ROYAL INSTITUTE
OF TECHNOLOGY

Intro to Cloud Computing

ID2210

Jim Dowling

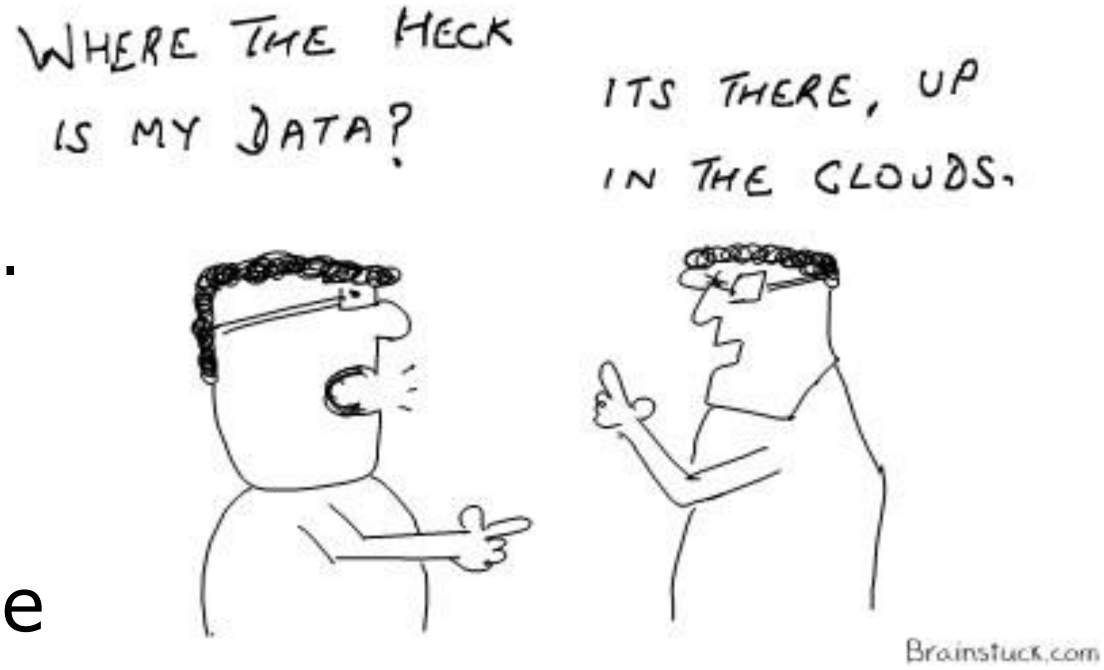
What Cloud Computing is Not

- CLOUDS OF CONFUSION AS GALWAY COUNCILLOR TELLS ANOTHER TO "GO F..K HIMSELF"
- A GALWAY councillor has refused to apologise for swearing at a County Council committee meeting after he told a fellow councillor to "go **** himself". Local area councillor Seamus Tiernan made the amazing outburst after he was told he was a "feckin eejit" for thinking that cloud computing was only suitable in areas with lots of rain. He had told the Infrastructure Committee meeting this week that his native Connemara would be ideal for cloud computing because it has heavy cloud cover for nine months of the year."
- The Independent councillor said that the Government should be doing more to harness clean industries for the Connemara area and he named wind energy and cloud computing as two obvious examples. "Connemara in particular could become a centre of excellence for wind energy harnessing, as it is open to the Atlantic. Also in terms of cloud computing, we have dense thick fog for nine months of the year, because of the mountain heights and the ability to harness this cloud power, there is tremendous scope for cloud computing to become a major employer in this region." However his mistake was pointed out by an incredulous Cllr Martin Shiels who said that "this is taking the biscuit. I've heard it all now. You must be a fecking eejit to think that the cloud computing had anything to do with climate."
- Cllr Tiernan took umbrage at the remarks of his colleague and called for them to be withdrawn. When Cllr Shields refused to do so, Tiernan said "go **** yourself, Cllr Shields." Chairman Sile Ni Baoill asked for both councillors to withdraw their comments, but Cllr Tiernan was repentant that Cllr Shields was wrong and that cloud computing is linked to cloud cover. "Tell me why large companies are opening server farms in cold wet countries then," he asked Cllr Shields.

Democratization of Large-Scale Computing

- **Cloud computing** is the delivery of hosting services that are provided to a client over the Internet.
  - Enable large-scale services without up-front investment.

- **New programming tools, databases and systems** have enabled the low-cost construction of large-scale services.



Clouds are Elastic

- NIST Definition of Cloud Computing

  "Cloud computing is a model for enabling 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."

Supporting Technologies

- Enormous computer data-centres containing **commodity hardware**.
- **Virtualization** of computation, storage, and communication.
  - Turn hardware and networking into software!
- Achieve **economies of scale**.
  - Reduce costs of electricity, bandwidth, hardware, software and use low-cost locations.
  - Lower-cost than provisioning own hardware.
- **NoSQL datastores and distributed filesystems** have enabled storage scalability to much higher levels than relational databases.

Cloud Computing Essentials

- Cloud computing is Utility Computing
- Cloud services are controlled and monitored by the cloud provider typically through a pay-per-use business model.

- An ideal cloud computing platform is:
  - efficient in its use of resources
  - scalable
  - elastic
  - self-managing
  - highly available and accessible
  - inter-operable and portable

Cloud Properties

- **Resource efficiency:** computing and network resources are pooled to provide services to multiple users. Resource allocation is dynamically adapted according to user demand.

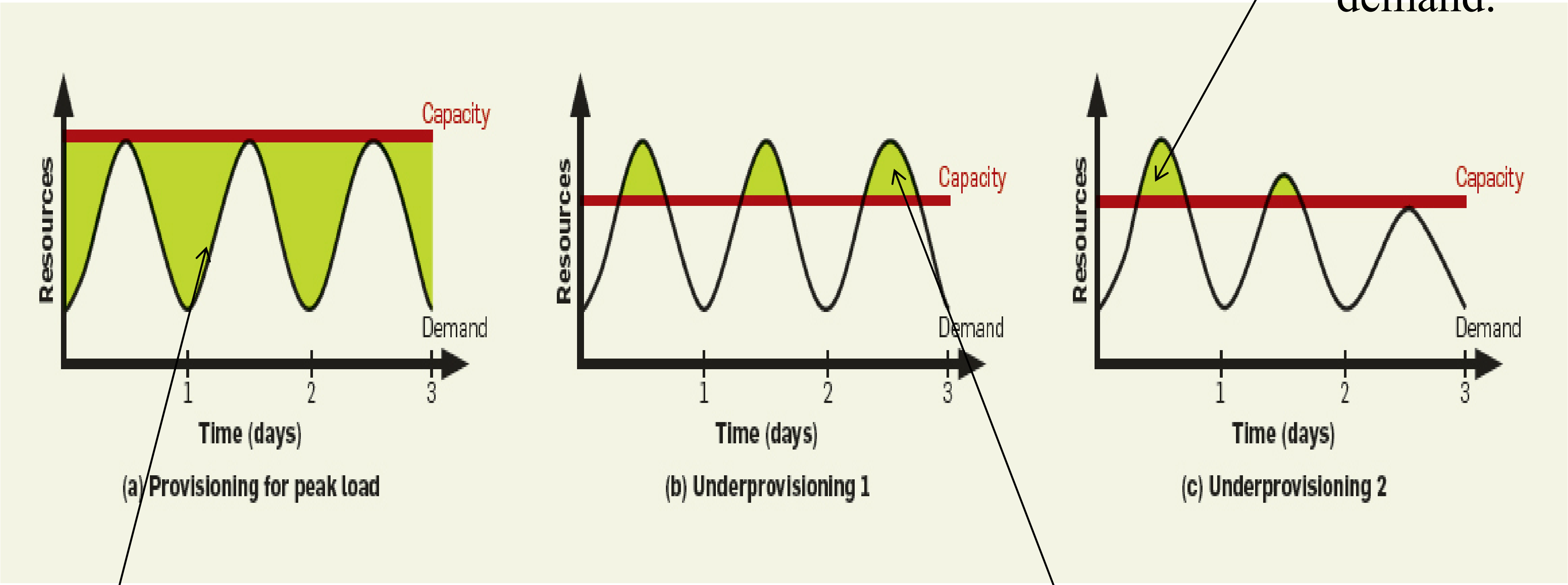
- **Elasticity:** computing resources can be rapidly and elastically provisioned to scale up, and released to scale down based on consumer’s demand.

Cloud Properties

- **Self-managing services:** a consumer can provision cloud services, such as web applications, server time, processing, storage and network as needed and automatically without requiring human interaction with each service's provider

- **Accessible and highly available:** cloud resources are available over the network anytime and anywhere and are accessed through standard mechanisms that promote use by different types of platform (e.g., mobile phones, laptops, and PDAs).

