INTRODUCTION TO SCIENTIFIC CLOUD COMPUTING

Stefano Bagnasco | INFN Torino

TWO USE CASES

- A small collaboration needs a 100-core farm to process its experimental data.
  - Relatively small data samples
  - Minimal manpower investment
  - There may be a large number of such tenants in a medium-sized computer centre
- A very large collaboration (think LHC or BES-III) needs a distributed computing infrastructure to simulate, reconstruct and analyze its data.
  - Big computing needs, big data
  - Relatively large manpower to develop, deploy and maintain software tools

TWO USE CASES: THE OLD WAY

- A small collaboration needs a 100-core farm to process its experimental data.
  - Buy a few servers, the relevant extra hardware and deploy a batch farm

    But:
  - manpower is needed to babysit the whole system, from the power supplies to the application software
  - the servers may be unused for a sizeable fraction of the time (at night, weekends, just after major conferences,...)

  - Possibly, share the batch system between different users

    But:
  - Different memory/CPU/OS/library requirements
  - And what can we do for “interactive” (i.e. non-batch) applications?

TWO USE CASES: THE OLD WAY

- A very large collaboration (think LHC or BES-III) needs a distributed computing infrastructure to simulate, reconstruct and analyze its data .
  - Develop and deploy a custom full-fledged Computing Grid (see later for exact definition)

    But:
  - a huge effort is needed to develop and maintain the non – industry standard middleware
  - collaboration and tool sharing across experiments has always been difficult
  - very difficult to use non-dedicated resources (all existing middleware is highly invasive)
  - and the resource sharing issue still holds.

Cloud Computing technologies promise to help solve such problems

What the cloud is:

- A technology to ease resource management, provisioning and sharing
- An industrial standard technology

What the cloud is not:

- A magical “resource multiplier”
- A complete end-to-end scientific computing system

TWO USE CASES: THE CLOUD WAY

- A small collaboration needs a 100-core farm to process its experimental data.
  - Deploy a "private cloud" in the Computer Centre, dynamically partition it according to needs and provision a "custom virtual computing farm" to each of the groups
- A very large collaboration (think LHC or BES-III) needs a distributed computing infrastructure to simulate, reconstruct and analyze its data.
  - Federate a number of such infrastructures and deploy a distributed computing system on top of them
  - Possibly, "cloudburst" to commercial providers such as Amazon

DEFINITION

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

Provisioning of ICT resources
As a Service

- 5 essential characteristics
- 3 service models
- 4 deployment models

http://goo.gl/DN1yrg

FUNDAMENTAL CHARACTERISTICS

- **On-demand self-service.**
  - A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
- **Broad network access.**
  - Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client.
- **Resource pooling.**
  - Computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.
- **Rapid elasticity.**
  - Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand.
- **Measured service.**
  - Cloud systems automatically control and optimize resource use by leveraging a metering capability at a level of abstraction appropriate to the type of service.

http://goo.gl/DN1yrg

FUNDAMENTAL CHARACTERISTICS

- **On-demand self-service.**
  - It is easy to rent a car, you can book it by phone or online
- **Broad network access.**
  - There is a broad network of rental car agencies around the world to give you access to a car rental
- **Resource pooling.**
  - The rental car companies manage a pool of cars in any given city to meet demand. You don't have to worry about it. If one agency is out of cars they will often refer you to another to help you find a car.
- **Rapid elasticity.**
  - Rental car companies move cars into a particular location when there is a large event and they know demand will be high. They scale up and down to meet the demand.
- **Measured service.**
  - You pay only for the time you used the car. Once you turn it back in you are done. No maintenance, insurance, fuel, tires, etc.

