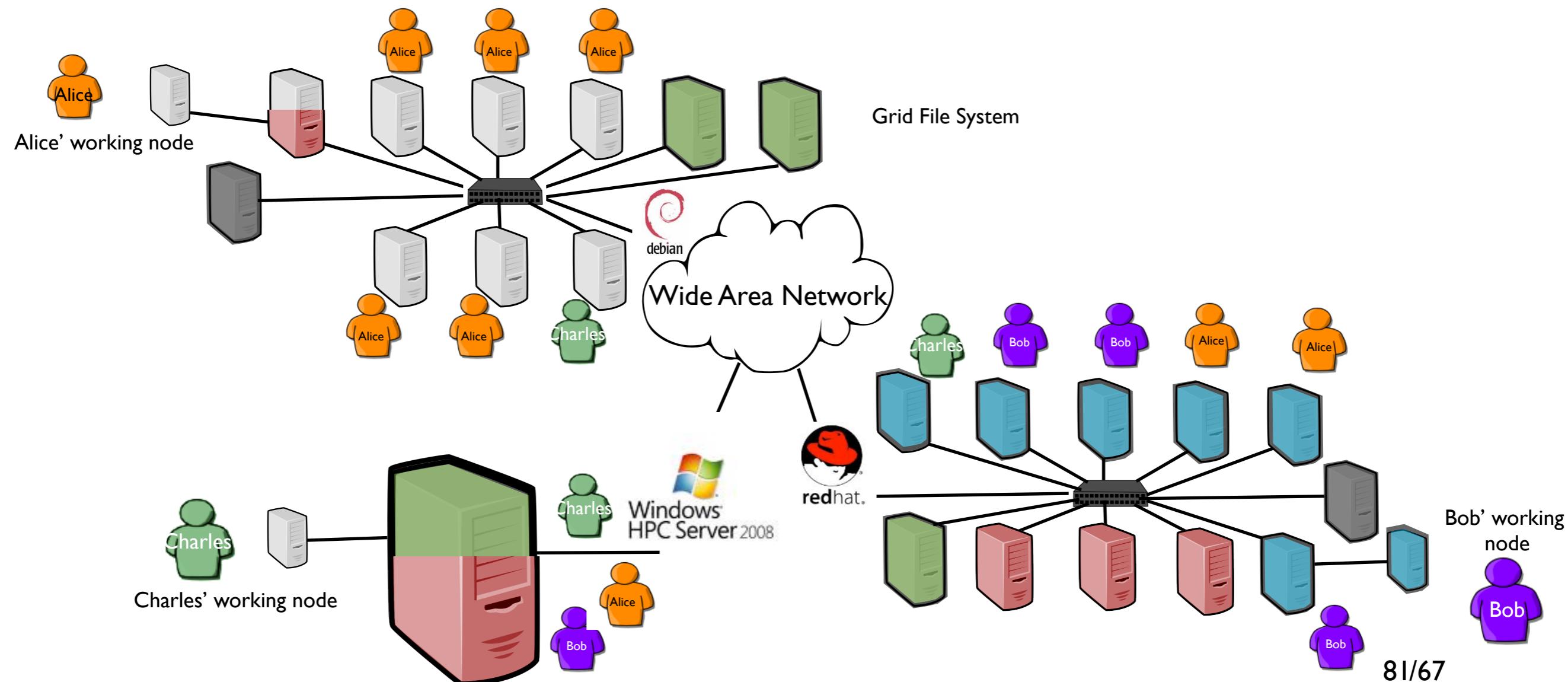
Utility Computing - Successive Generations

- Network of Workstations 1990 / 20~~xx~~
- The Grid 1997 / 20~~xx~~



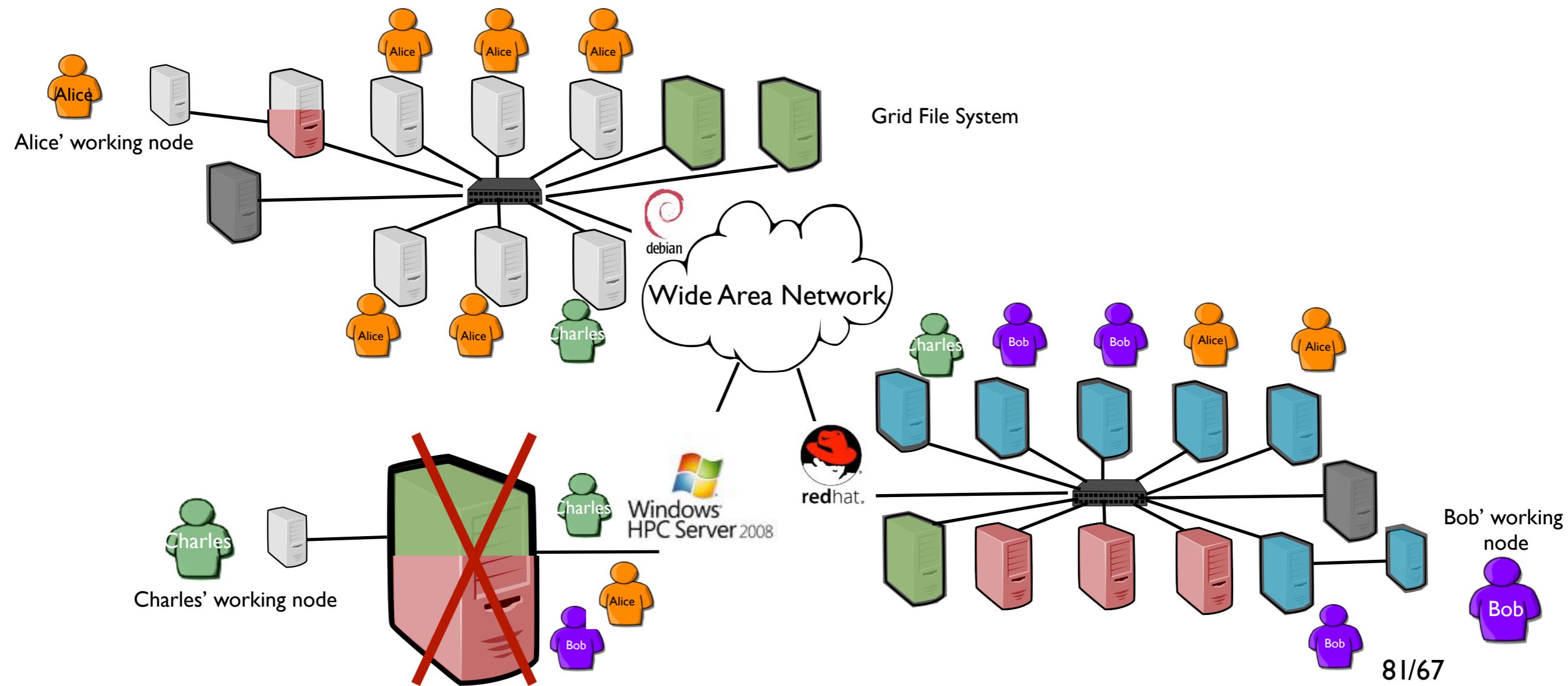
Utility Computing - Successive Generations