Over or Under-Provisioning



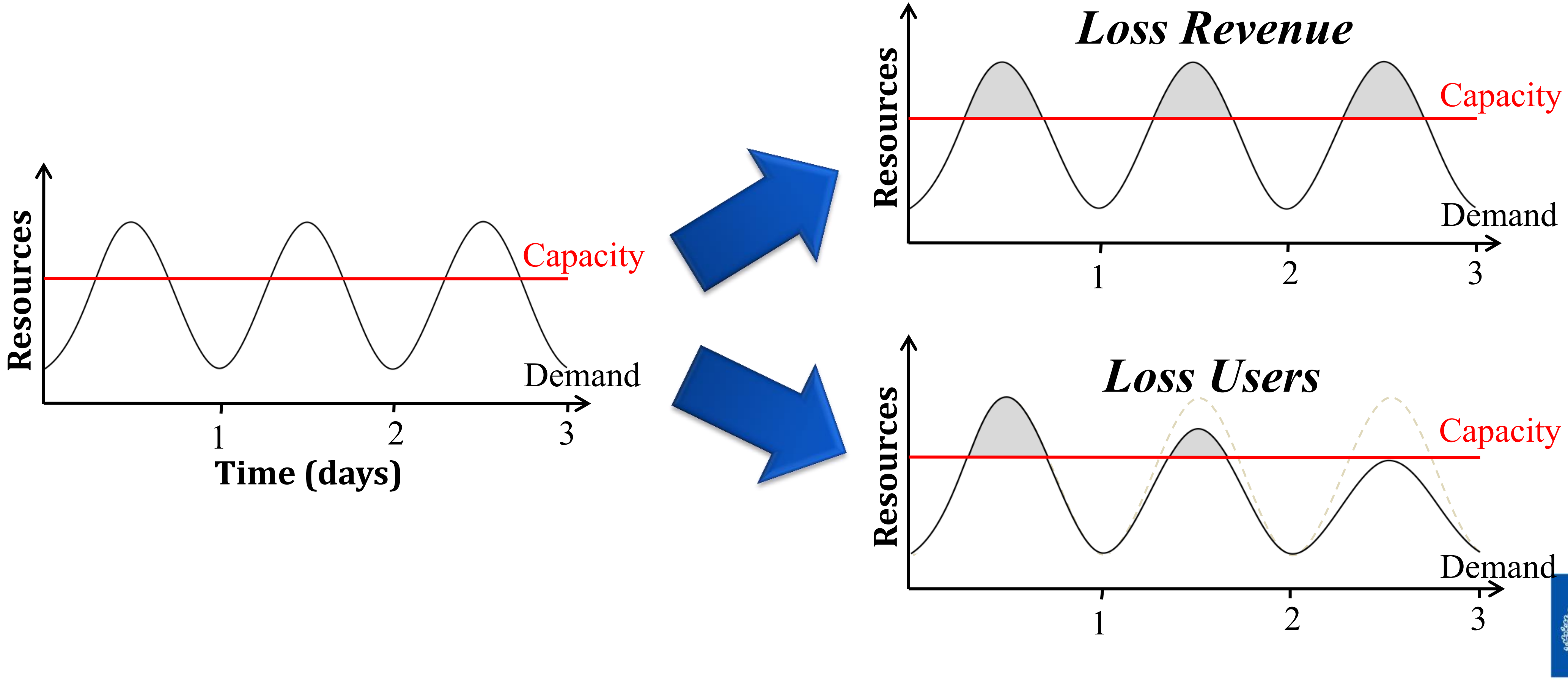
Less and less demand.

Shaded area is unused capability.

Shaded area represents requests not served.

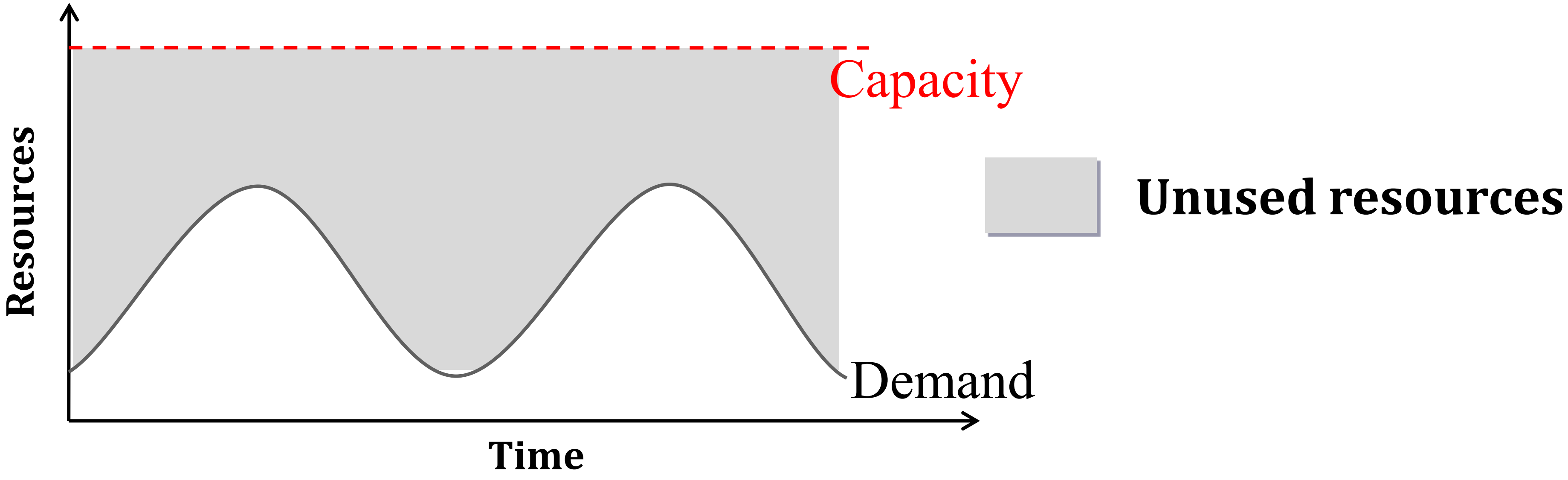
Dynamic Provisioning

- In traditional computing model, two common problems :
  - Underestimate system utilization which result in under provision



Dynamic Provisioning

- Overestimate system utilization which result in low utilization



- How to solve this problem ??
  - Dynamically provision resources

Real world estimates

- Average server utilization is 5% to 20%.
- Peak workload exceeds the average by factors of 2 to 10.
- Users provision for the peak.
- Peak loads may occur based on the time of day or based on other factors (e.g. photo sharing after the holidays, drop/add within two weeks of start of term, etc.)

Dynamic Provisioning

- Cloud computing enables server computing instances to be provisioned or deployed from a administrative console or client application by the server administrator, network administrator, or any other enabled user.

- *Self-managing* systems perform dynamic provisioning on behalf of a user or administrator in order to ensure quality of service (QoS) contracts are not broken and/or to meet some policy objectives.

Multi-tenancy for efficient use of hardware

- Image a single physical machine that is currently running 10 virtual machines (VMs), where each VM running has 5 active java programs.

- Assuming no virtualized application server, how many JVMs processes are running on the physical machine?

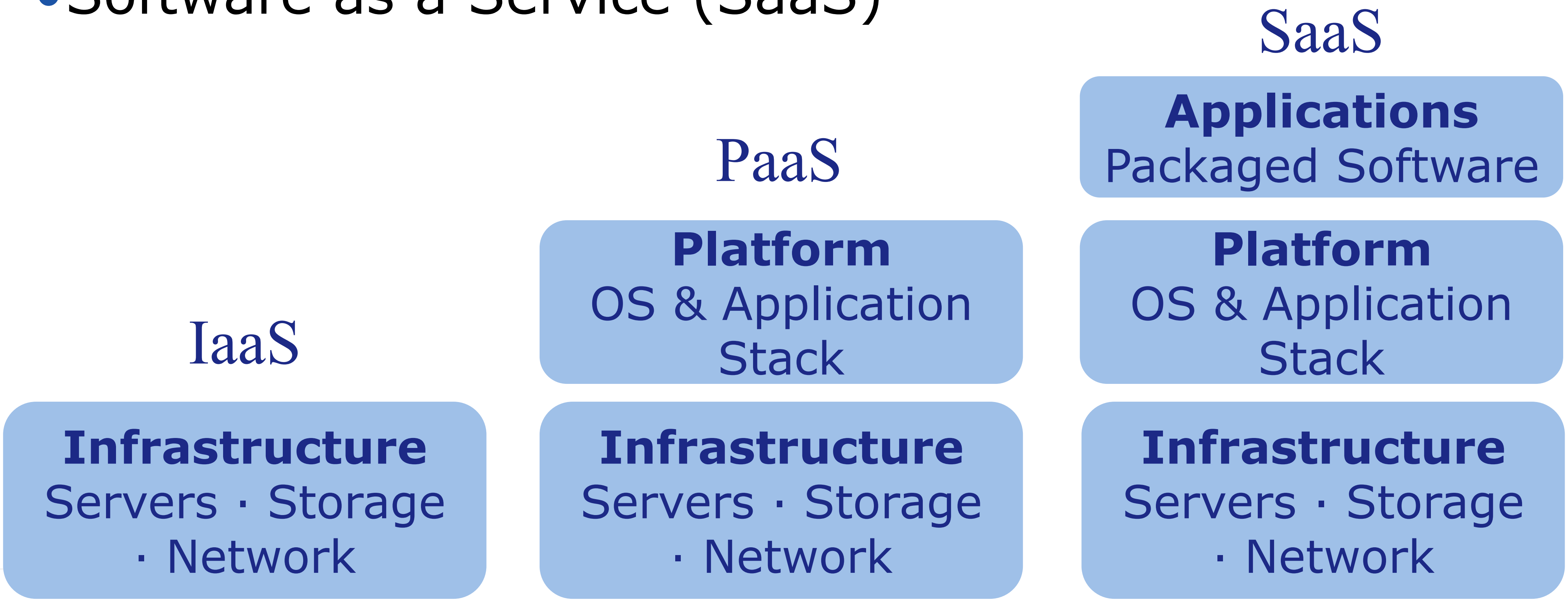
Multi-tenancy

- Multi-tenancy is where a single instance of the software runs on a server, serving multiple clients.
  - Think multiple users in a MySQL database
  - Java 8 will support multi-tenancy (many java programs running in the same JVM)