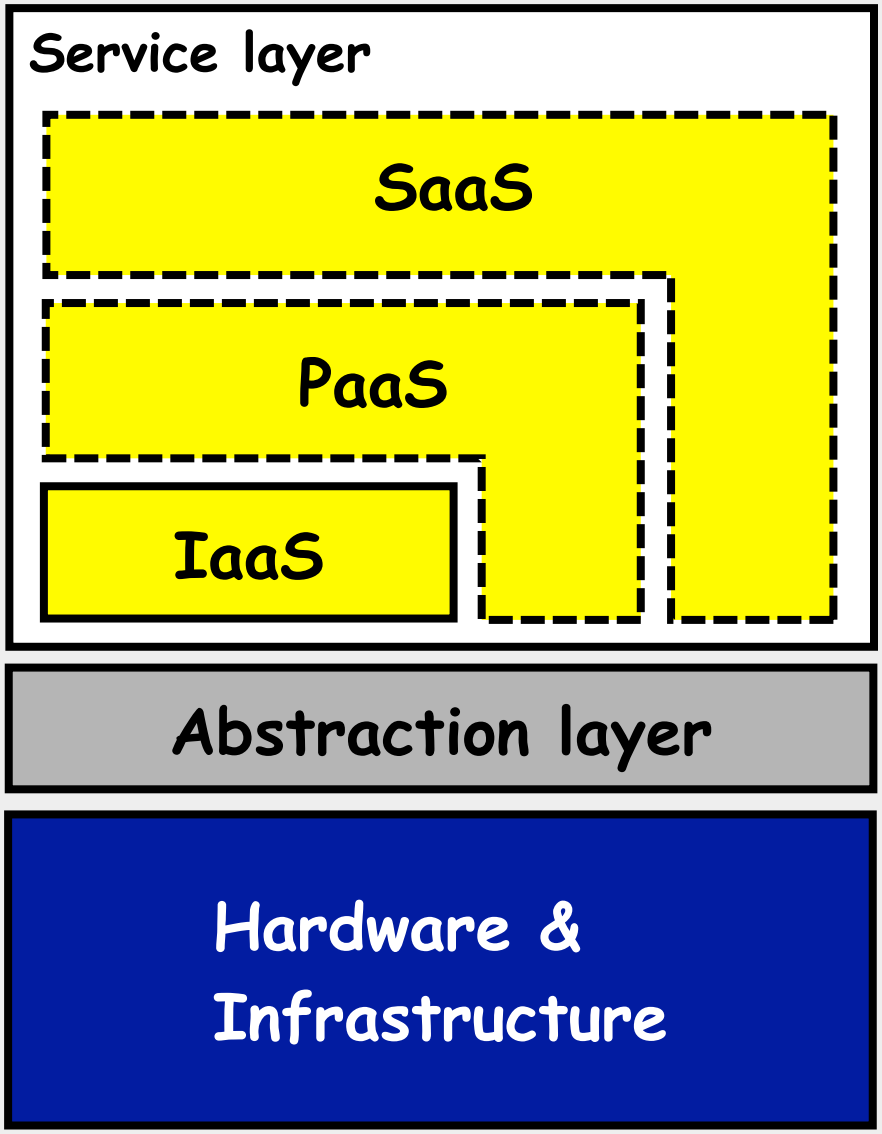
http://goo.gl/cEa8M

SERVICE MODELS

- **IaaS:** Infrastructure as a Service.
  - The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).
- **PaaS:** Platform as a Service.
  - The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.
- **SaaS:** Software as a Service.
  - The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited userspecific application configuration settings.