- Network of Workstations 1990 / 20~~XX~~
- The Grid 1997 / 20~~I~~~~X~~



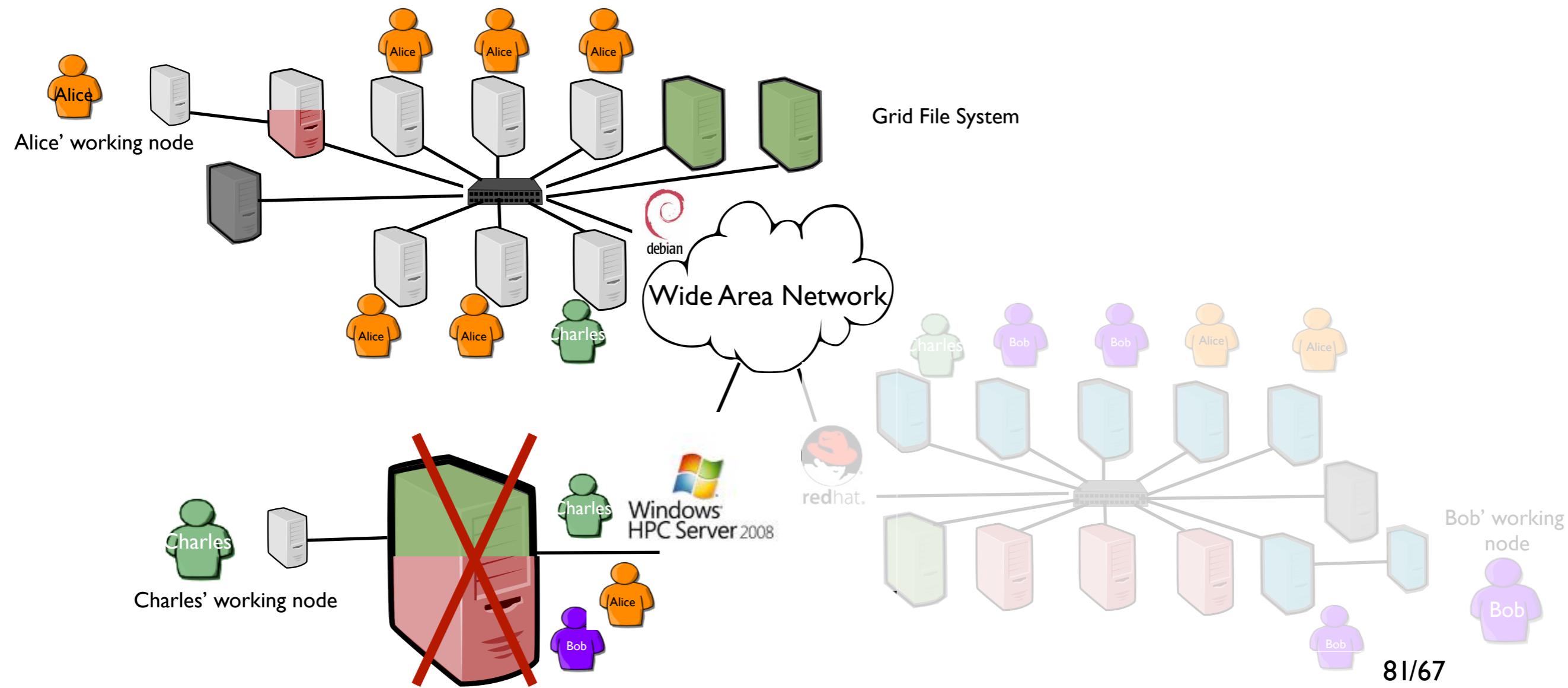
Utility Computing - Successive Generations

- Network of Workstations 1990 / 20~~XX~~
- The Grid 1997 / 20~~I~~~~X~~



Utility Computing - Successive Generations

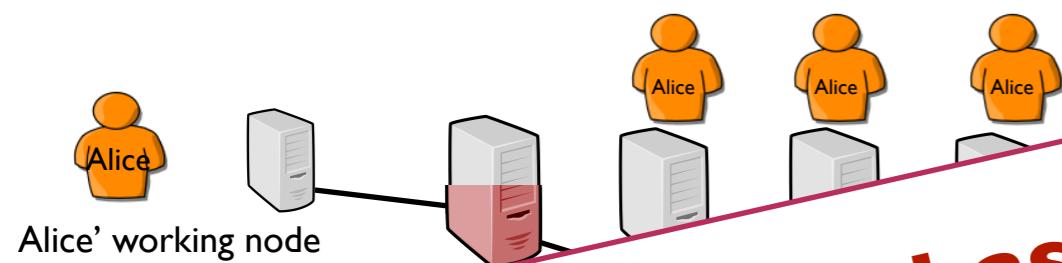
- Network of Workstations 1990 / 20~~XX~~
- The Grid 1997 / 20~~I~~~~X~~



81/67

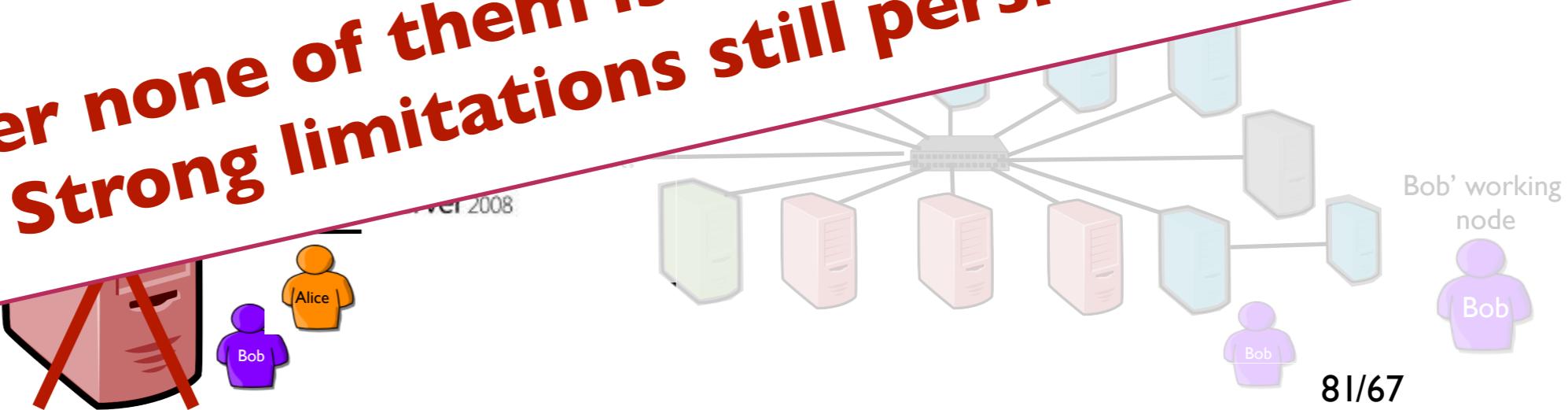
Utility Computing - Successive Generations

- Network of Workstations 1990 / 20~~xx~~
- The Grid 1997 / 20~~xx~~



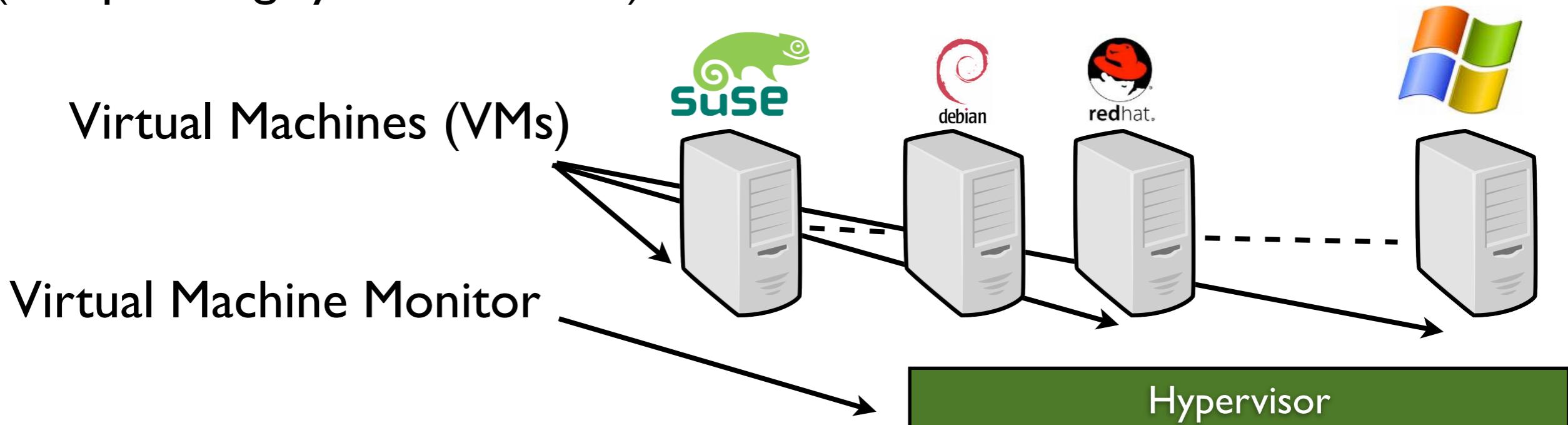
A lot of progress has been done since the 90's and several proposals partially addressed these concerns.

However none of them is mature enough and Strong limitations still persist !



Here Comes System Virtualization

- One to multiple OSes on a physical node thanks to a hypervisor (an operating system of OSes)



“A **virtual machine** (VM) provides a faithful implementation of a physical processor’s hardware running in a protected and isolated environment.

