Introduction to Cloud Computing

Cloud Computing I (intro)

15-319, spring 2010

2nd Lecture, Jan 14th

Majd F. Sakr

Lecture Motivation…

- **General overview on cloud computing**
  - What is cloud computing
    - Services
    - Types
  - Advantages and disadvantages
  - Enabling technologies
  - An example infrastructure

Lecture Outline

- What is Cloud?
- What is Cloud Computing?
- Cloud Computing Services
- History of Cloud Computing
- Why Cloud Computing
- Drawbacks of Cloud Computing
- Types of Clouds

A Cloud is …

- **Datacenter hardware and software** that the vendors use to offer the computing resources and services

Cloud Computing

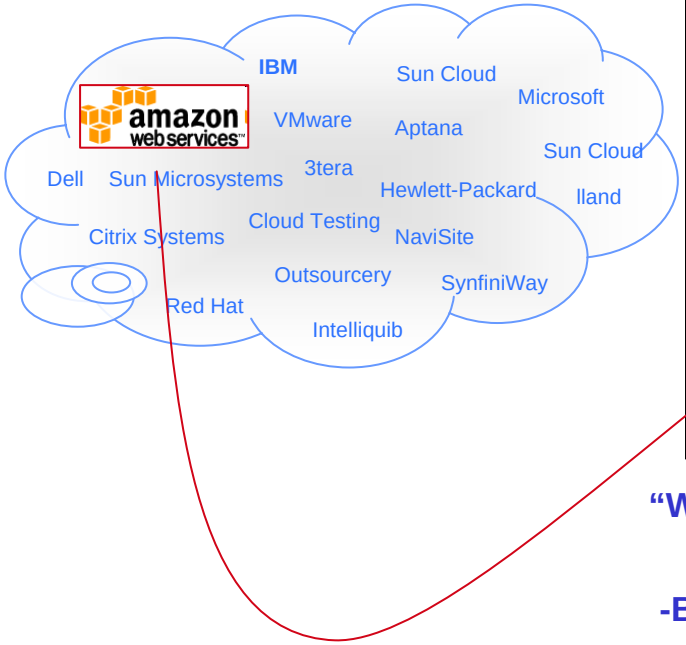
- **Represents both the cloud & the provided services**
- **Why call it “cloud computing”?**
  - Some say because the computing happens out there "*in the clouds*"

    Wikipedia: "*the term derives from the fact that most technology diagrams depict the Internet or IP availability by using a drawing of a cloud.*"

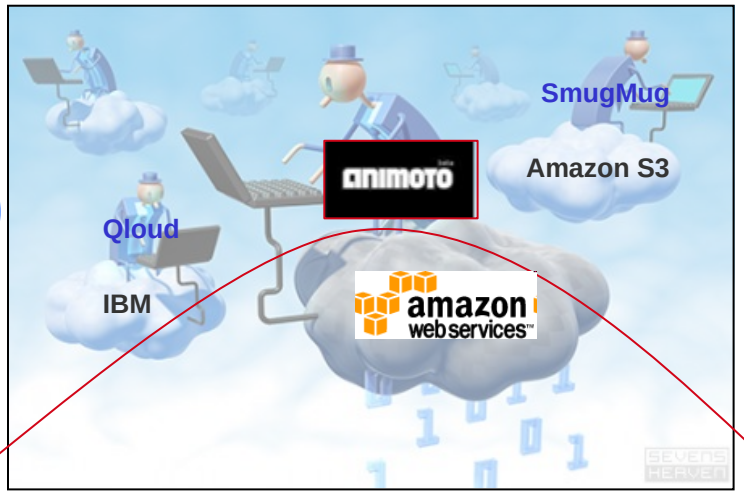
Cloud Computing

- **Who is Who…**

Cloud providers



Cloud Users & Service Providers



Service Users

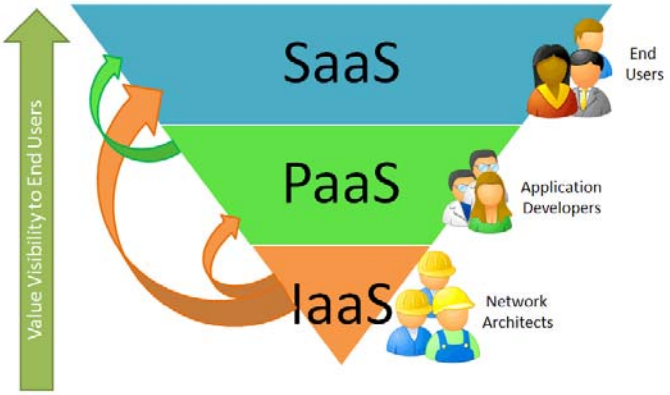
"With Amazon [AWS], on Day One of launch we could scale to the world."