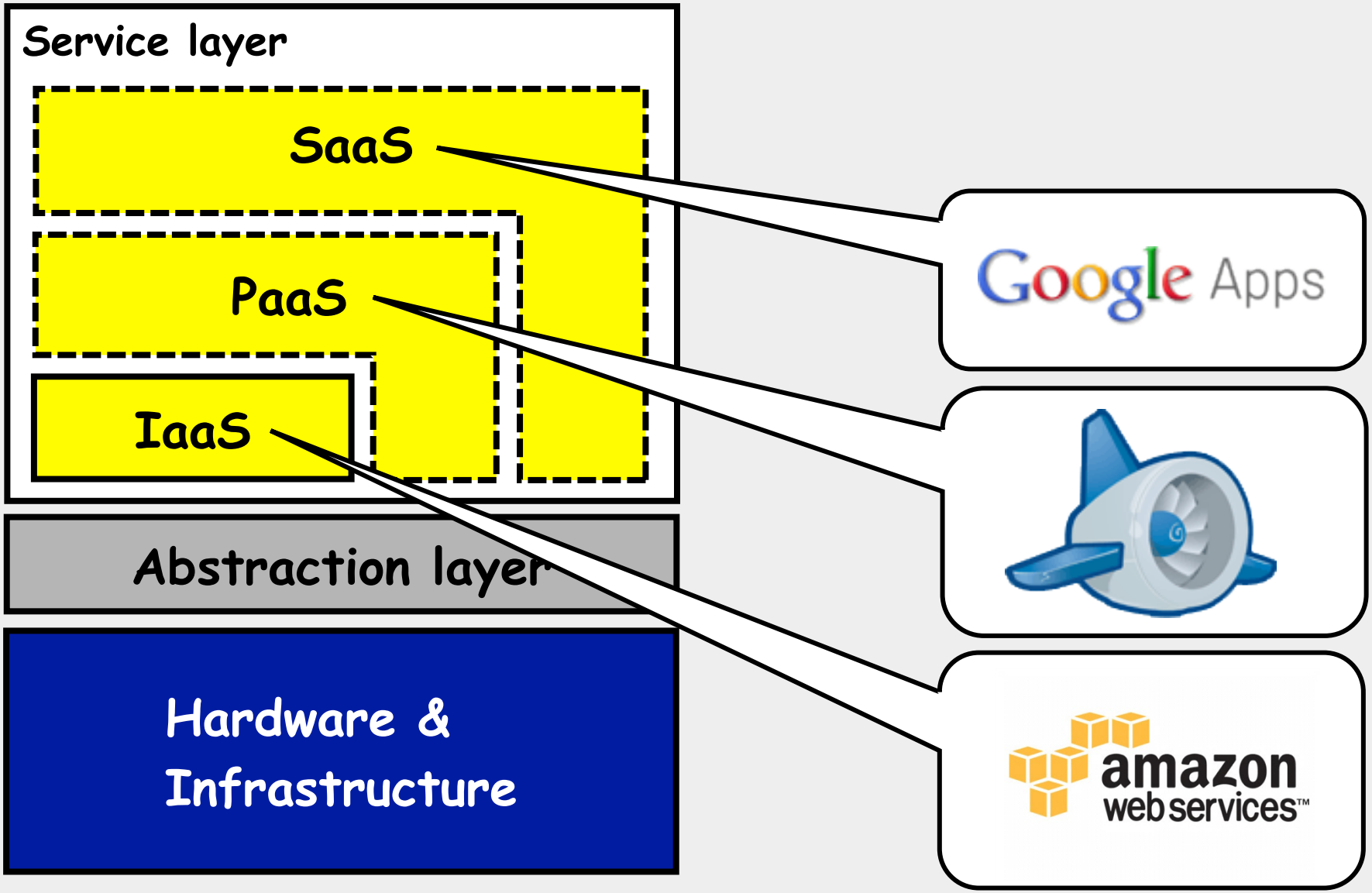
http://goo.gl/DNlyrg

SERVICE MODELS

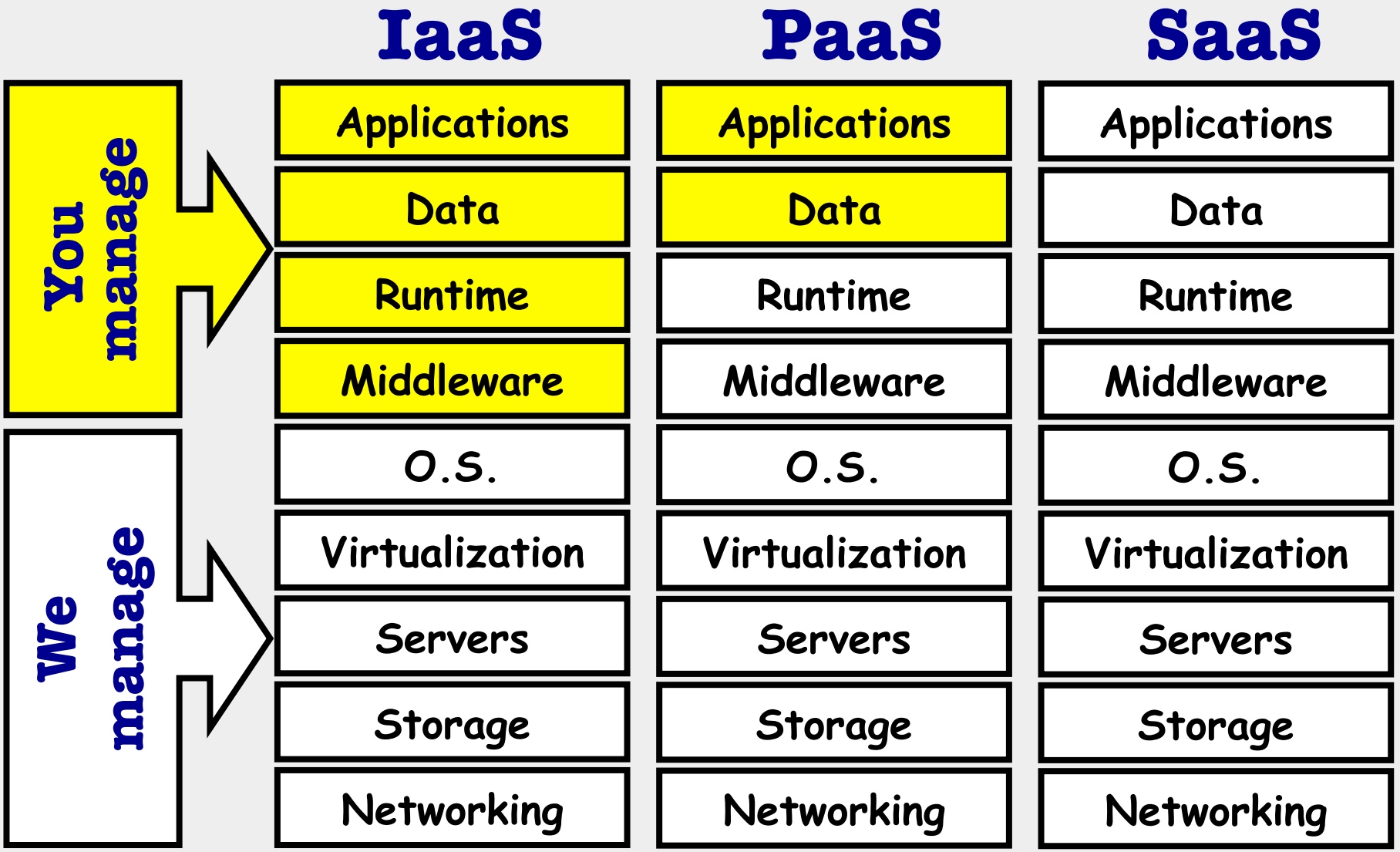


- Software-as-a-Service
- Platform-as-a-Service
- Infrastructure-as-a-Service

SERVICE MODELS



SERVICE MODELS



http://goo.gl/hfaVDX

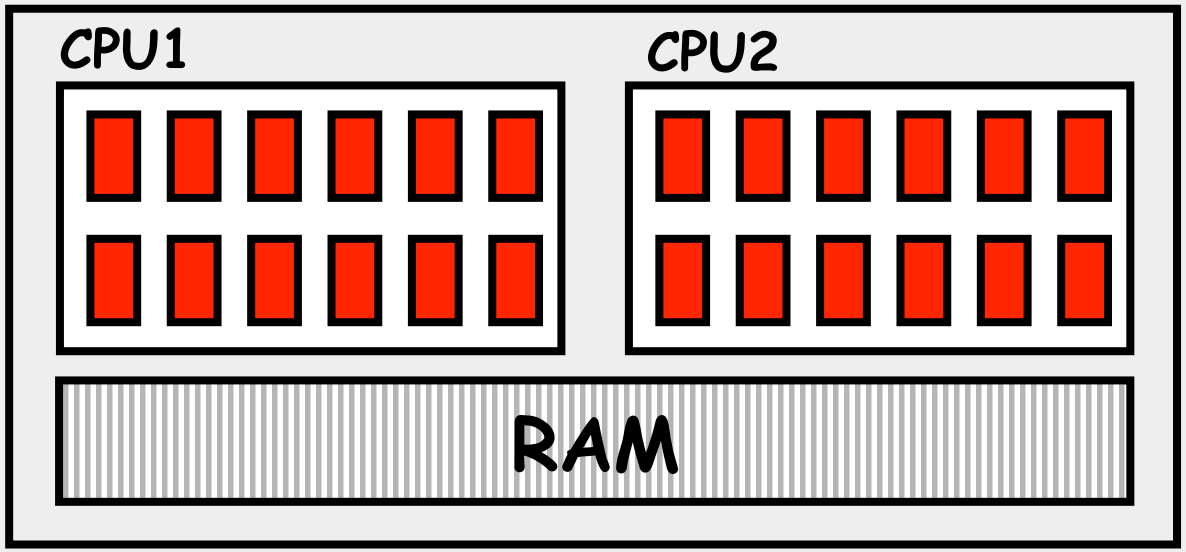
DEPLOYMENT MODELS

- **Private cloud**
  - The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise.
- **Community cloud**
  - The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise.
- **Public cloud**
  - The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.
- **Hybrid cloud**
  - The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for loadbalancing between clouds).