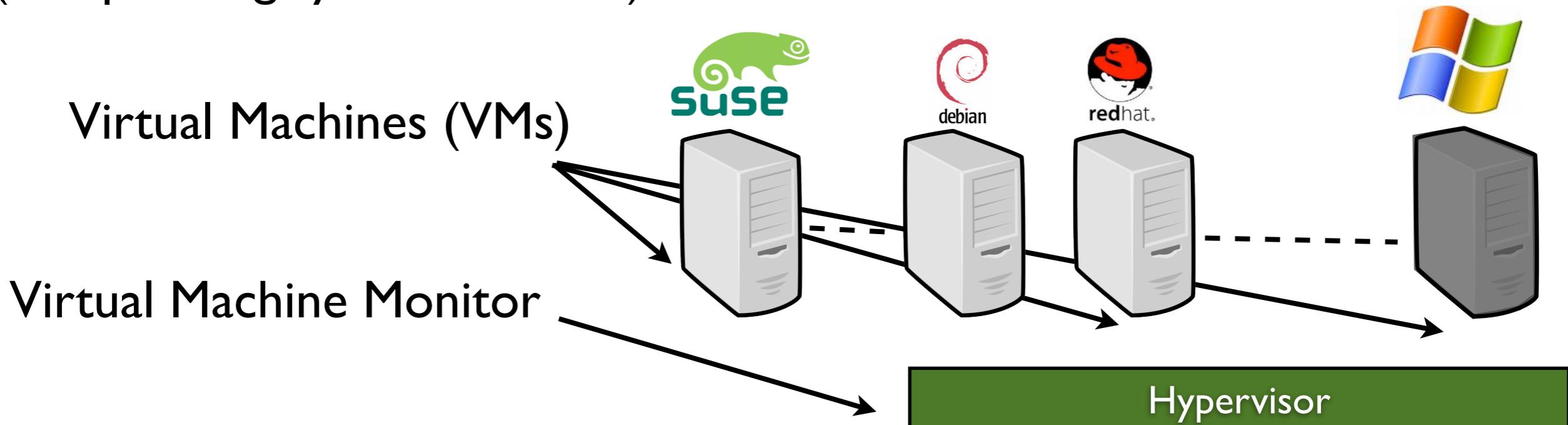
Virtual machines are created by a software layer called the **virtual machine monitor** (VMM) that runs as a privileged task on a physical processor.”



Physical Machine (PM)

Here Comes System Virtualization

- One to multiple OSes on a physical node thanks to a hypervisor (an operating system of OSes)



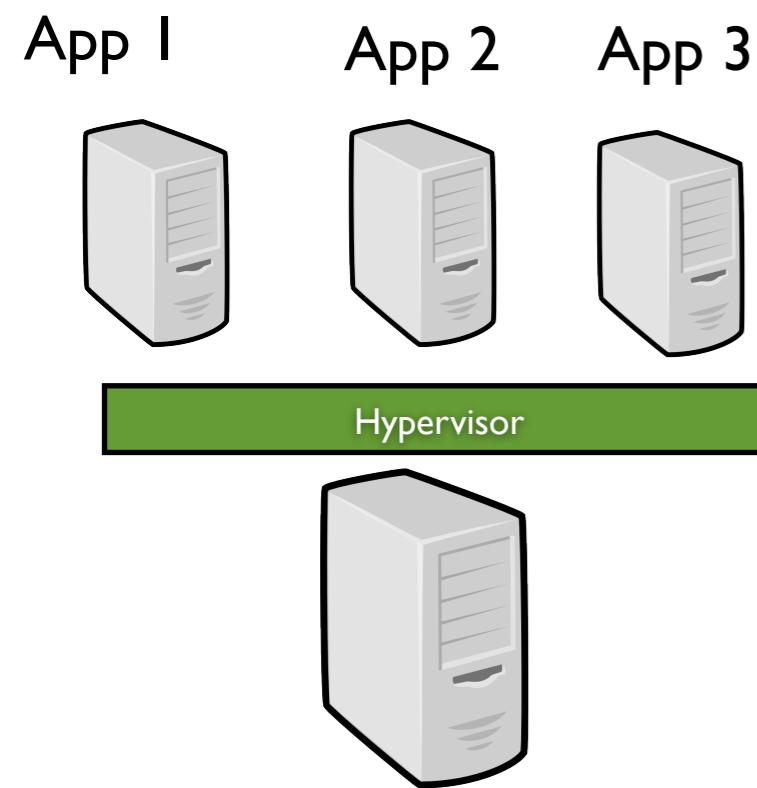
"A **virtual machine** (VM) provides a faithful implementation of a physical processor's hardware running in a protected and isolated environment.

Virtual machines are created by a software layer called the **virtual machine monitor** (VMM) that runs as a privileged task on a physical processor."



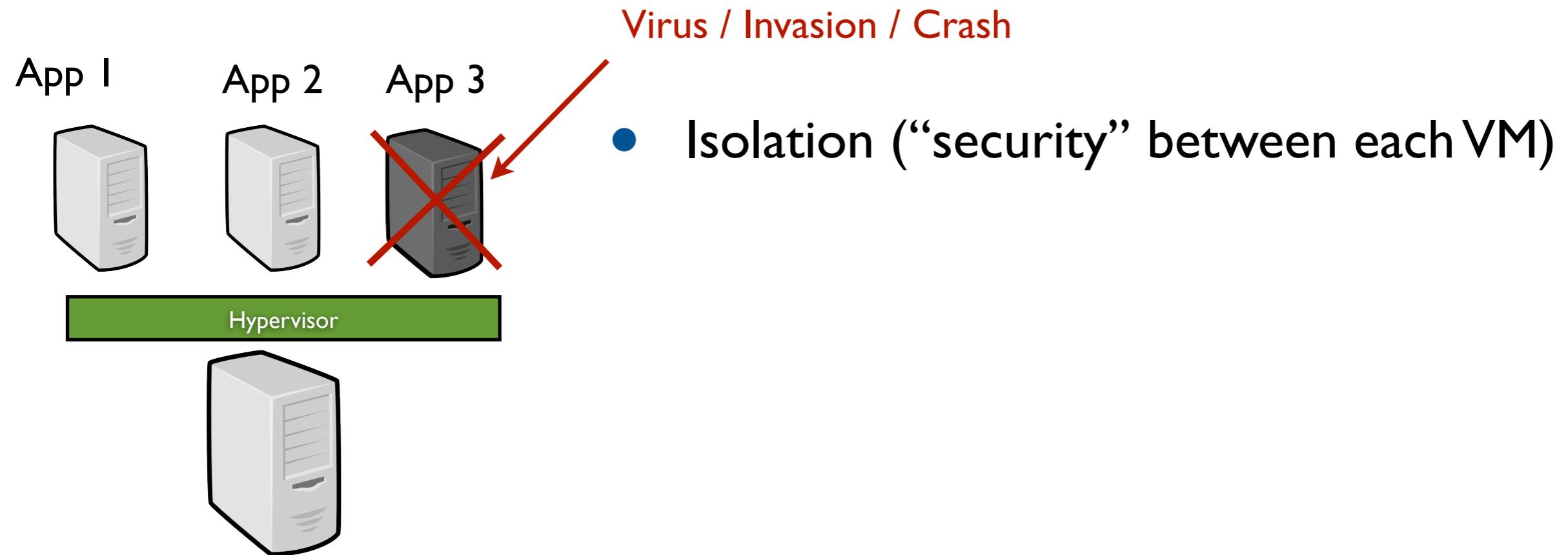
Physical Machine (PM)

VM Capabilities

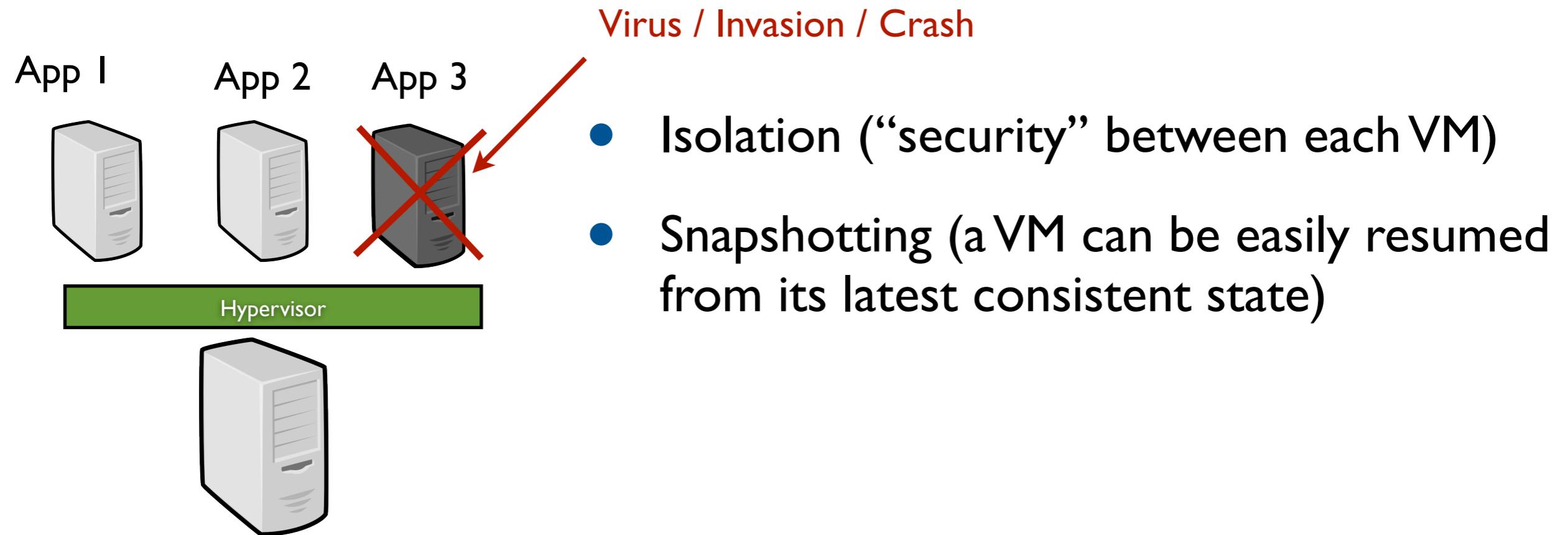


- Isolation (“security” between each VM)