- The software should be able to provide a single service to all customers by setting configurations
  - More efficient use of server resources

IaaS, PaaS and SaaS

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



IaaS

Infrastructure
Servers · Storage · Network

PaaS

Platform
OS & Application Stack

Infrastructure
Servers · Storage · Network

SaaS

Applications
Packaged Software

Platform
OS & Application Stack

Infrastructure
Servers · Storage · Network

Spectrum of Cloud Users

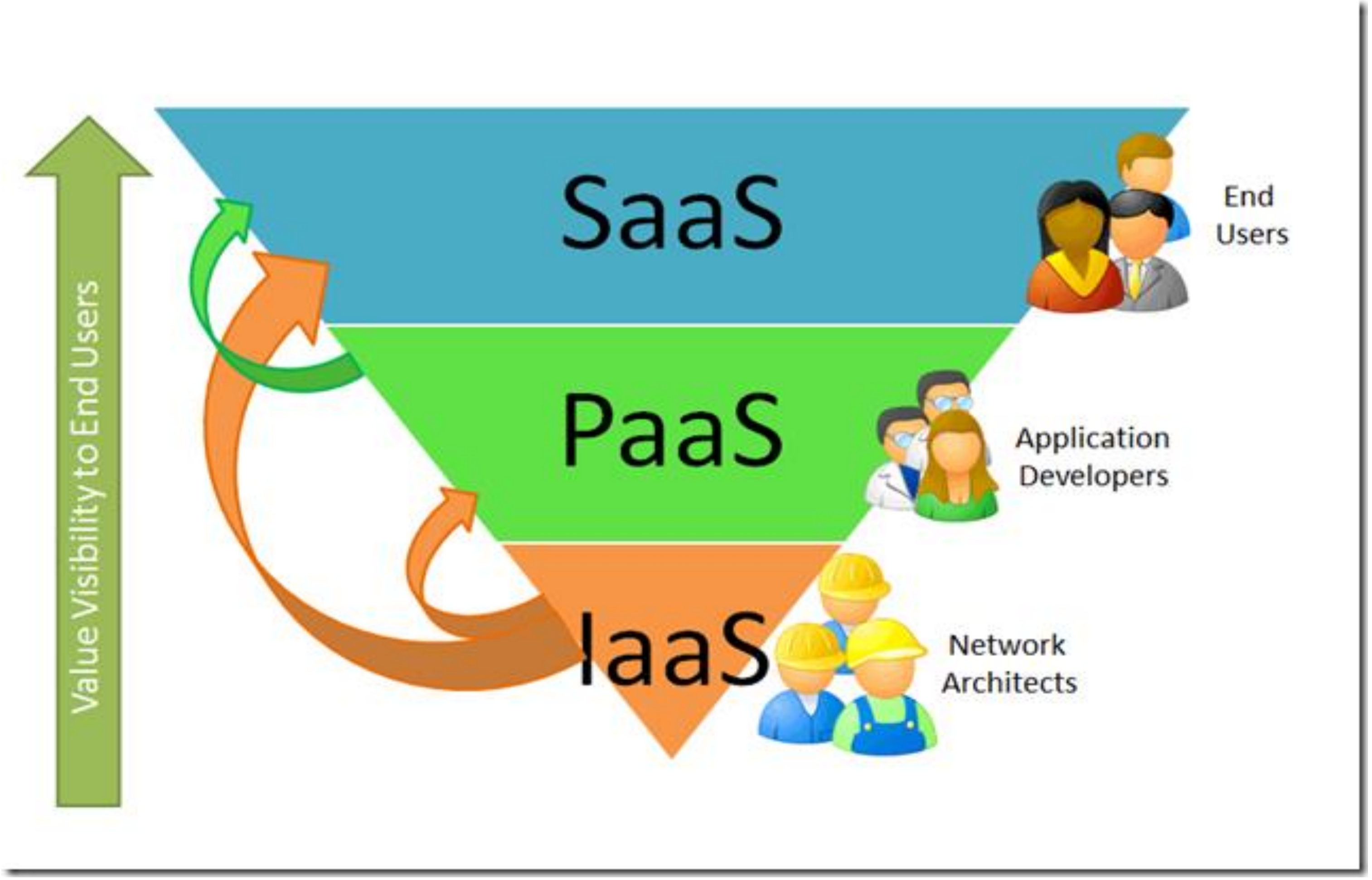
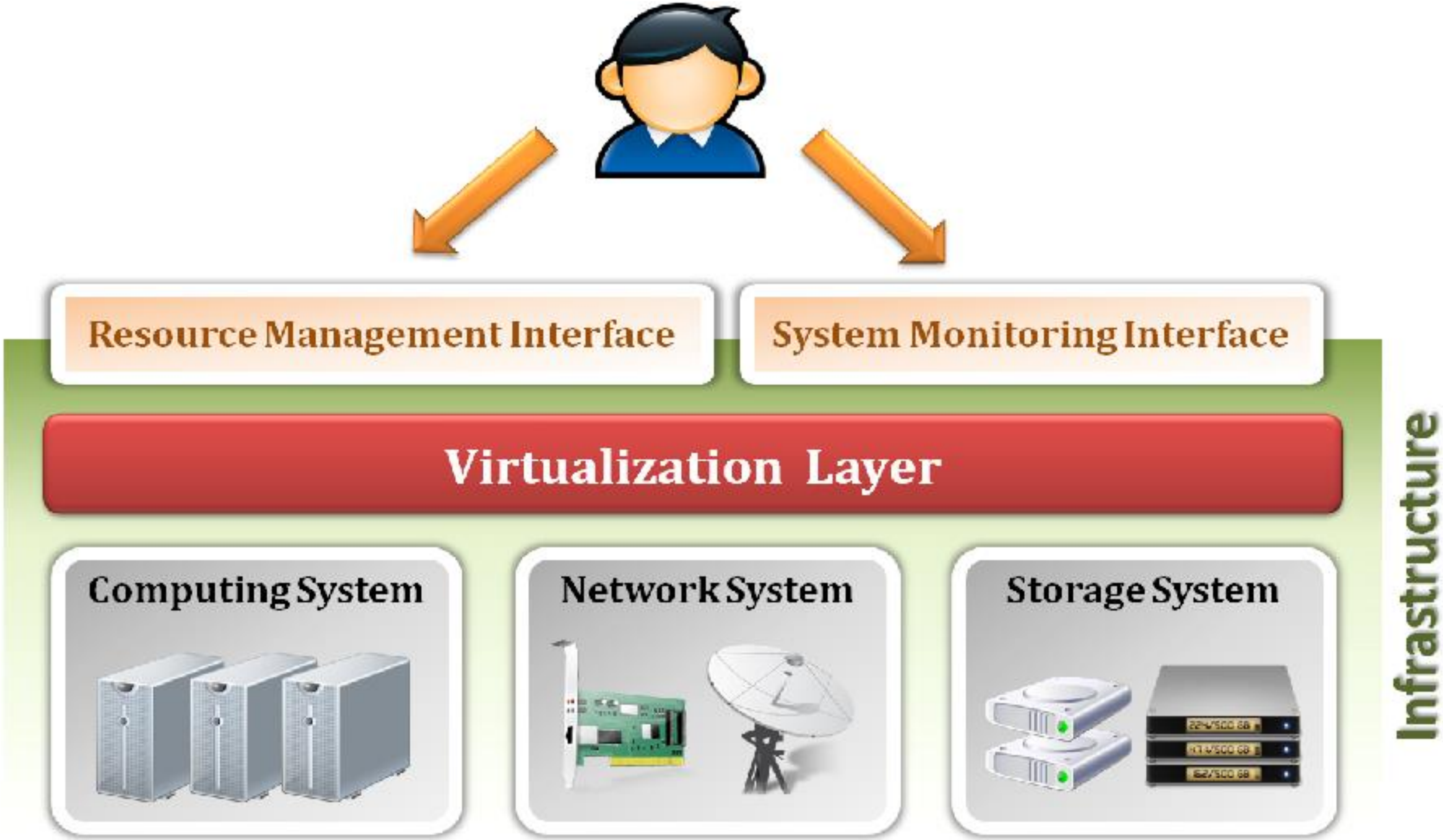


Image credit:

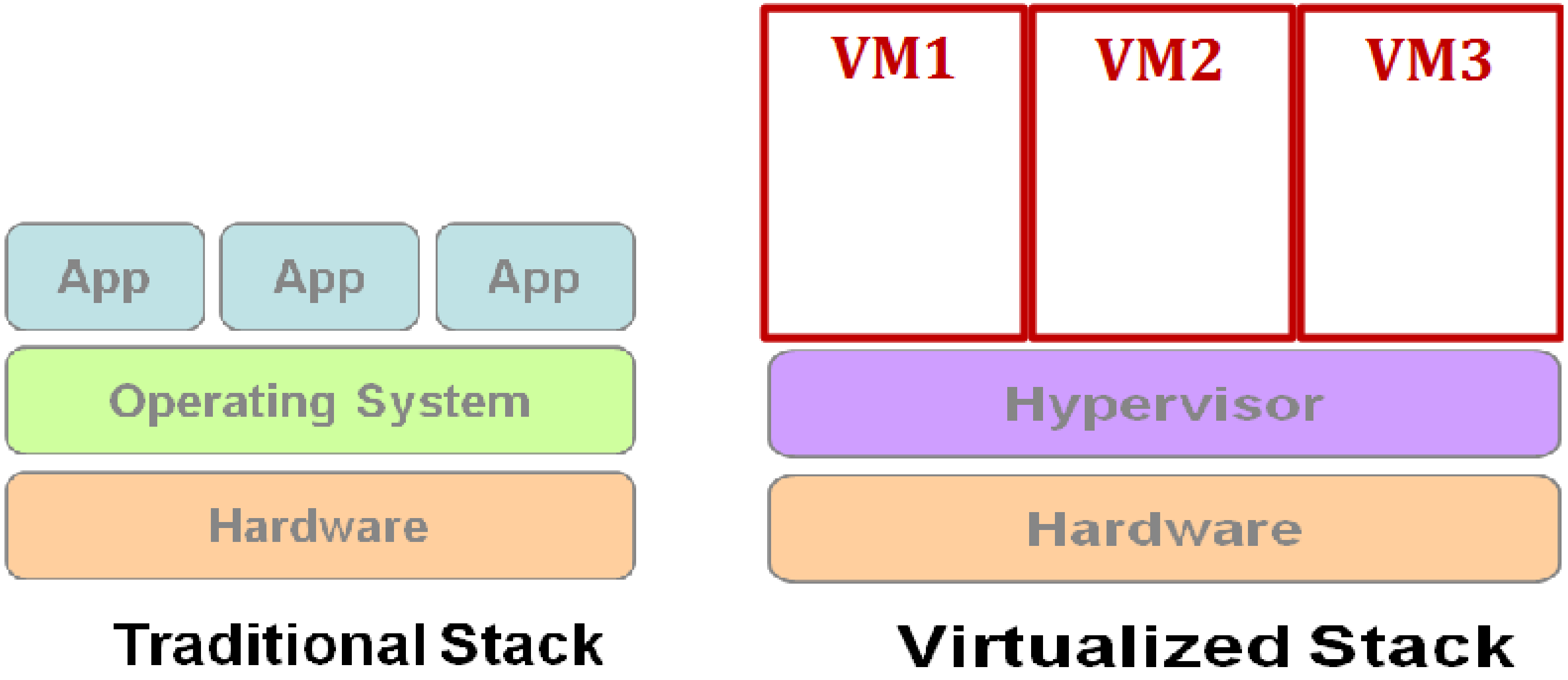
http://blogs.msdn.com/b/seliot/archive/2010/03/04/what-the-heck-is-cloud-computing-another-re-look-with-pretty-pictures.aspx

IaaS - Virtualization



Virtualization

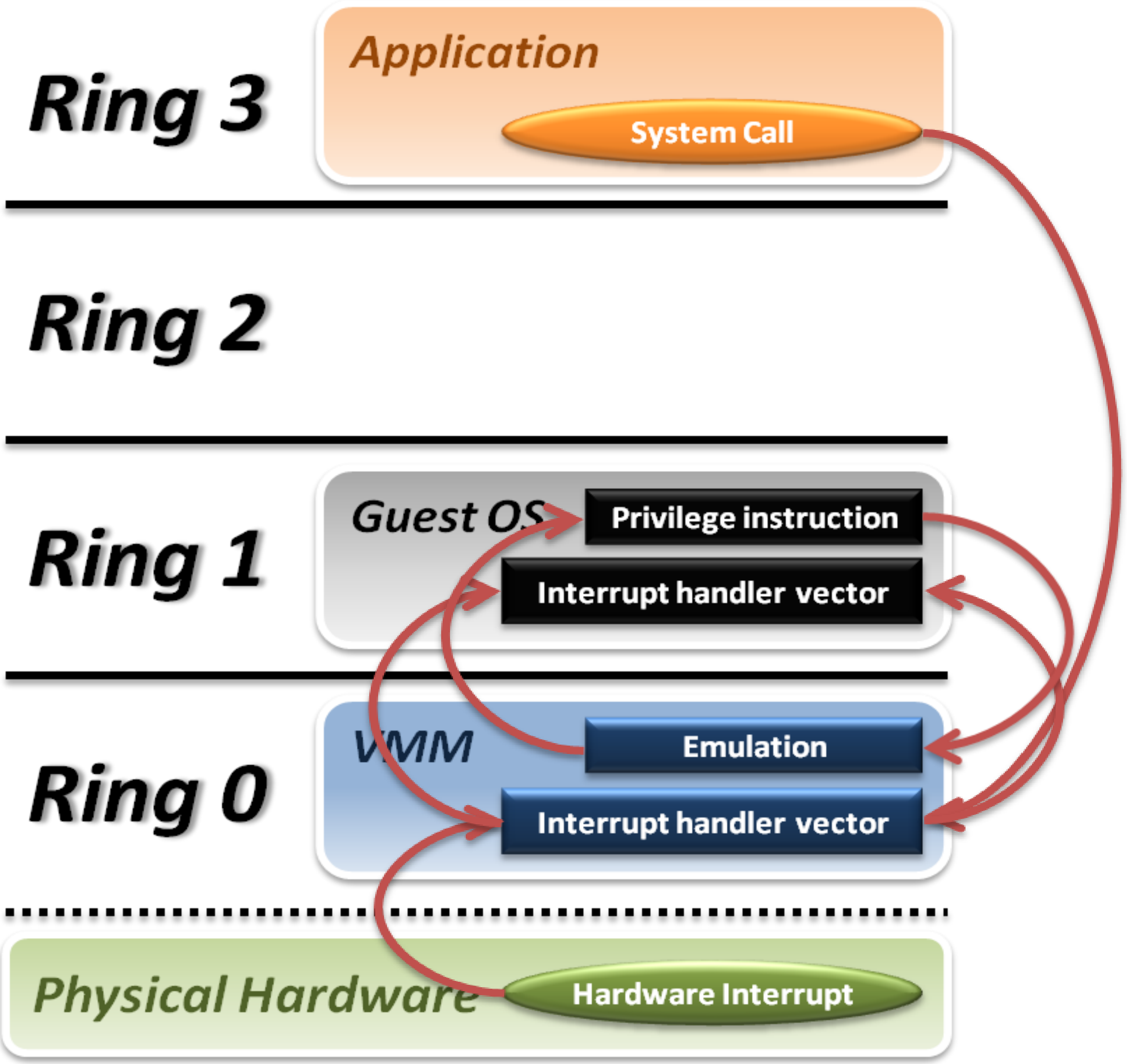
- **Virtualization** is the abstraction of logical resources away from underlying physical resources.
- A **hypervisor** (or **Virtual Machine Monitor** (VMM)) virtualizes a platform’s operating system.
  - The hypervisor manages OS’ as virtual machines (VMs) , enabling multiple OS’ to share the same physical hardware.



Hypervisor’s Trap and Emulate Model

- The hypervisor’s virtualization paradigm is *trap and emulate* :
  - Normal instructions of guest OS
    - run directly on processor in user mode.
  - System Calls
    - CPU will trap to interrupt handler vector of Hypervisor.
    - Hypervisor jump back into guest OS.
  - Hardware Interrupts
    - Hardware makes CPU trap to interrupt handler of Hypervisor.
    - Hypervisor jumps to corresponding interrupt handler of guest OS.
  - Privilege Instructions
    - Running privilege instructions in guest OS will be trapped to Hypervisor for instruction emulation.
    - After emulation, the Hypervisor jumps back to guest OS.

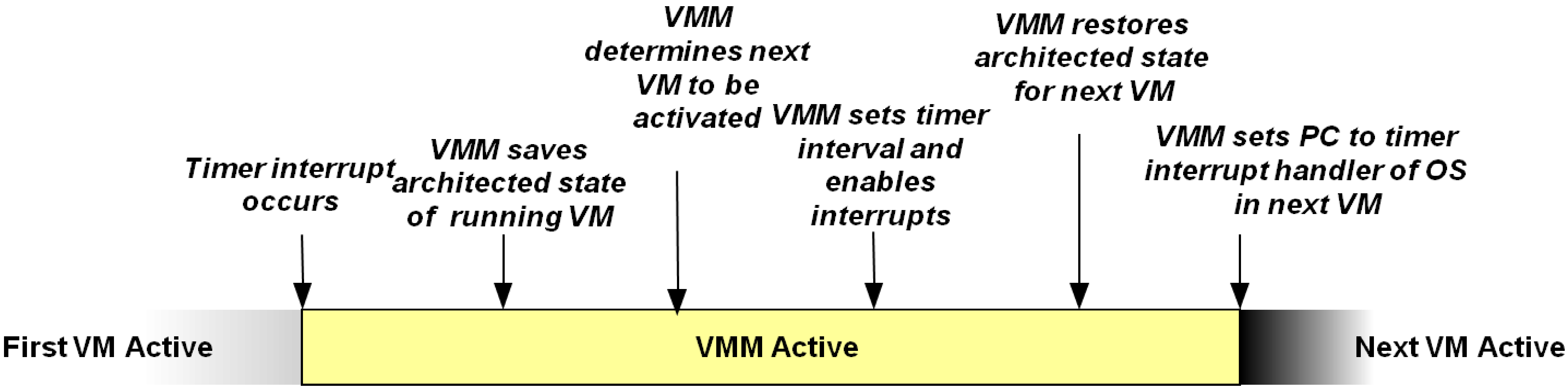
Trap and Emulate Model (VMM=Hypervisor)