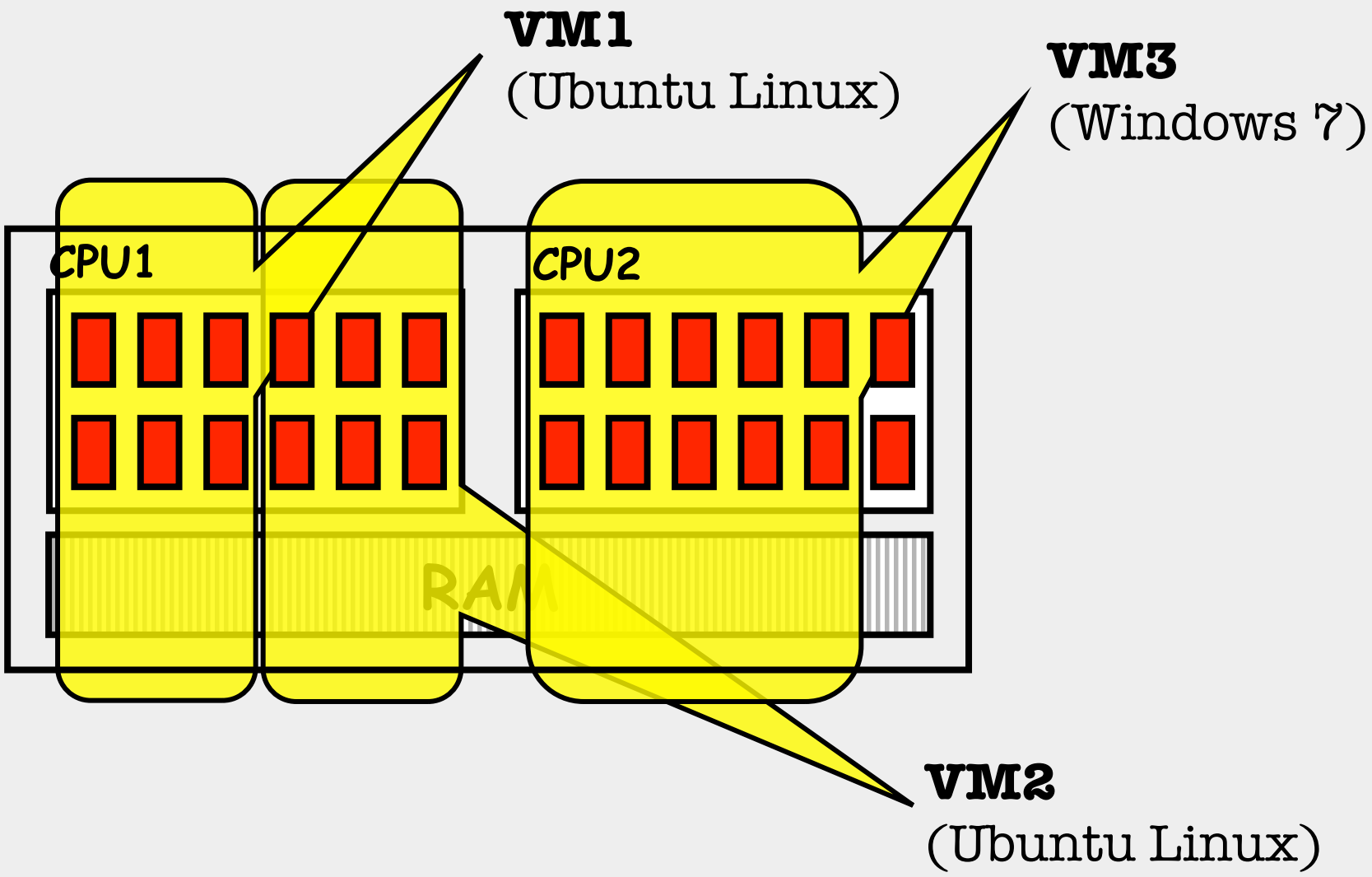
http://goo.gl/DN1yrg

HOW DOES IT WORK IN PRACTICE?