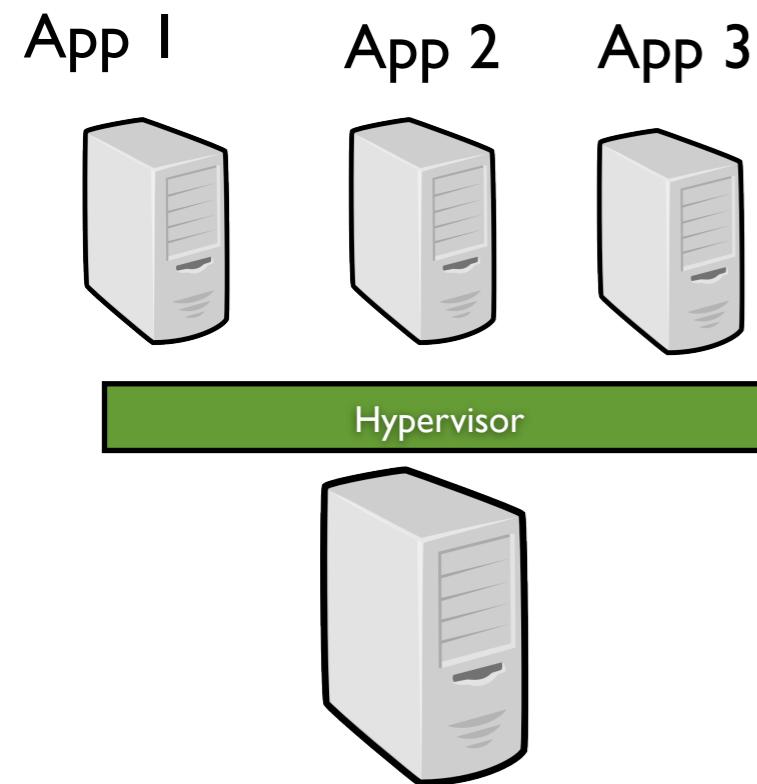
VM Capabilities



VM Capabilities

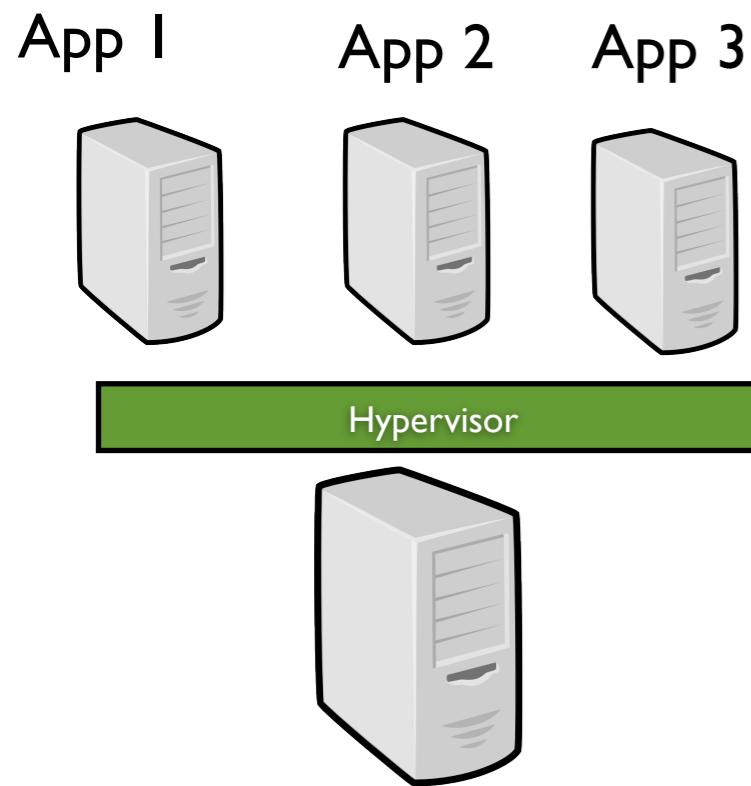


VM Capabilities



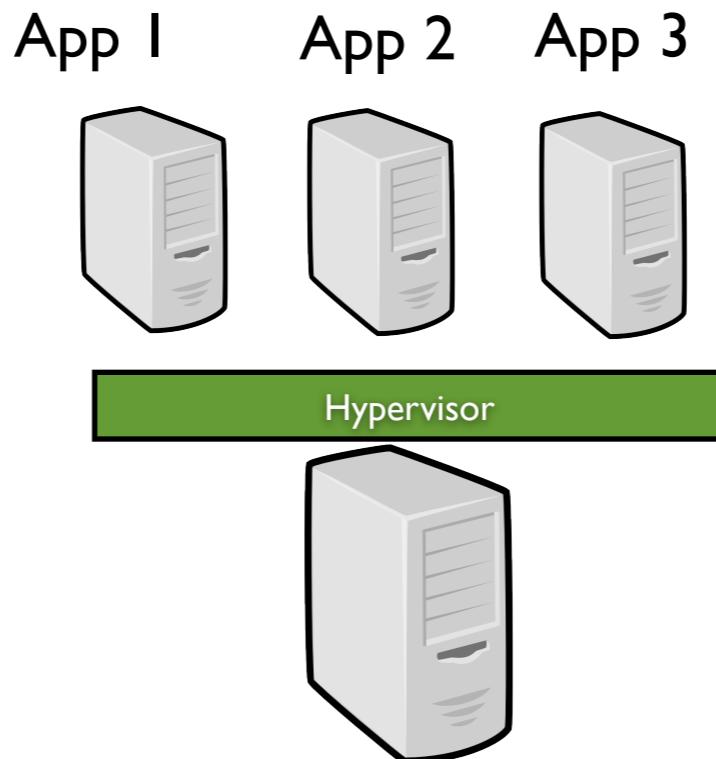
- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resumed from its latest consistent state)

VM Capabilities

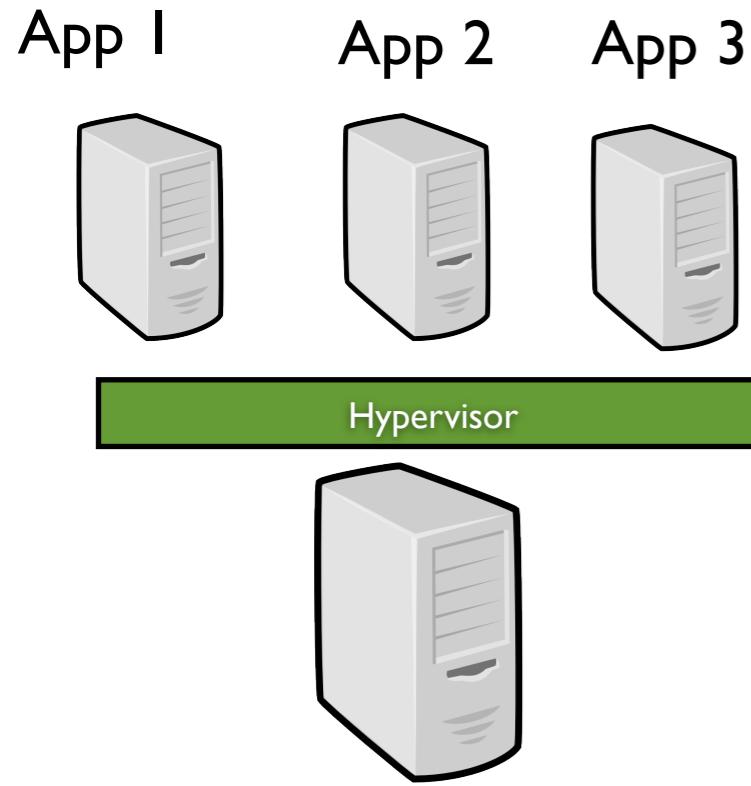


- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resumed from its latest consistent state)