VM Context Switching

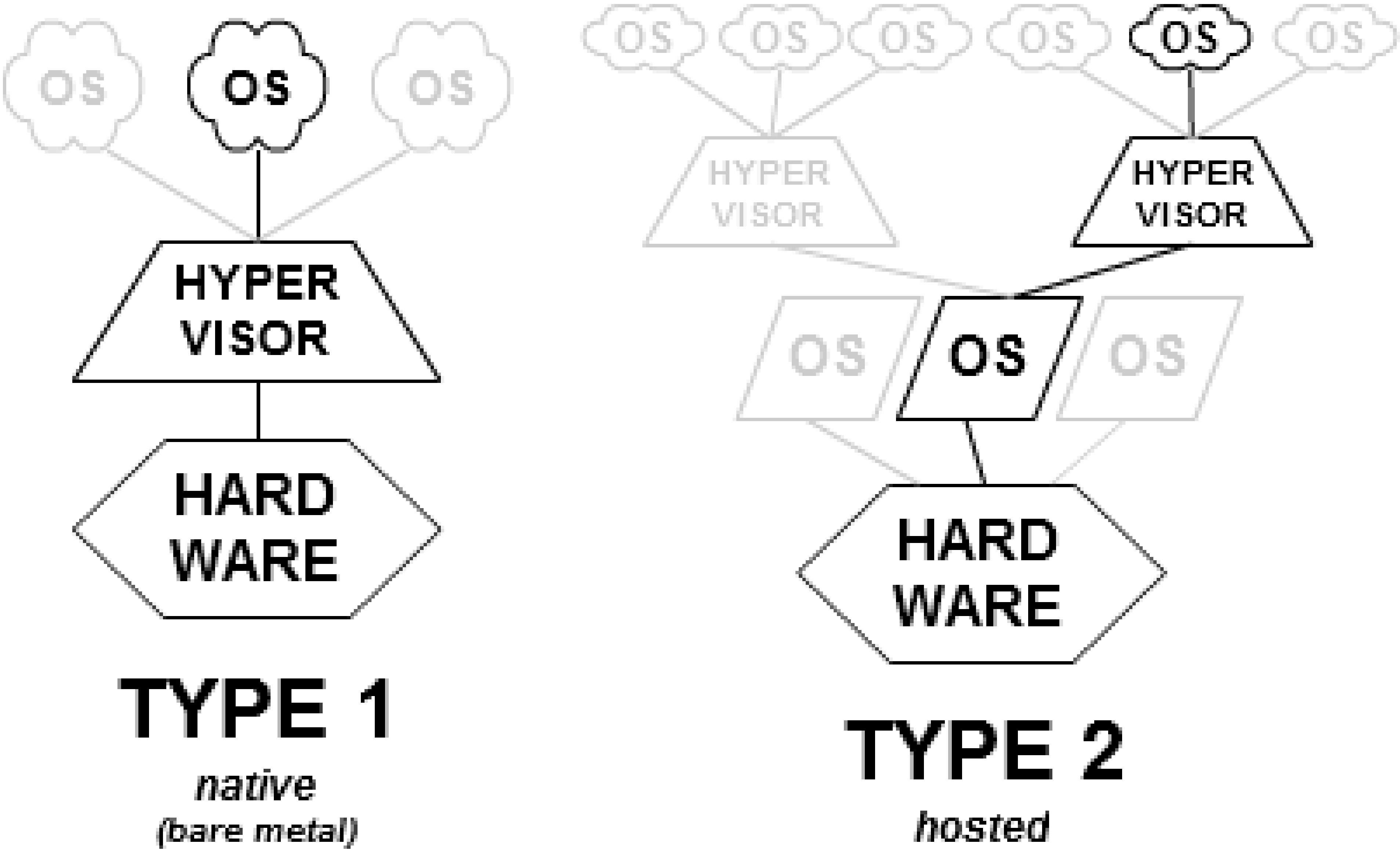
- The hypervisor context switches virtual machines:
  1. Timer Interrupt in running VM.
  2. Context switch to Hypervisor.
  3. Hypervisor saves state of running VM.
  4. Hypervisor determines next VM to execute.
  5. Hypervisor sets timer interrupt.
  6. Hypervisor restores state of next VM.
  7. Hypervisor sets the program counter to timer interrupt handler of next VM.
  8. Next VM active.

VM Context Switching



VMM determines next VM to be activated
VMM restores architected state for next VM
VMM sets timer interval and enables interrupts
VMM saves architected state of running VM
VMM sets PC to timer interrupt handler of OS in next VM
Timer interrupt occurs
First VM Active
VMM Active
Next VM Active

Hypervisor Models



[https://en.wikipedia.org/wiki/Hypervisor]

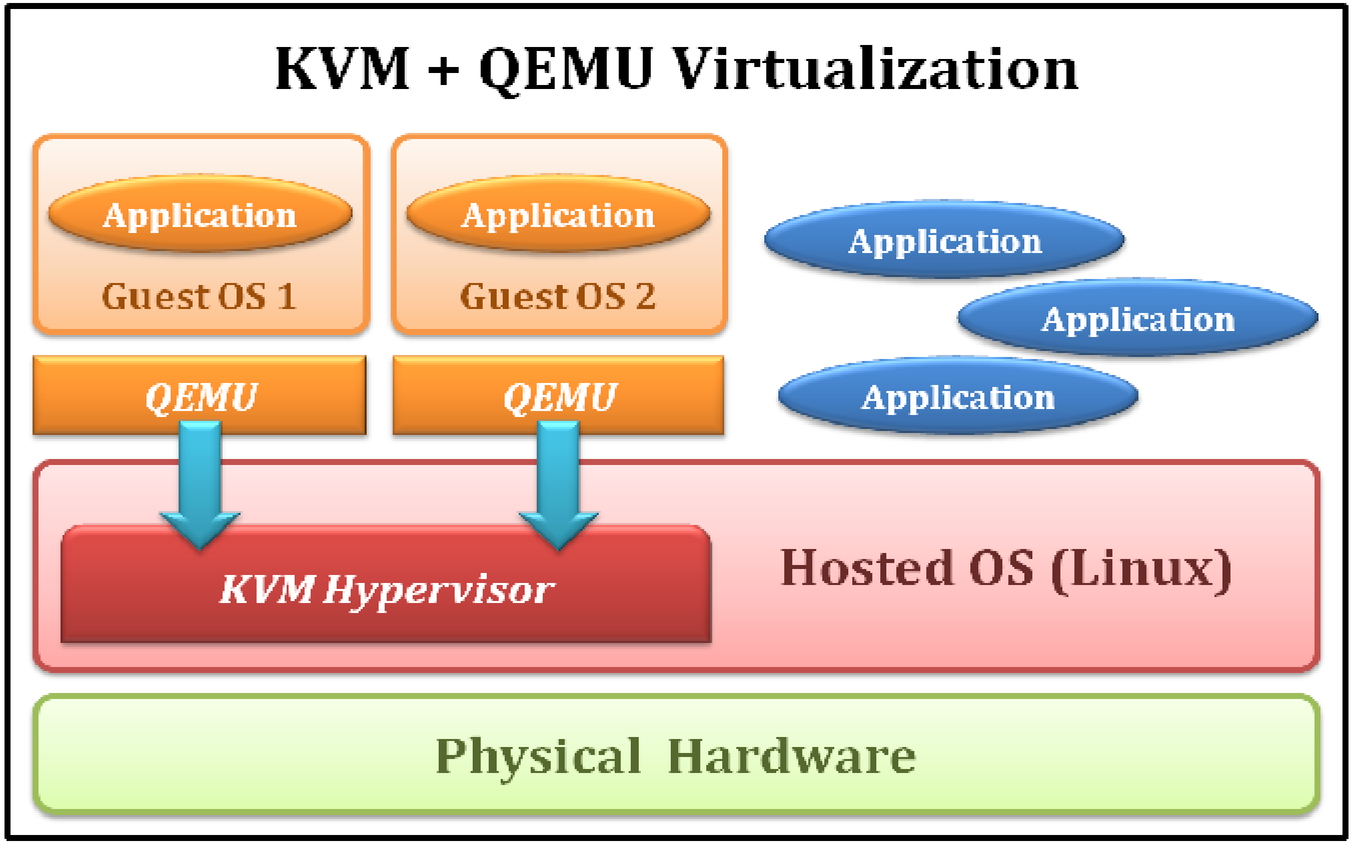
KVM (Kernel-based Virtual Machine)

- VMWare and Xen are the best-known virtualization platforms.

- KVM (Kernel-based Virtual Machine) is an open-source virtualization platform
  - Linux host OS
  - Run multiple virtual machines (Windows, MAC, etc) on your linux box
  - IO is virtualized using a device model in KVM
  - KVM requires a modified QEMU (open-source processor emulator) for its IO virtualization framework.
  - Type 1 Hypervisor, as it is a kernel-level module.