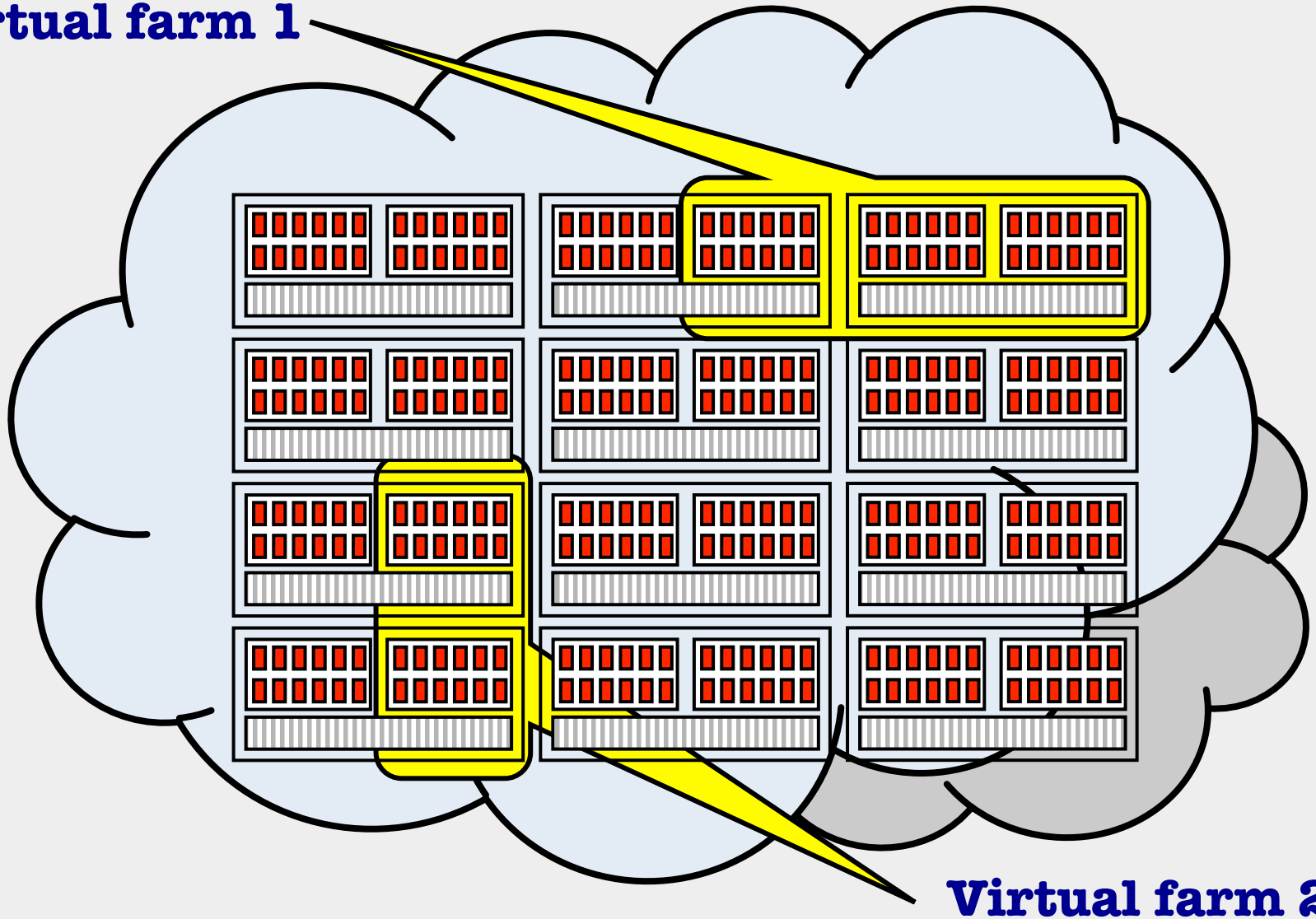
- The key technology is **virtualization**
  - Many more details about it in next lesson



HOW DOES IT WORK IN PRACTICE?



HOW DOES IT WORK IN PRACTICE?



Cloud Computing is not (just) virtualization!

- Remember?
  - Self-service.
  - Broad network access.
  - Resource pooling.
  - Rapid elasticity.
  - Measured service.

You have to provide tools and resources to allow for this, or you're just virtualizing your data center!

SHOPPING LIST

- Hardware
  - Servers, storage, network
- Virtualization tools
  - E.g. KVM (or Xen, VMWare,...)
- OS Images
  - See next lesson
- Contextualization tools
  - See below
- Cloud platform
  - OpenNebula (but also OpenStack, CloudStack, Eucalyptus, VMWare,...)
  - Includes APIs and User Interfaces

TWO APPLICATION FEATURES

- High Availability
  - Virtual machines can be migrated from a host to another (either with a short interruption or even “live”)
  - Critical services are resilient to hardware failures or scheduled maintenances
- Scalability
  - New identical virtual machines can be instantiated quickly and easily from a common image + context
  - Horizontally scalable applications can easily scale out (and back in)

CLOUD AWARE APPS

- Cloud-aware applications:
  - Intrinsically distributed
  - Stateless
  - Failover managed in-app
  - Scaling managed in-app