- Suspend/Resume

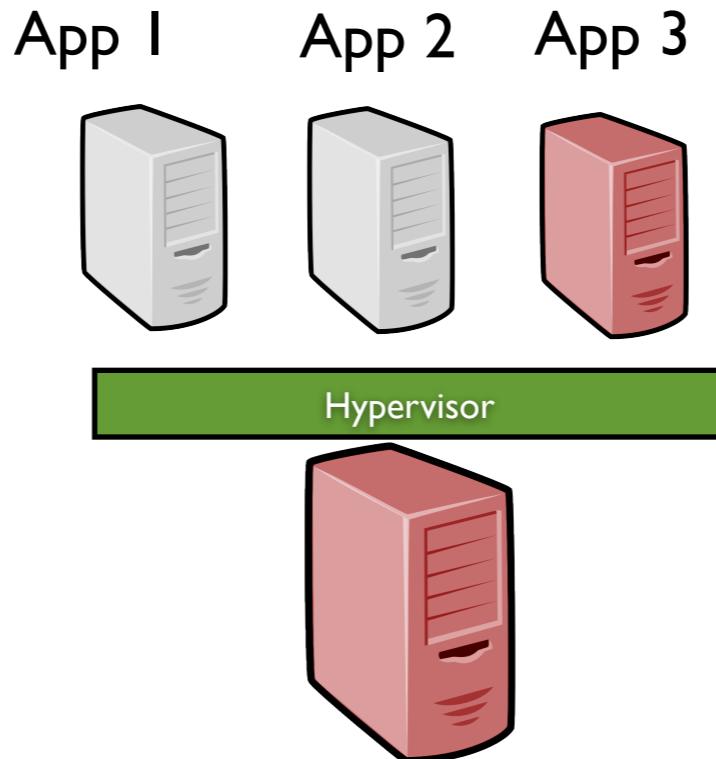


VM Capabilities

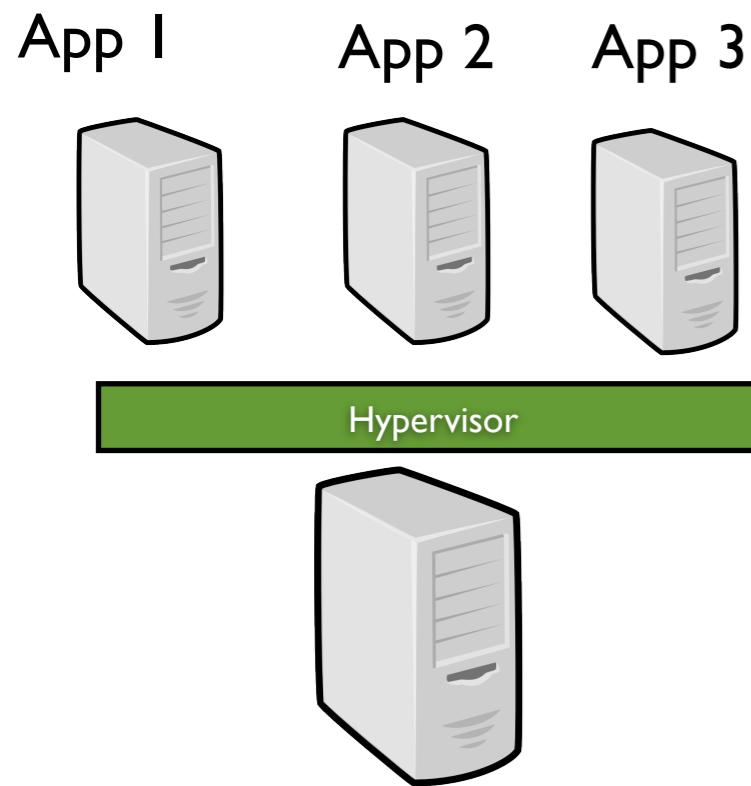


- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resumed from its latest consistent state)

- Suspend/Resume

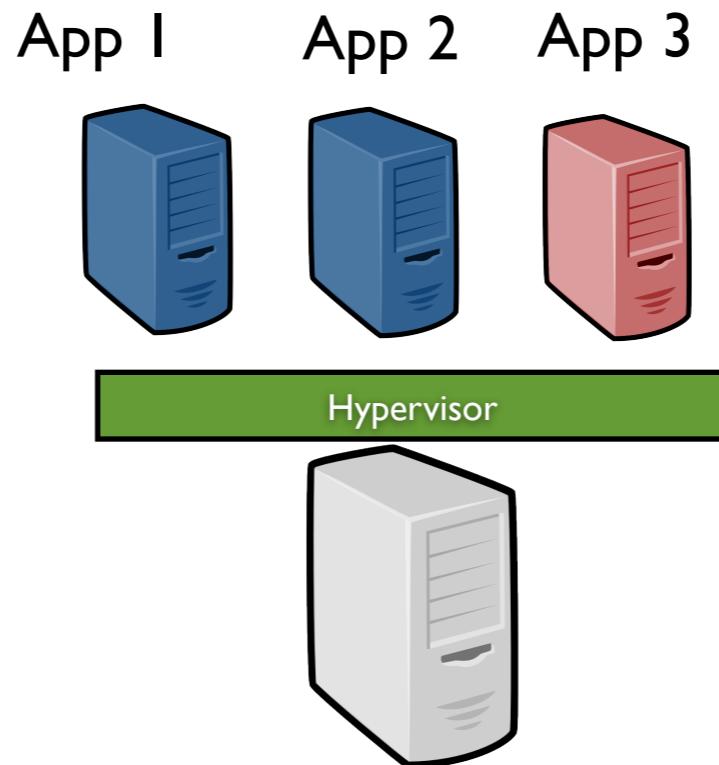


VM Capabilities

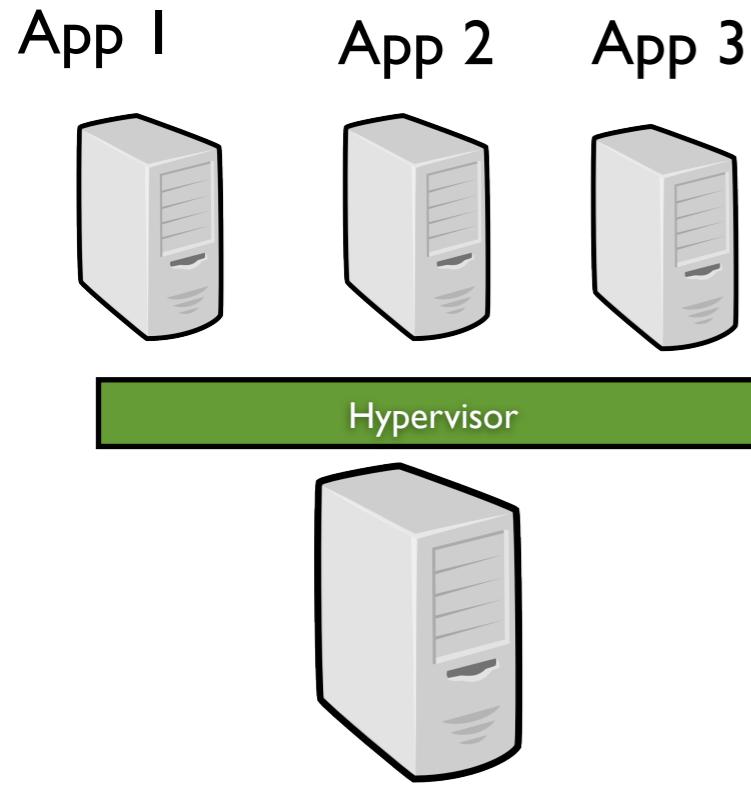


- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resumed from its latest consistent state)