Virtualization using KVM in Linux

- KVM is a loadable kernel module
  - **kvm.ko**
    - provides the core virtualization infrastructure
  - **kvm-intel.ko / kvm-amd.ko**
    - processor specific modules



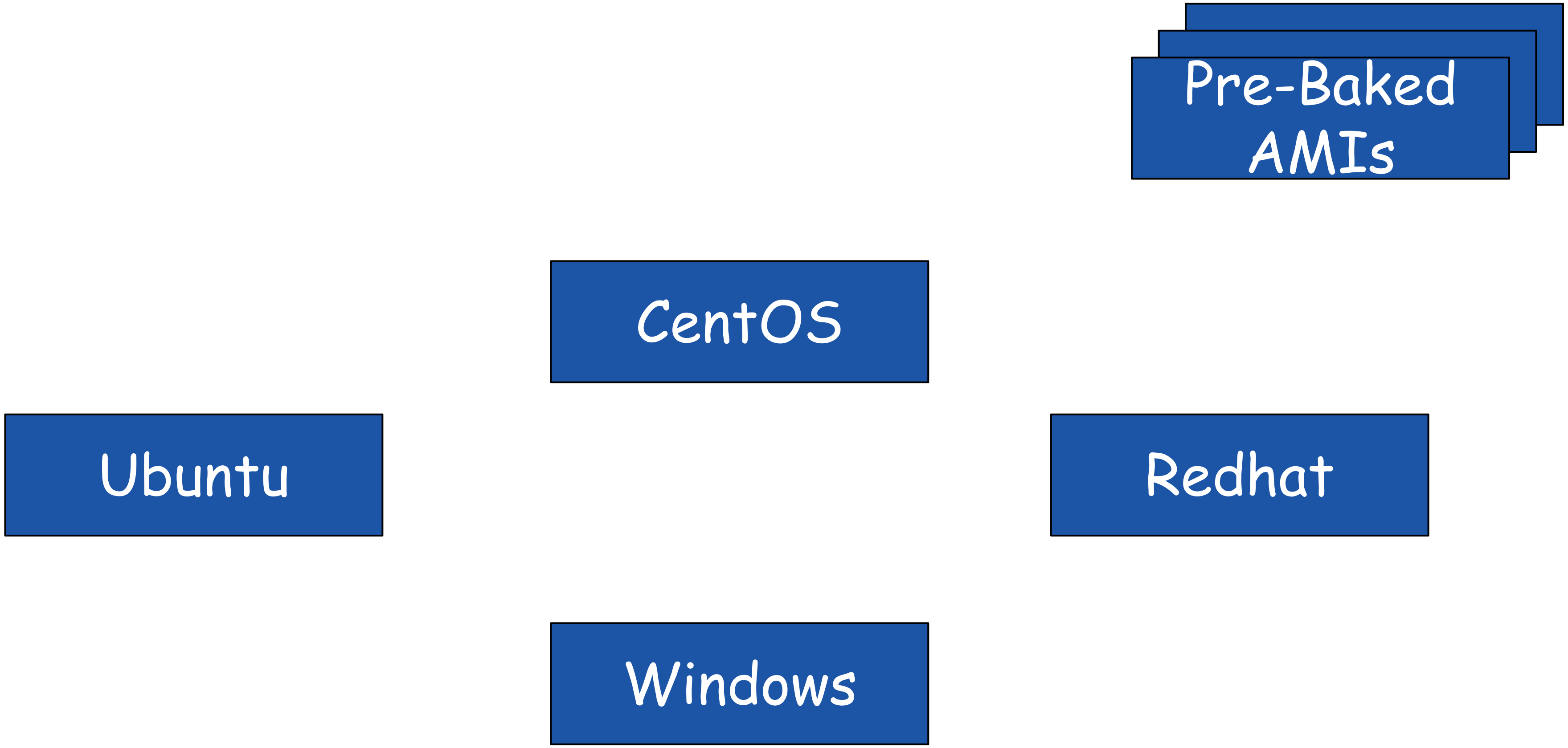
Platform-as-a-Service (PaaS)

IaaS is not Enough

- IaaS provides virtual machines, but it cannot provide *elastic computing* by itself, where services scale up and down to meet user demand.
  - **Dynamic provisioning**

- Existing IaaS’ do not provide support for the sharing middleware platforms among different VMs
  - **Multi-tenancy**

IaaS - what you get



Pre-Baked AMIs

CentOS

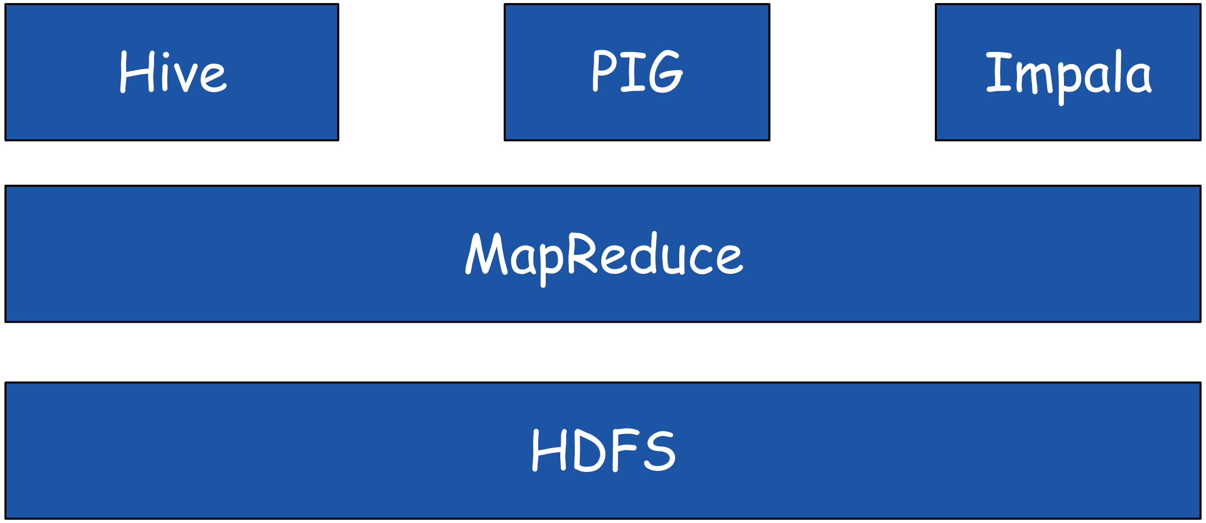
Ubuntu

Redhat

Windows

Infrastructure-as-a-Service

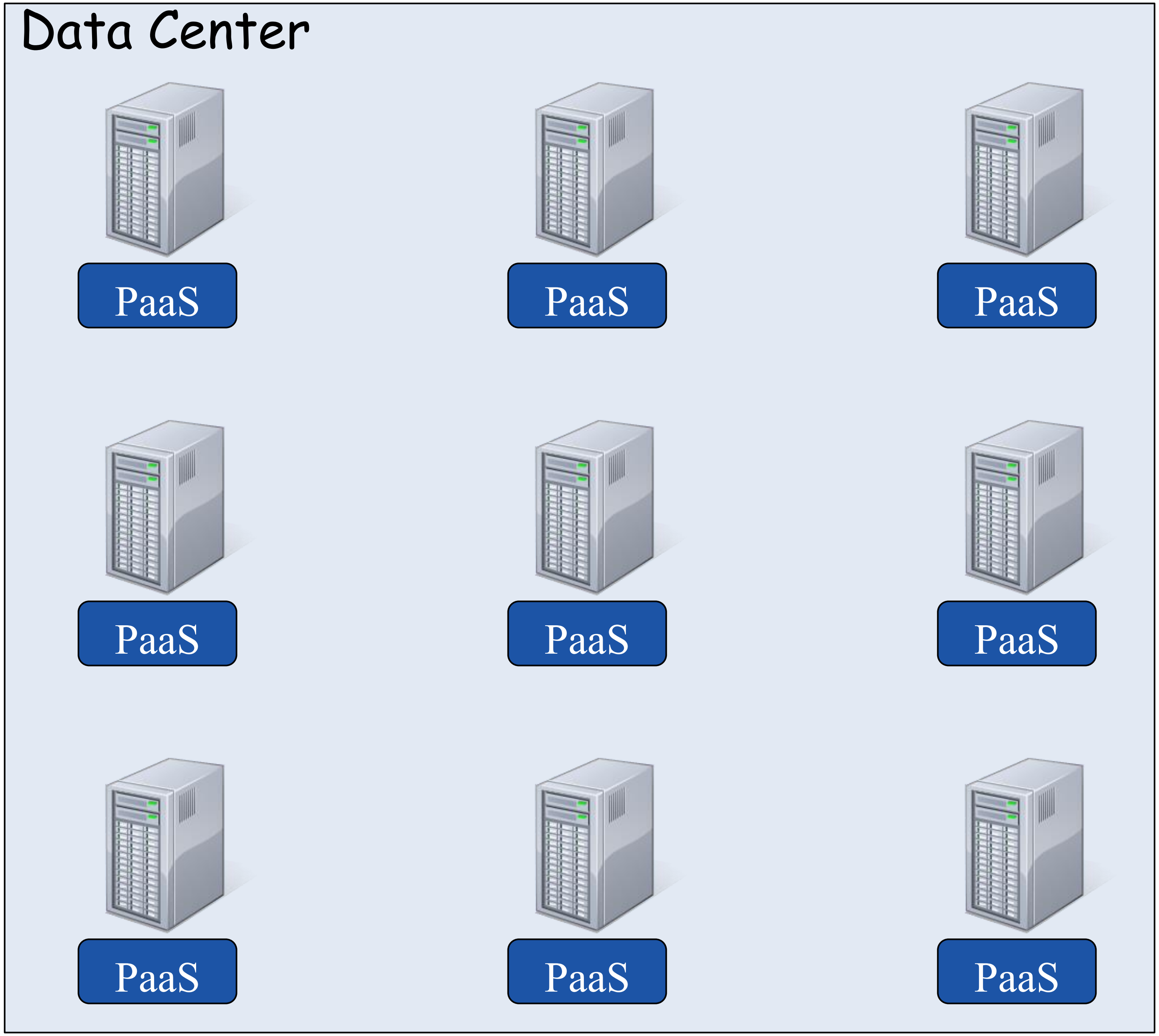
You might prefer this...