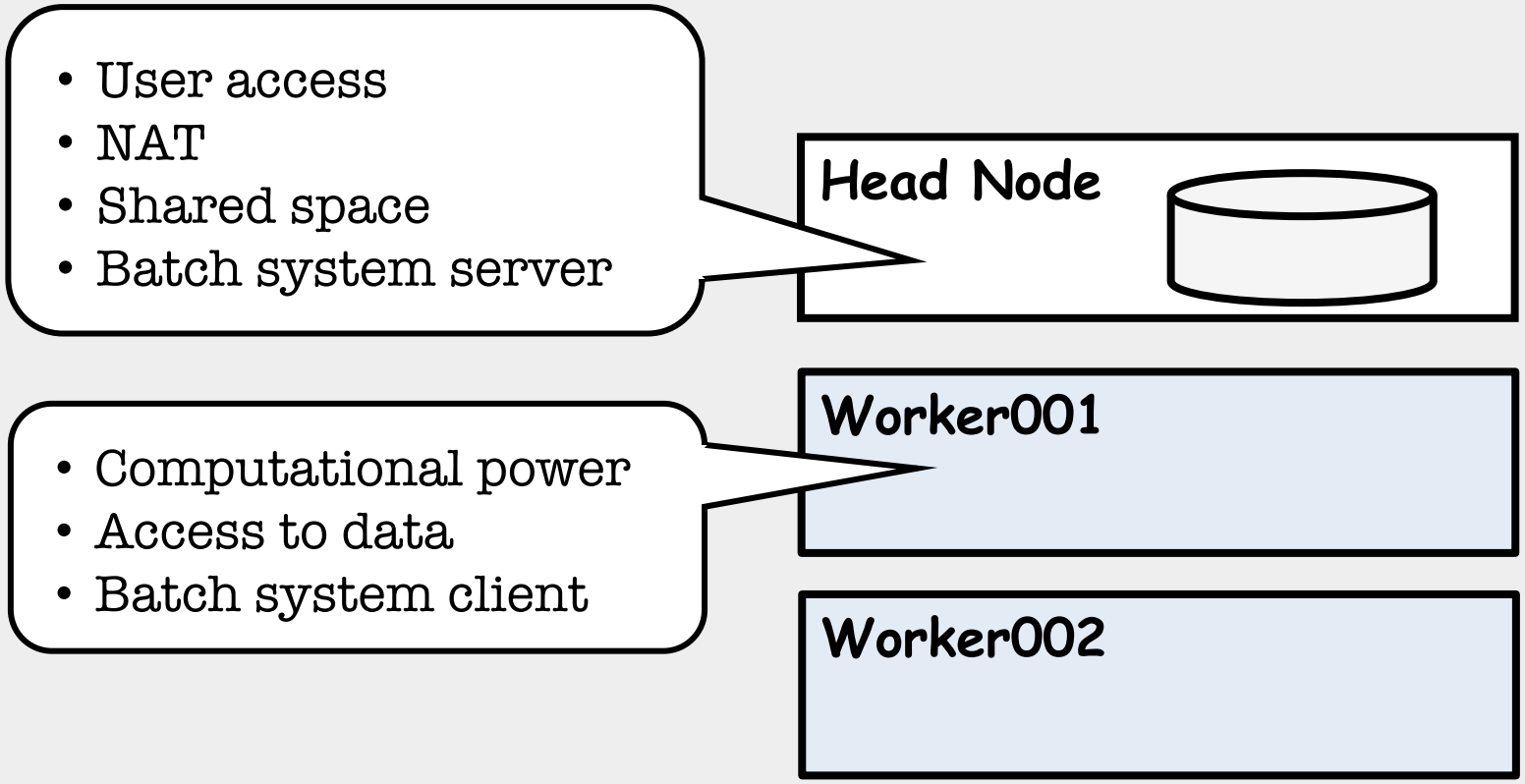
- “Legacy” applications:
  - Client-server
  - No horizontal scalability
  - Failover managed by infrastructure
  - Scaling managed by infrastructure

CATTLE AND PETS

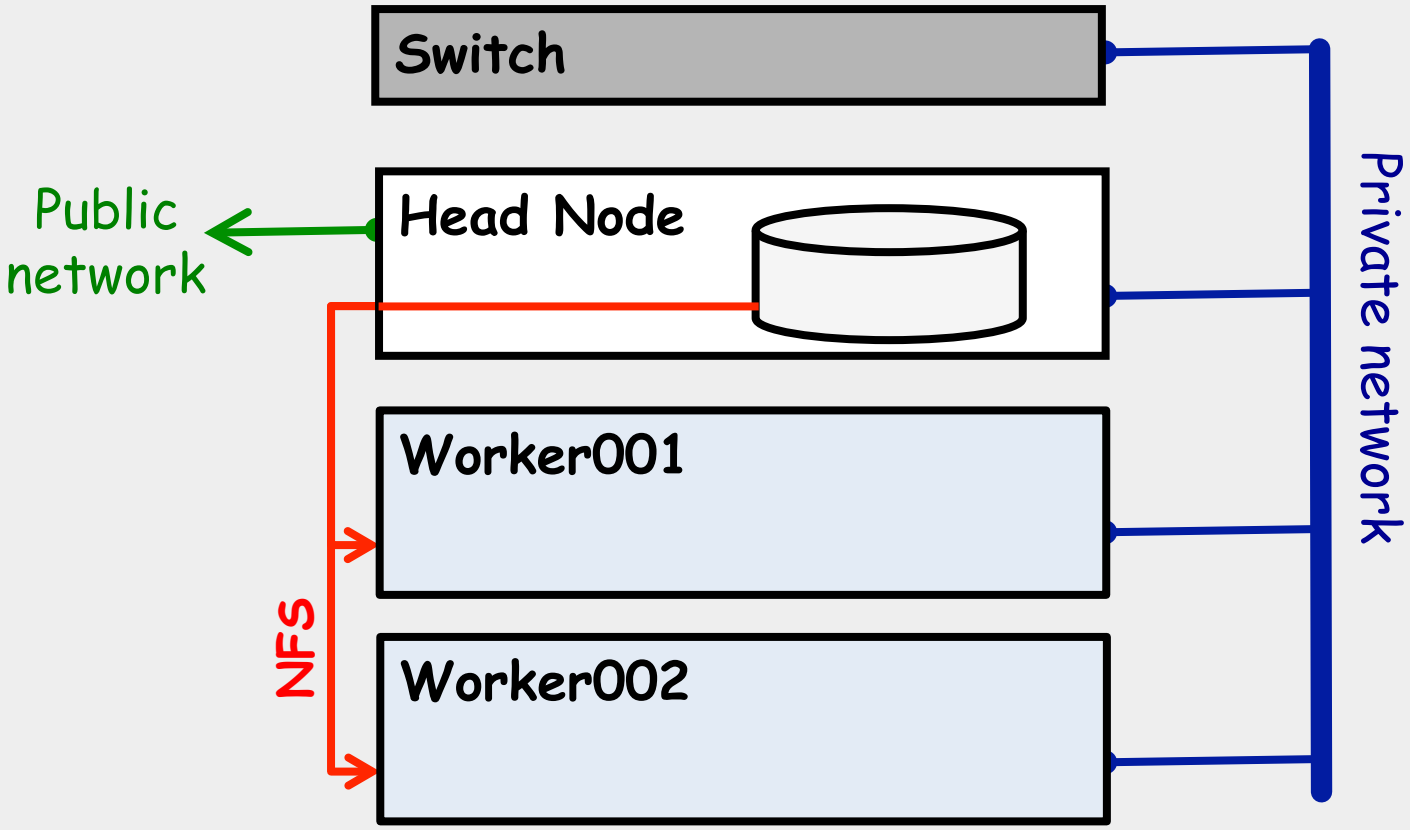
- Cloud-aware applications are like cattle
  - They're expendable
  - If they get sick, you replace them with another one