- Suspend/Resume

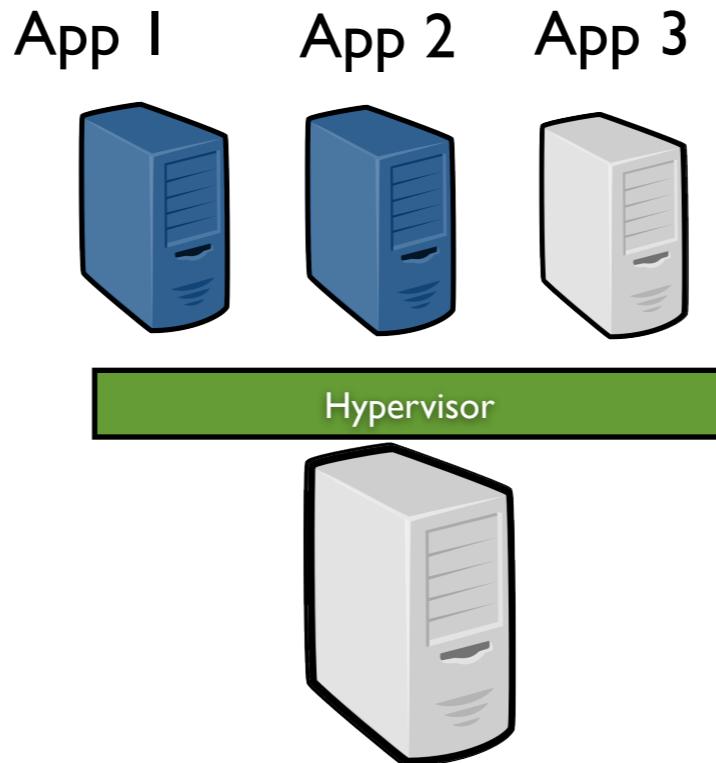


VM Capabilities

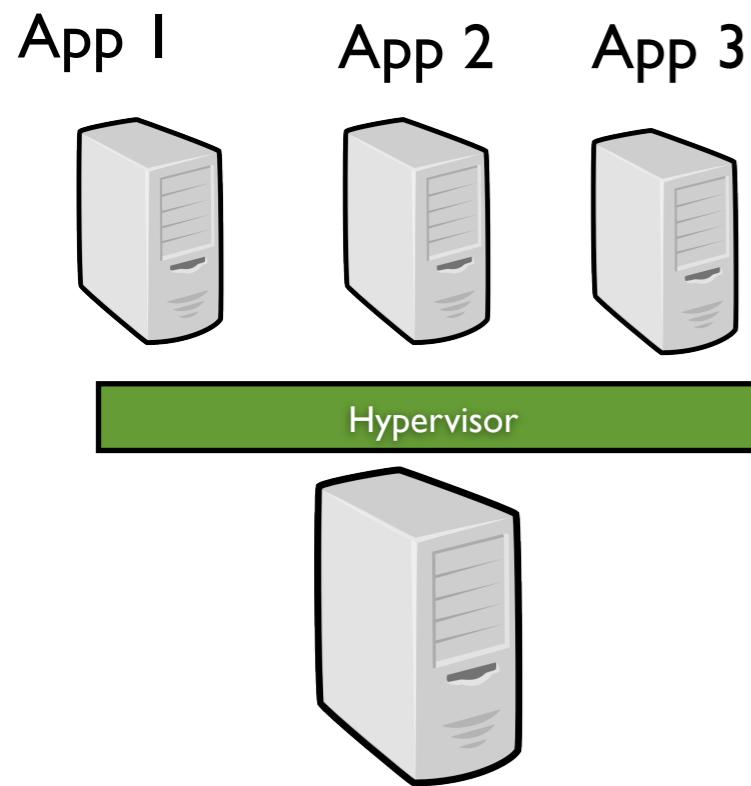


- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resumed from its latest consistent state)

- Suspend/Resume

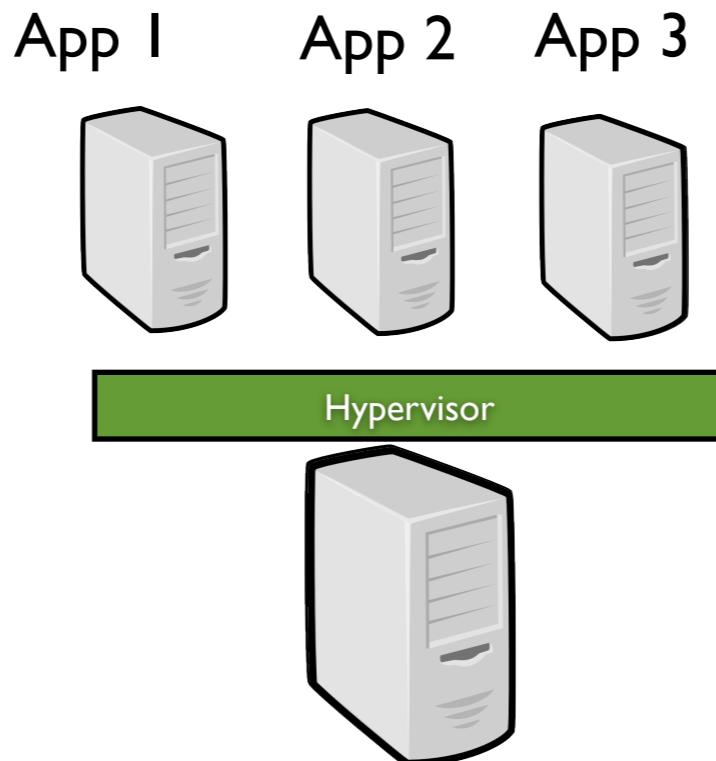


VM Capabilities

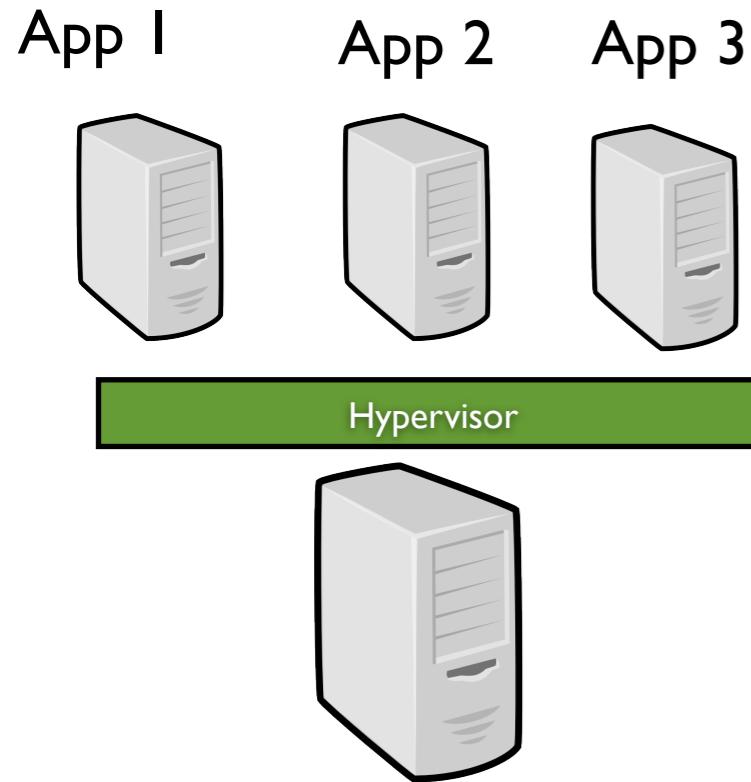


- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resumed from its latest consistent state)

- Suspend/Resume

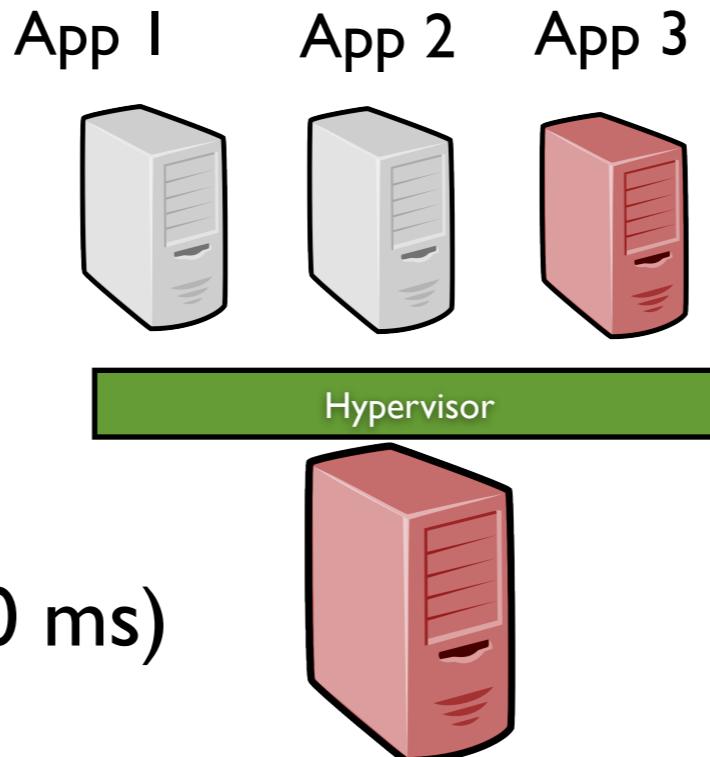


VM Capabilities

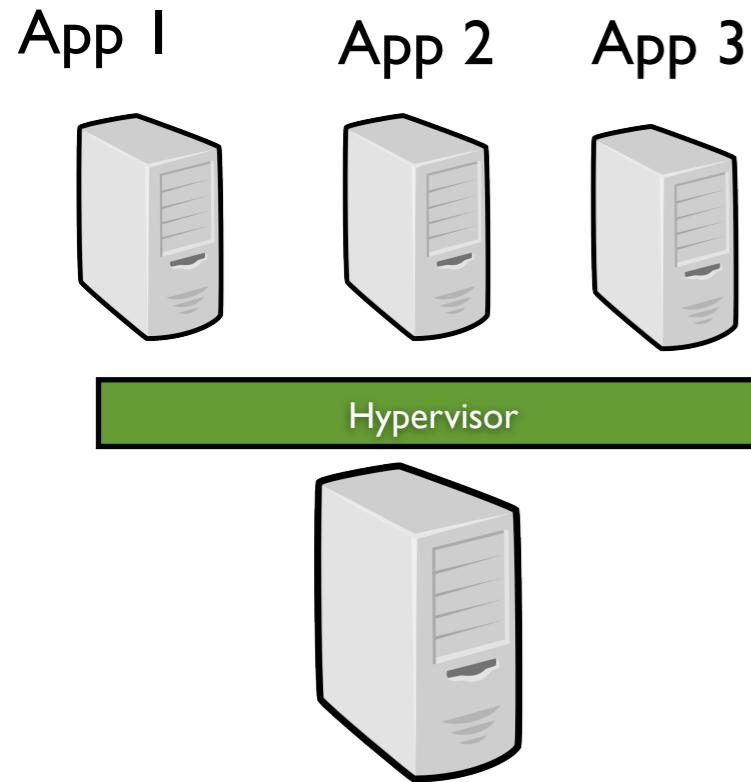


- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resumed from its latest consistent state)