Configured stack of servers, dependencies, and firewalls and your app installed.

A Platform-as-a-Service

Running on lots of machines…



Described like this

```
{{
    "name": "master",
      "roles": [ "hadoop_namenode", ”nephele”],
      "instanceNum": 2
 },{
    "name": "worker",
      "roles": ["tasktracker", "hadoop_datanode”],
      "instanceNum": 10
},{
    "name": "client",
      "roles": ["client", "pig”, ”imp”],
      "instanceNum": 1
}}
```

Platform-as-a-Service (PaaS)

- **Platform as a Service (PaaS)** is a computing platform that abstracts the infrastructure, OS, and middleware to drive developer productivity.
- PaaS leverages **dynamic provisioning**
- PaaS leverages **multi-tenancy**

What type of PaaSes are out there?

Closed PaaS

- A **closed PaaS** provides a fixed set of services you can use. You cannot install your own services.
- They are typically hosted at some IaaS provider.

Closed PaaS	Supported Langs/Services
Heroku	Ruby, Node.js, JVM-langs, Python, SQL-DB, KV-Store
AppFog	PHP, Ruby, Node.js, Python, SQL-DB, KV-Store
AppEngine (Google)	Python, JVM-langs, Go
AWS Beanstalk, RightScale, EngineYard, CloudBees, …	……….

Open PaaS

- An **open PaaS** provides support for you to configure and upgrade the services on your VM after deployment and can be deployed on any cloud (public or private).

CLOUD FOUNDRY™

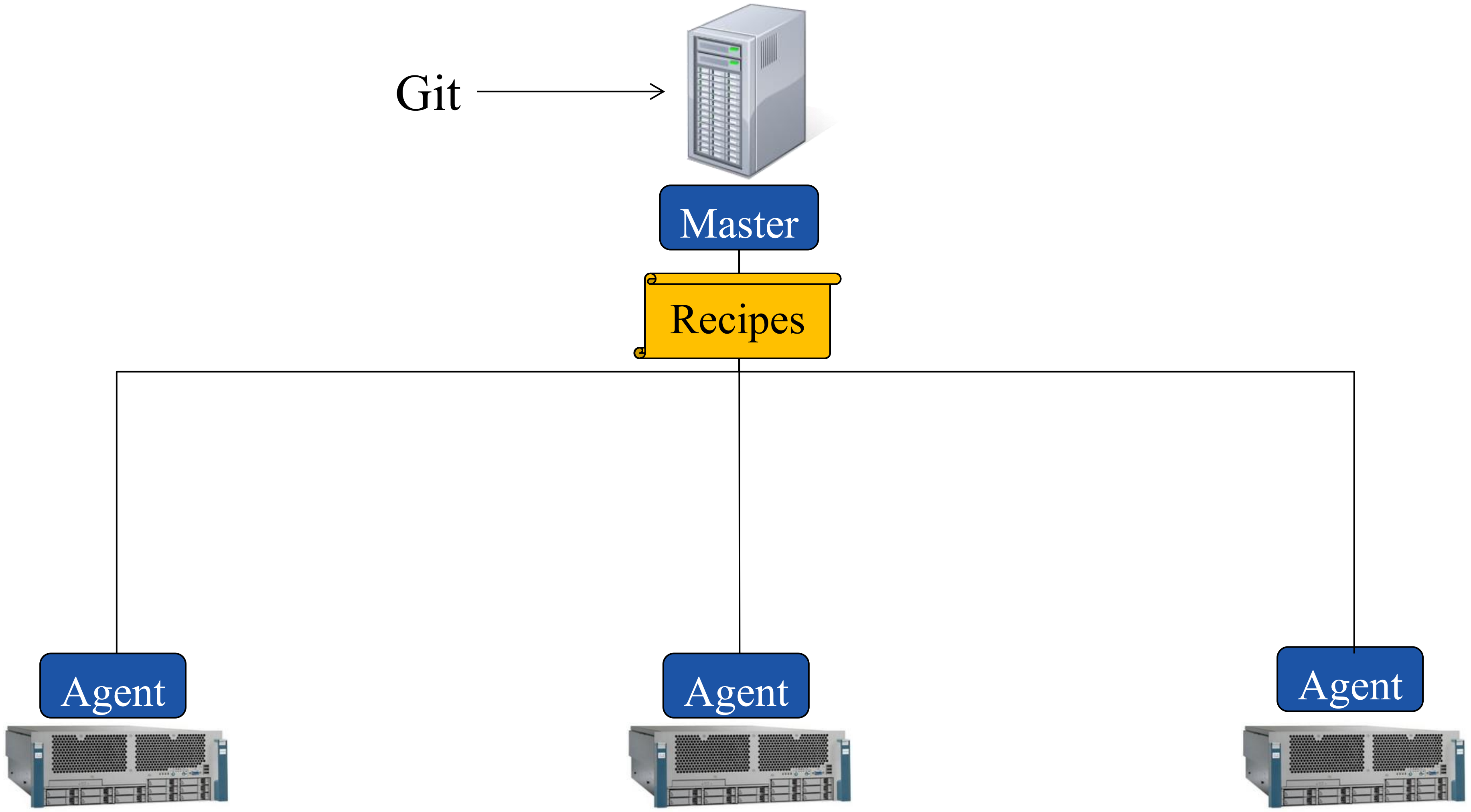
Cloudify

Any App. On Any Cloud. Your Way.

DevOps Platforms for building an Open PaaS

DevOps Platform	Prog. Framework	Orchestration
Pre-baked AMIs	EC2/Jenkins/etc..	Scripts
Chef	Full Ruby DSL	Ironfan DSL
Puppet	Limited Ruby DSL	Mcollective API
Bosh	YML	CloudFoundry
Pallet	Clojure DSL	Clojure DSL
Cloudify	Groovy DSL	Groovy DSL
JClouds	Bash	Java

Chef/Puppet/Bosh/Cloudify



OpenStack Compute REST API Features

- Authentication
- Servers
  - List Servers IPs
  - Create Server
  - Delete Server
  - Reboot Server
- Flavors (hardware config)
  - List Flavors
  - Get Flavor Details
- Images
  - List Images
  - Create Image/Snapshot
  - Get Image Details
  - Delete Image
- Backup Schedules
  - List Backup Schedules
  - Create/Update
  - Disable

http://docs.rackspacecloud.com/servers/api/v1.0/cs-devguide-20110112.pdf

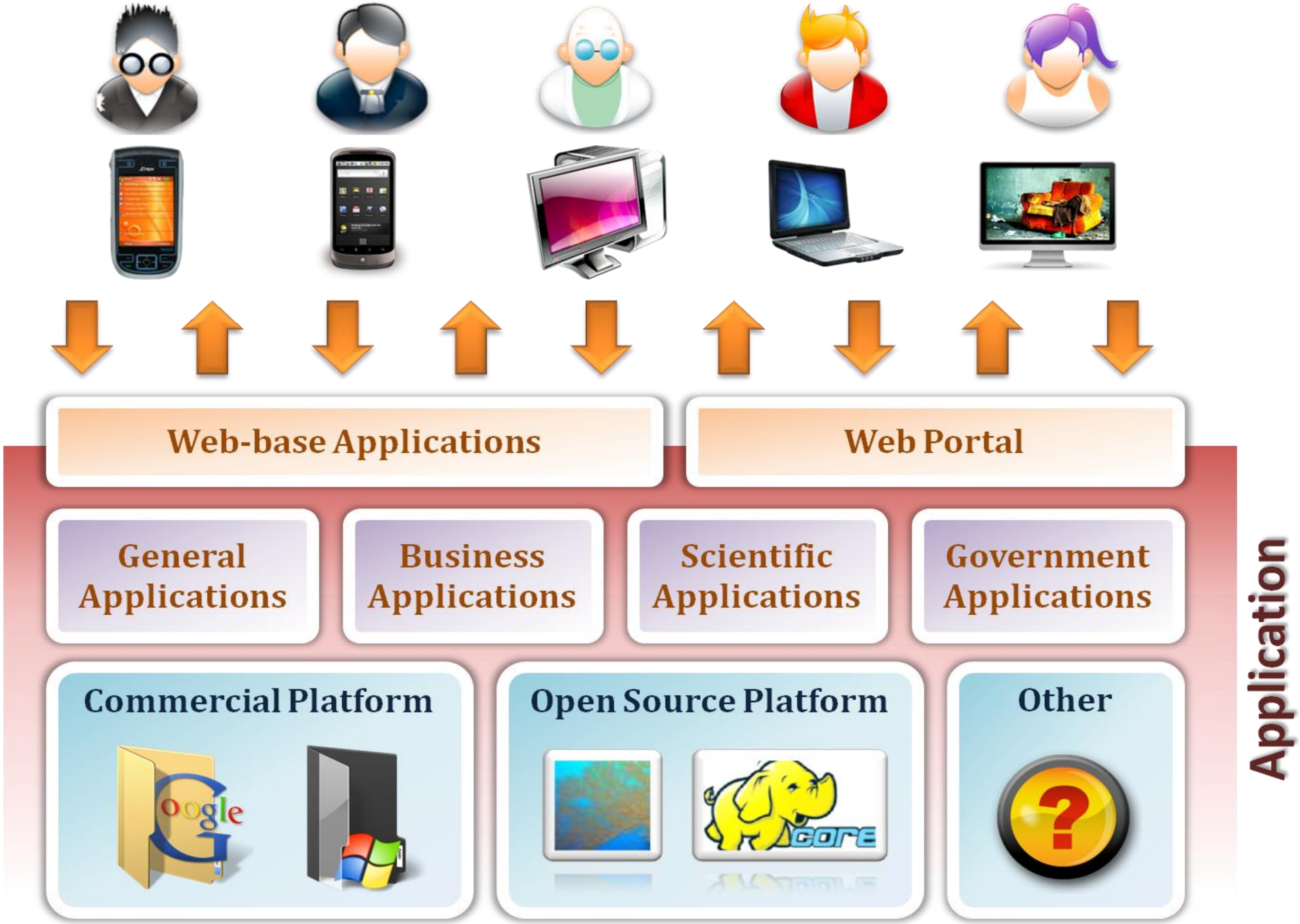
Software-as-a-Service (SaaS)