- Legacy applications are like pets
  - You take care of them
  - If they're sick, you call the vet

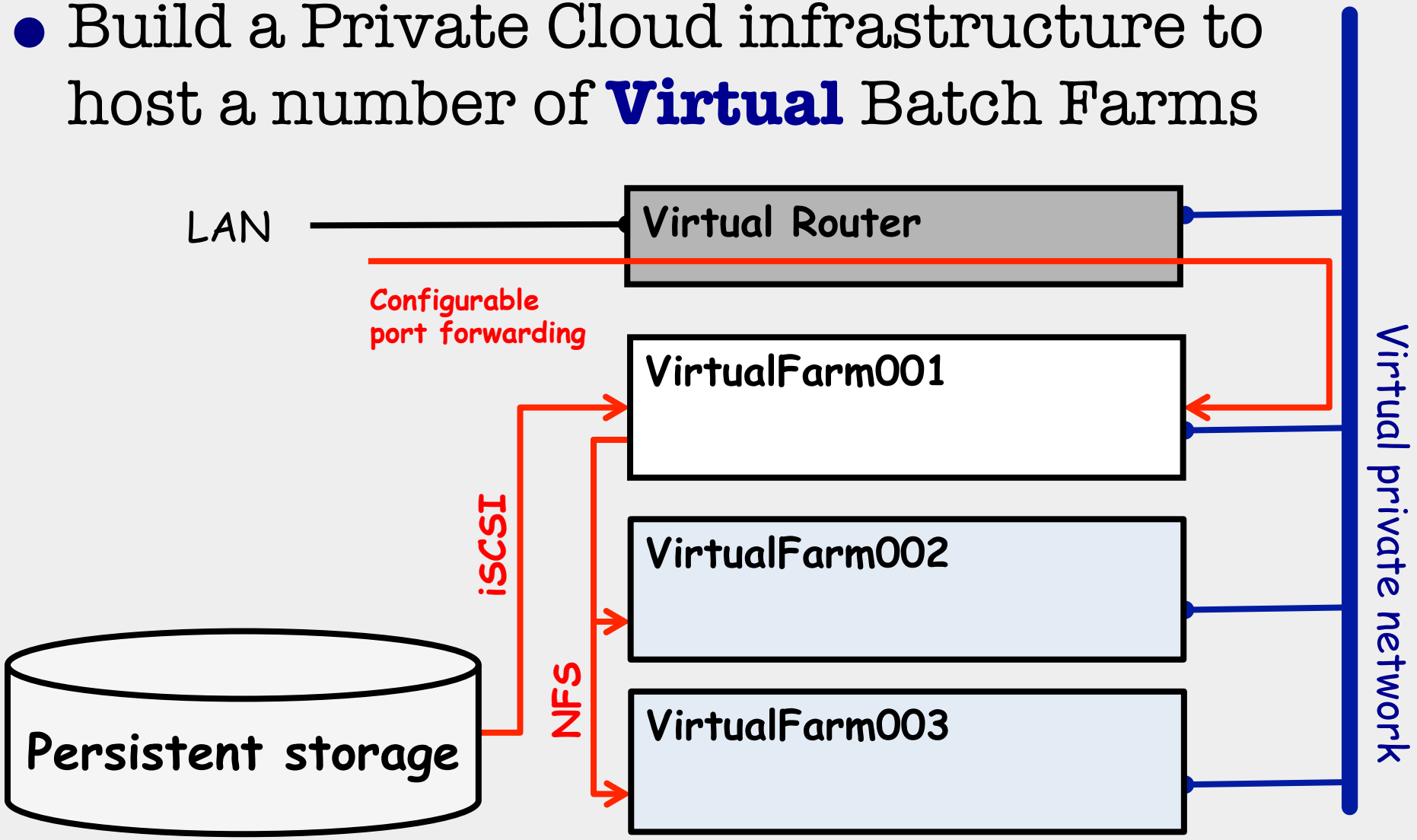
- Build a Private Cloud infrastructure to host a number of Virtual Batch Farms