- Suspend/Resume
- Live migration
(negligible downtime ~ 60 ms)
Post/Pre Copy

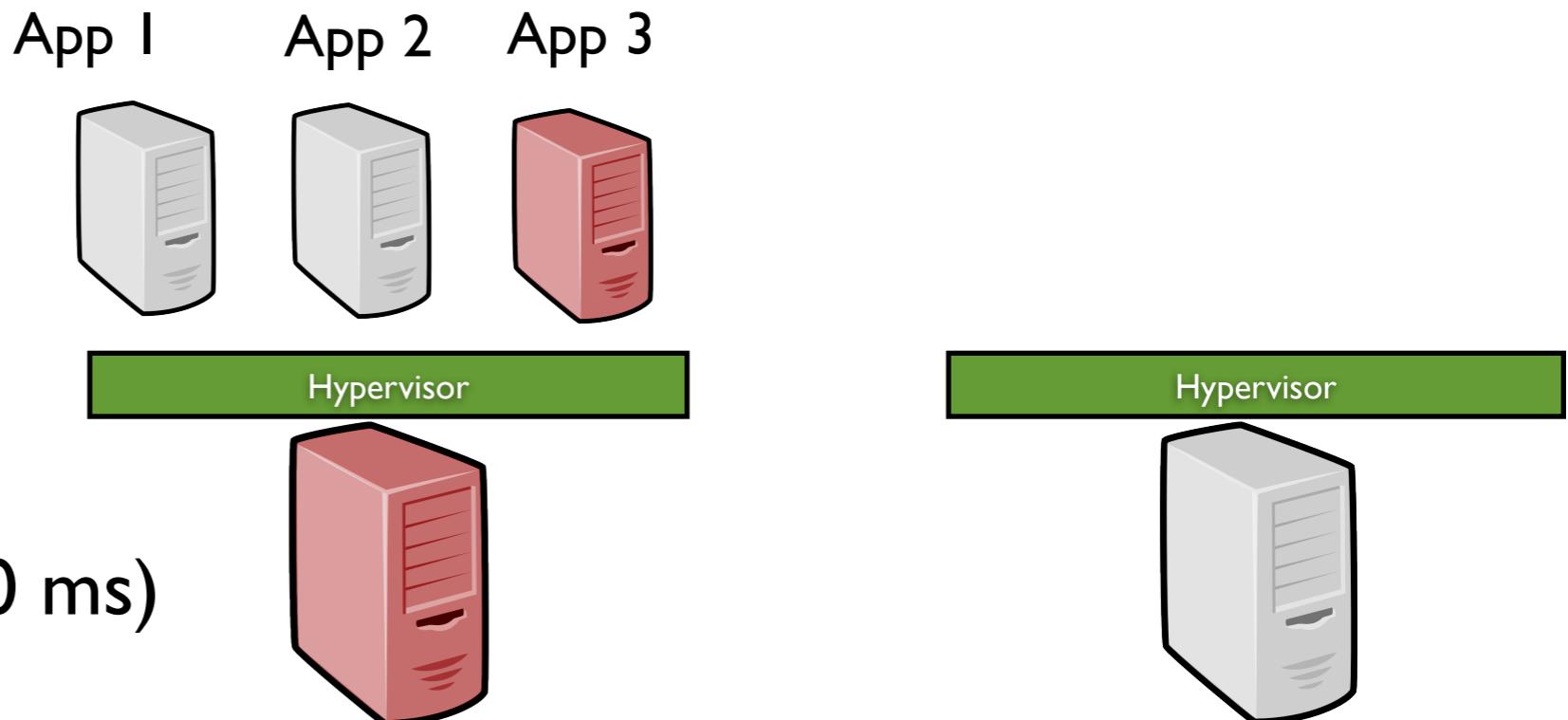


VM Capabilities

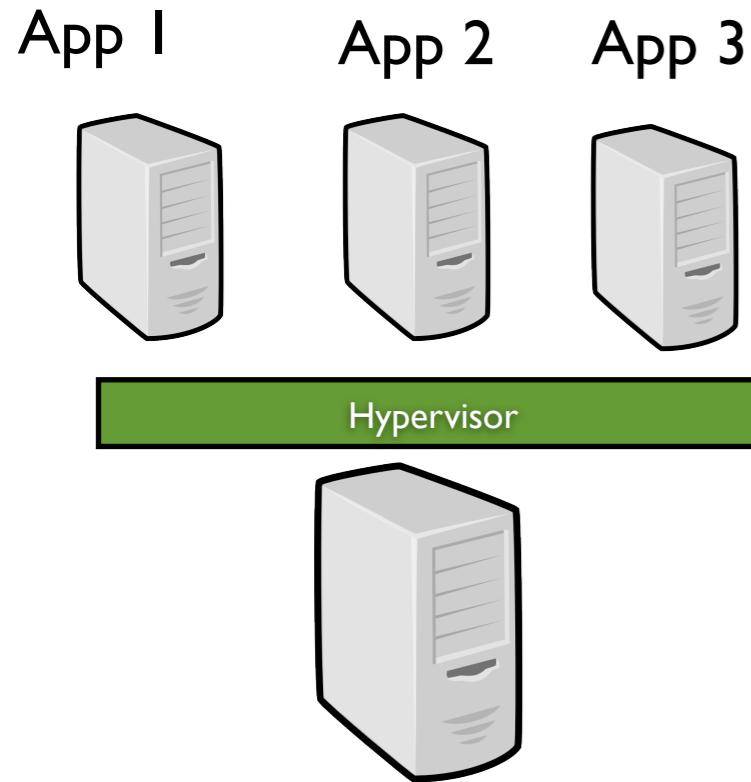


- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resumed from its latest consistent state)