Software as a Service

- Software as a Service - SaaS
  - Run applications on a provider's on a cloud infrastructure.
  - Applications are accessible from various client devices through a thin client interface such as a web browser.
  - User is oblivious to the underlying cloud infrastructure

- Examples
  - Dropbox
  - Google Apps (e.g., Gmail, Google Docs, Google sites,..)
  - SalesForce.com

Software as a Service



Web-base Applications

Web Portal

General Applications

Business Applications

Scientific Applications

Government Applications

Commercial Platform

Open Source Platform

Other

Application

Deployment Model

- There are four primary cloud deployment models :
  - Public Cloud
  - Private Cloud
  - Community Cloud
  - Hybrid Cloud

Public Clouds

- Public clouds are owned by cloud service providers who charge for the use of cloud resources.
- Basic characteristics:
  - Homogeneous infrastructure, Common policies
  - Shared resources and multi-tenancy
  - Leased or rented infrastructure
  - Economies of scale
- **EC2** (Amazon) Elastic Compute Cloud. General purpose computing.
- **Azure** (Microsoft) General purpose computing on a Microsoft platform.
- **AppEngine** (Google) Build scalable web applications fast. Not general purpose.

Private Clouds

- The cloud infrastructure belongs to and is operated by only one organization.
- Basic characteristics :
  - Heterogeneous infrastructure; Customized policies
  - Dedicated resources
  - In-house infrastructure; End-to-end control
- Examples include:

Public vs. Private Clouds

	Public Cloud	**Private Cloud**
Infrastructure	Homogeneous	Heterogeneous
Policy Model	Common defined	Customized & Tailored
Resource Model	Shared & Multi-tenant	Dedicated
Cost Model	Operational expenditure	Capital expenditure
Economy Model	Large economy of scale	End-to-end control

Other types of Clouds

- 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).

- 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.

Obstacles To Cloud Computing

- Data Lock-in
- No standardized APIs (yet).
  - Jclouds API
- Data Confidentiality/Auditability
- Data transfer bottlenecks/costs
- Performance unpredictability for systems apps

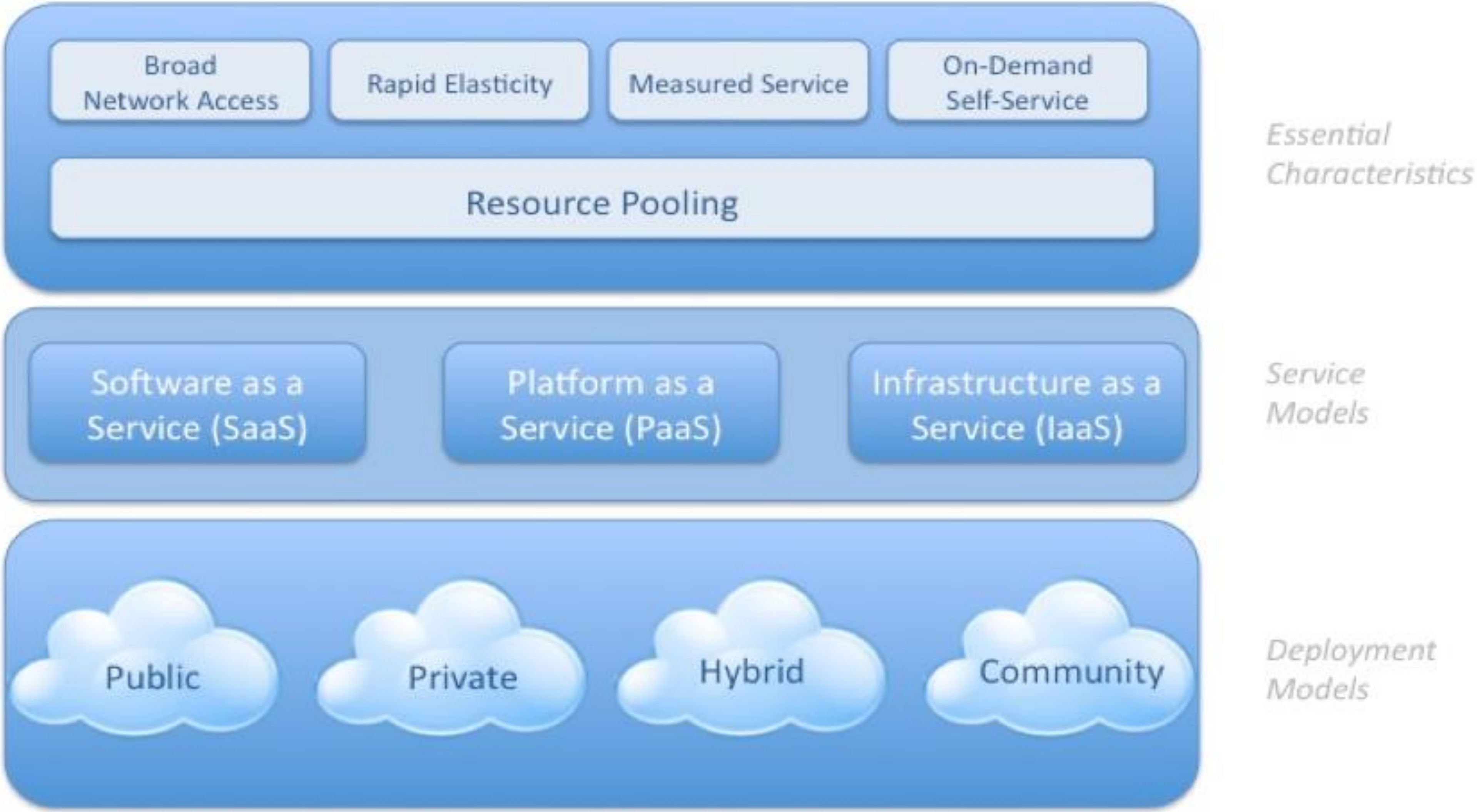
Amazon, Microsoft, Google Cloud Offerings

	Amazon Web Services	**Microsoft Azure**	**Google AppEngine**
Computation model(VM)	x86 ISA via Xen VM	Microsoft CLR VM	Predefined application structure and framework
Storage model	SimpleDB, S3	SQL Data Services	MegaStore/BigTable
Networking model	Declarative specification of IP level topology	Automatic based on programmer’s declarative descriptions of app components	Fixed topology to accommodate 3-tier Web app structure

Summary of Cloud Computing Architecture

Visual Model Of NIST Working Definition Of Cloud Computing

http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html



Broad Network Access

Rapid Elasticity

Measured Service

On-Demand Self-Service

Resource Pooling

Essential Characteristics

Software as a Service (SaaS)

Platform as a Service (PaaS)

Infrastructure as a Service (IaaS)

Service Models

Public

Private

Hybrid

Community

Deployment Models

Conclusions

- Cloud computing has enabled an explosion in large-scale computing services and applications.
- Clouds provide services at three main levels: IaaS, PaaS, SaaS.
- New programming models enable easier development of large-scale applications.

References

Dean et al., MapReduce: simplified data processing on large clusters, Comms of ACM, vol 51(1), 2008.

Armburst et al., “Above the Clouds: A Berkeley View of Cloud Computing”

“Cloud Computing: Principles and Paradigms,” R. Buyya et al. (eds.), Wiley, 2010.

“Cloud Computing: Principles, Systems and Applications,” L. Gillam et al. (eds.) Springer, 2010.

Jeffrey Dean and Sanjay Ghemawat: “MapReduce: Simplified Data Processing on Large Clusters” in *OSDI* 2004

Senjay Ghemawat, : “The Google File System”. *SIGOPS Operating Systems Review* 37(5), 2003

M. Isard et al.: “Dryad: Distributed Data-parallel Programs from Sequential Building Blocks” in *EuroSys* 2007