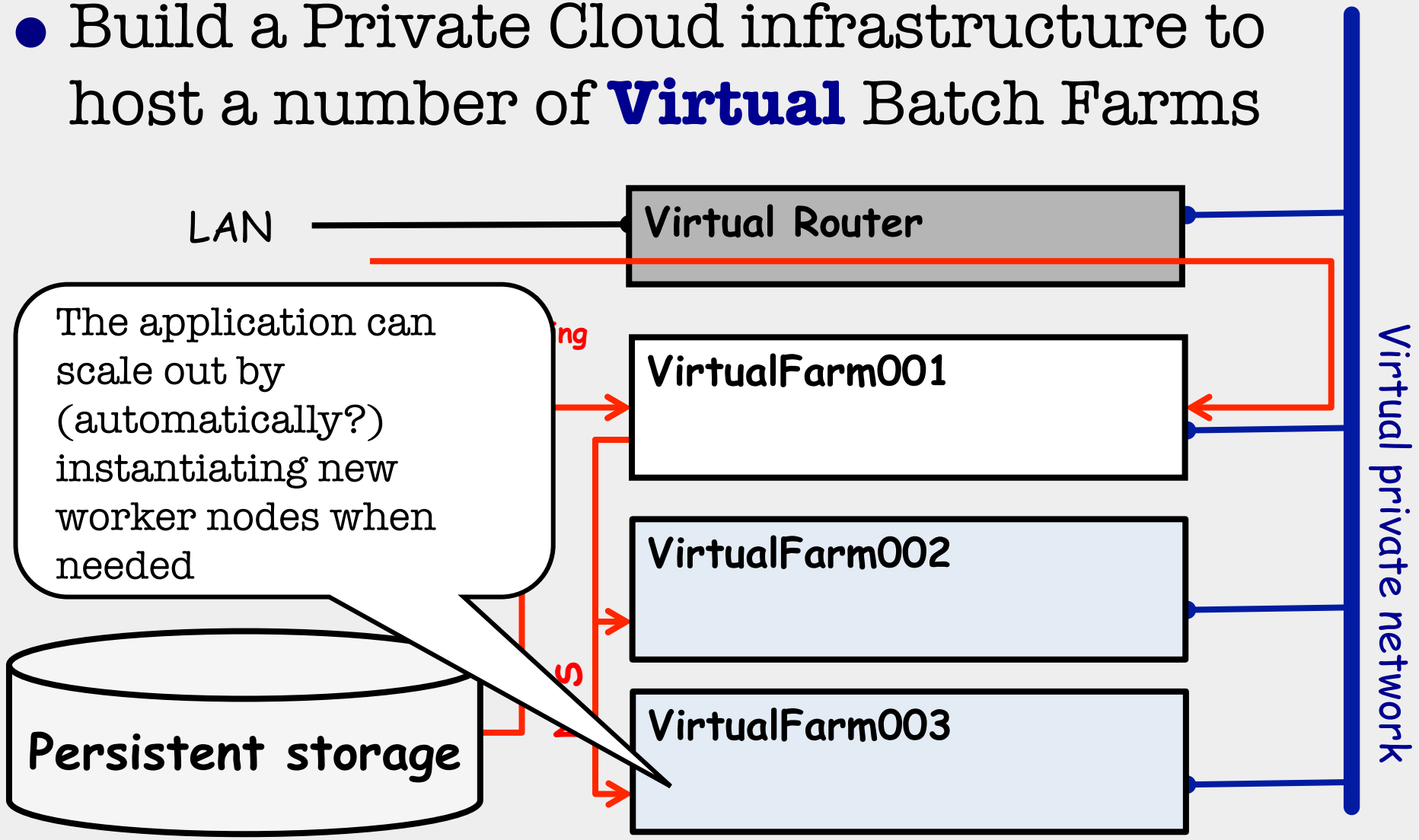
- Build a Private Cloud infrastructure to host a number of Virtual Batch Farms



- Build a Private Cloud infrastructure to host a number of **Virtual** Batch Farms



- Build a Private Cloud infrastructure to host a number of **Virtual** Batch Farms



A WORD ON SHARES, QUOTAS AND PRIORITIES

- Perfect elasticity work only with infinite resources
  - Or a reasonable approximation thereof
  - Commercial cloud providers have virtually infinite resources: you need more, they buy more, you pay more
- Scientific computing has infinite needs
  - The more computing power you give to scientists, the more complex the problem they will try to solve
  - And, the resources are usually scarce
- So, scientific cloud computing facilities run in saturation most of the time

A WORD ON SHARES, QUOTAS AND PRIORITIES

- Scientific cloud computing facilities run in saturation most of the time
  - Cloud schedulers are FIFO buffers, not FairShare systems
  - A VM either runs nearly immediately (if there are available resources) or fails (if the infrastructure is saturated or the user exhausted her quota)
  - “The tragedy of the commons”
- This is still an unsolved problem
  - Keep some resources artificially free at all time?
  - Use short-lived VMs?
  - Write more sophisticated schedulers?

When instantiating a new VM...

- **Contextualisation** is the sequence of actions required to start from a basic OS image and configure it for its role:
  - Create users
  - Partition and mount ephemeral disk
  - Mount persistent disks
  - Install extra software
  - Configure whatever is to be configured
  - ...

When instantiating a new VM...

- Start from a very basic OS image and use tools to configure it
  - Decouple OS from application
  - Maintain only a limited number of images
  - Basic OS images are small
  - Contextualization scripts can be complex and difficult to maintain
  - Complex contextualisations can be (very) slow
- Instantiate an OS image, configure it, save it and re-use it for subsequent instantiations
  - Short VM start-up time
  - Saved images are always consistent
  - Large number of images to maintain
  - Fully configured images can be very large

REFERENCES

- P. Mell and T. Grance, *The NIST Definition of Cloud Computing*, NIST Special Publication 800-145 (2011)
- T. Erl, Z. Mahmood and R. Puttini, *Cloud Computing. Concepts, Technology and Architecture.* Upper Saddle River: Prentice Hall (2013)
- OpenNebula.org, *Open Cloud Reference Architecture* (2015)
- T. Sandholm and D. Lee, “Notes on Cloud computing principles” *Journal of Cloud Computing* **3**:21 (2014)