- Suspend/Resume
- Live migration
(negligible downtime ~ 60 ms)
Post/Pre Copy

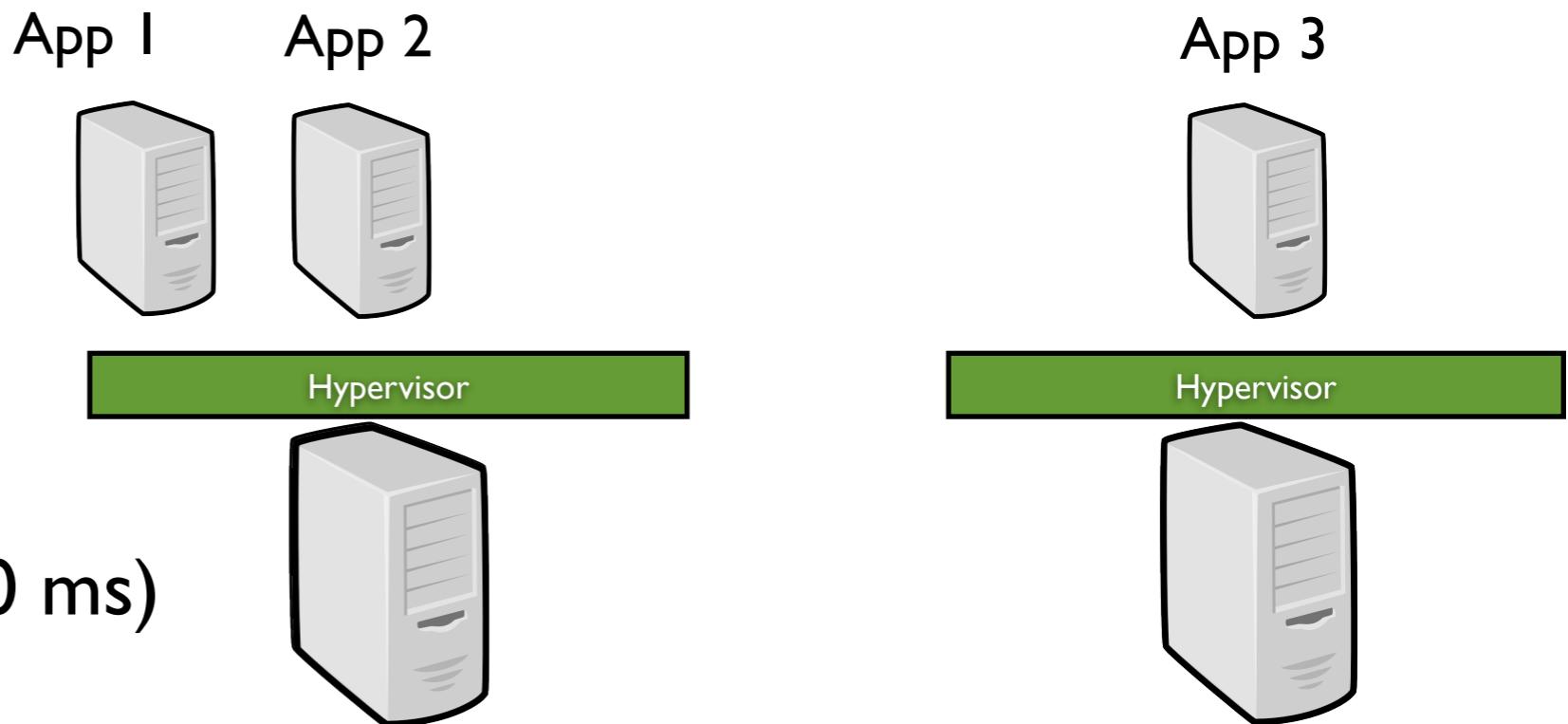


VM Capabilities

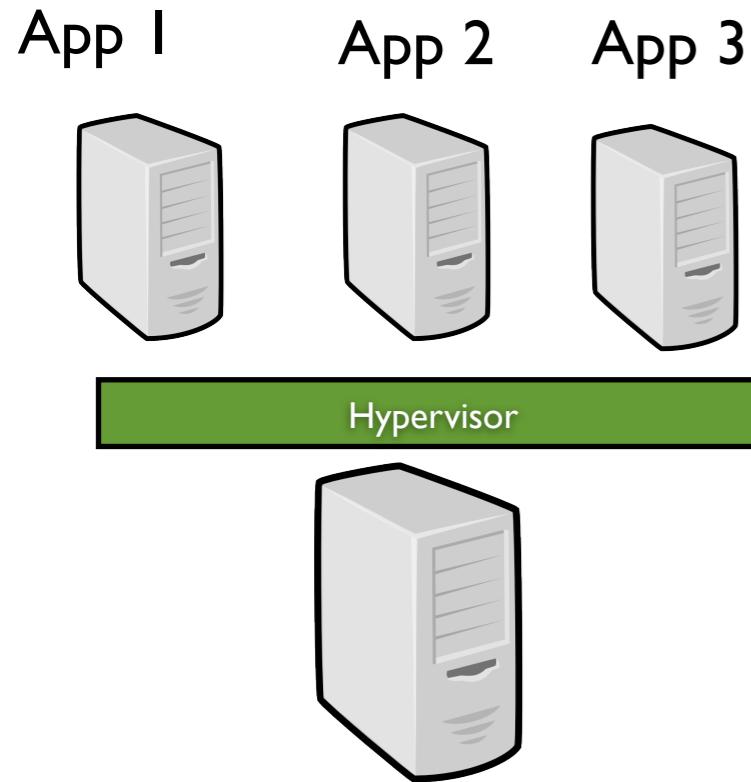


- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resumed from its latest consistent state)

- Suspend/Resume
- Live migration
(negligible downtime ~ 60 ms)
Post/Pre Copy

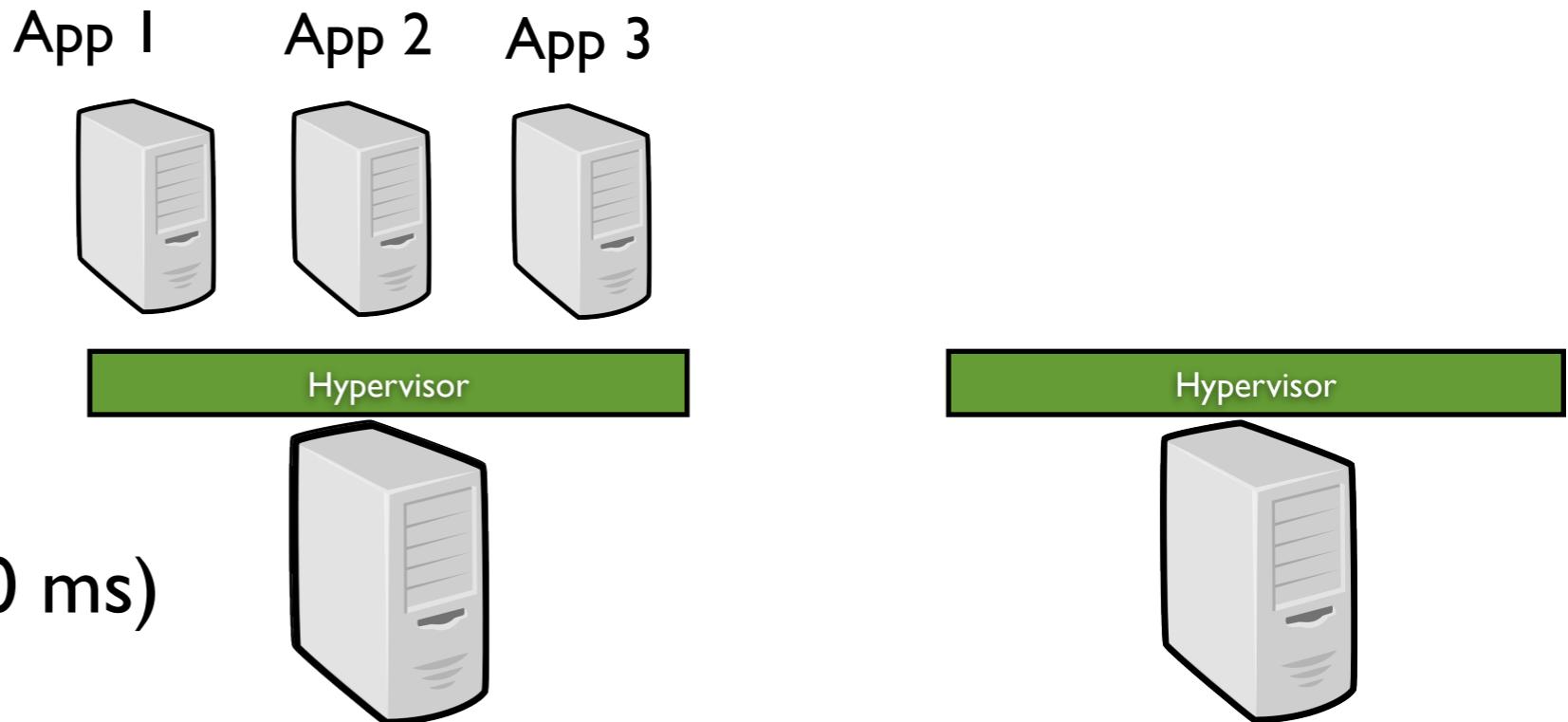


VM Capabilities

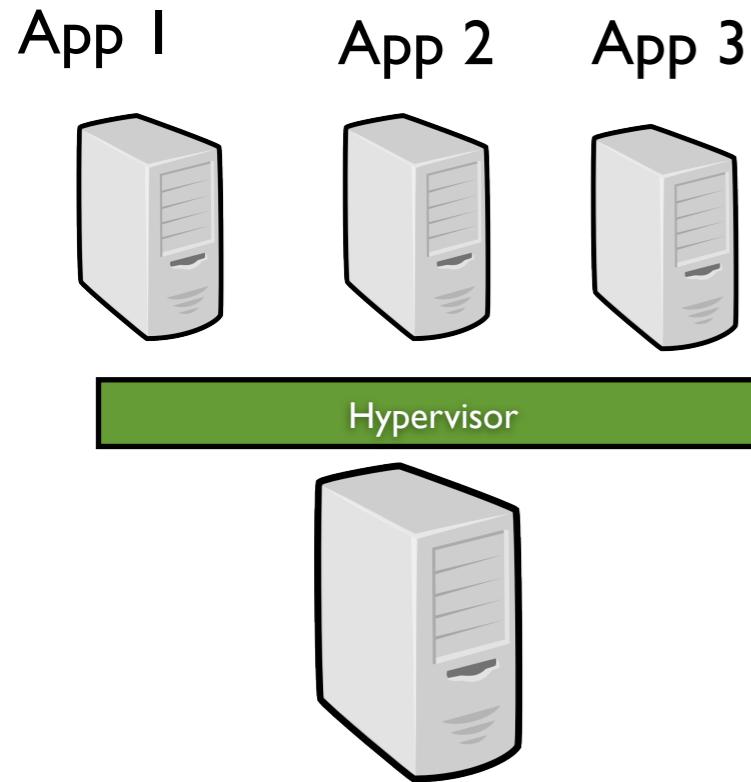


- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resumed from its latest consistent state)

- Suspend/Resume
- Live migration
(negligible downtime ~ 60 ms)
Post/Pre Copy



VM Capabilities



- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resumed from its latest consistent state)

- Suspend/Resume
- Live migration
(negligible downtime ~ 60 ms)
Post/Pre Copy