-Brad Jefferson, Co-Founder & CEO, Animoto

"Animoto has partnered with Amazon to leverage multiple offerings in their Web Services (AWS) platform which, in conjunction with Animoto's own render farm, constitutes the Animoto web infrastructure."

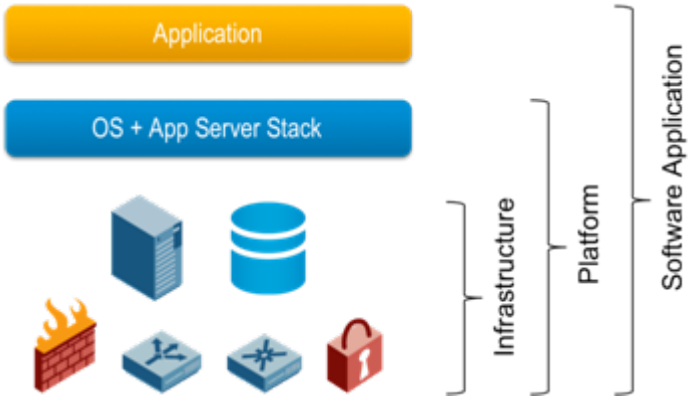
Users use it to produce video pieces from their photos, video clips and music.

Cloud Computing Services

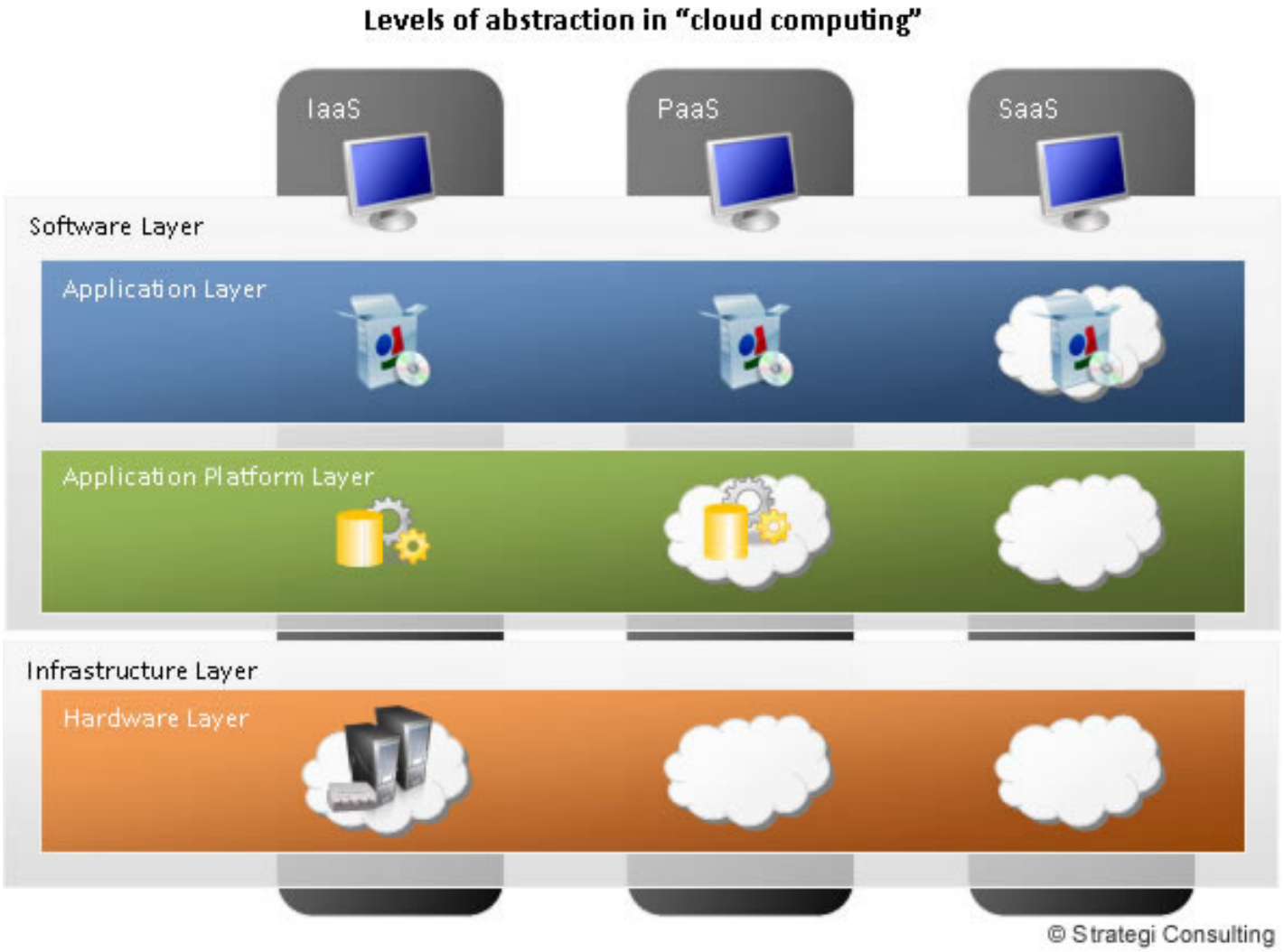


Three basic services:

- **Software as a Service (SAAS) model**
  - Apps through browser
- **Platform as a Service (PAAS) model**
  - Delivery of a computing platform for custom software development as a service
- **Infrastructure as a Service (IAAS) model**
  - Deliver of computer infrastructure as a service
- **XAAS, the list continues to grow…**



Cloud Services (XaaS)



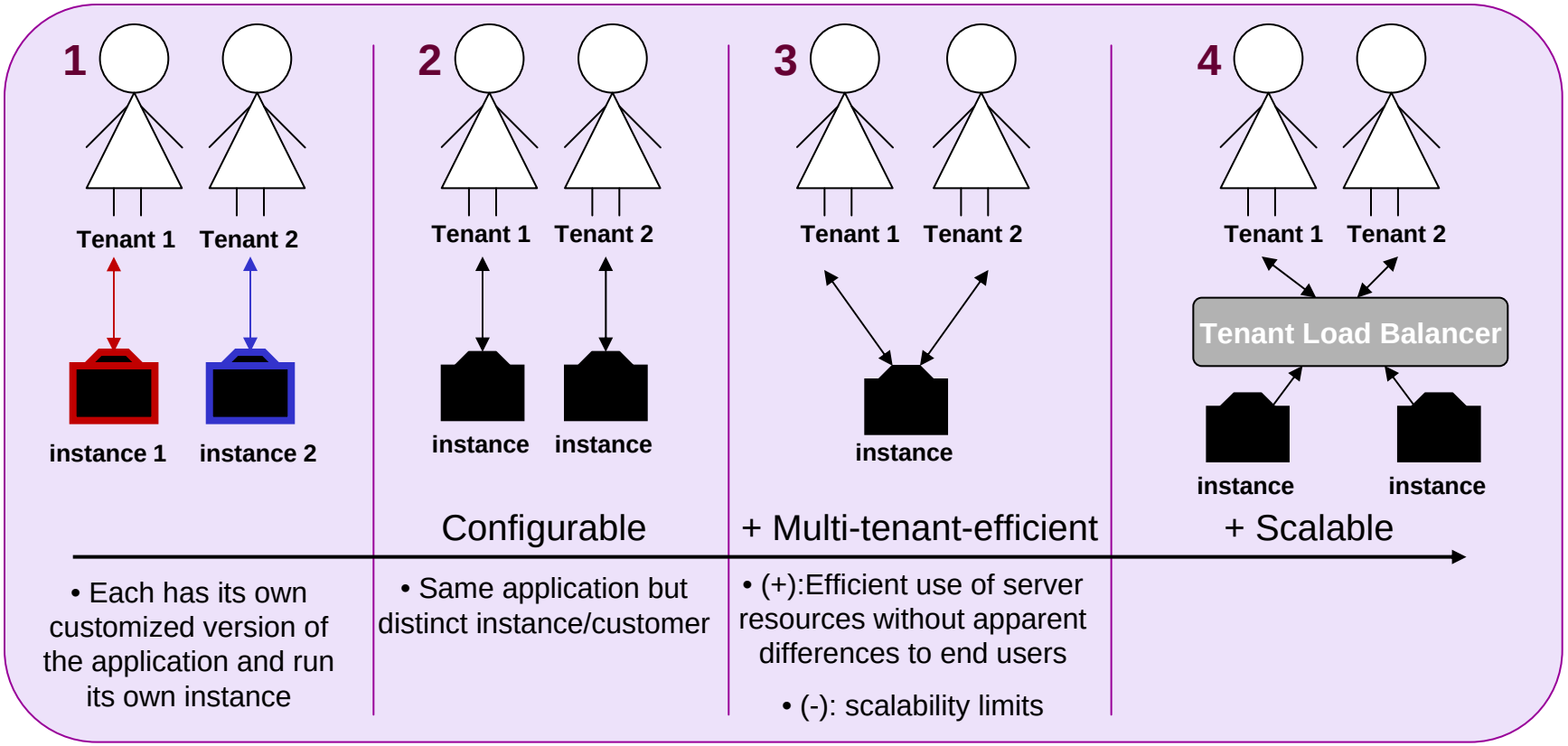
SaaS (1/3)

- **Started around 1999**
- **Application is licensed to a customer as a service on demand**
- **Software Delivery Model:**
  - Hosted on the vendor's web servers
  - Downloaded at the consumer's device and disabled when on-demand contract is over

SaaS (2/3)

SaaS

- **SaaS architecture/ Maturity levels:**
  - **Distinguishing attributes:** configurability, multi-tenant efficiency, scalability

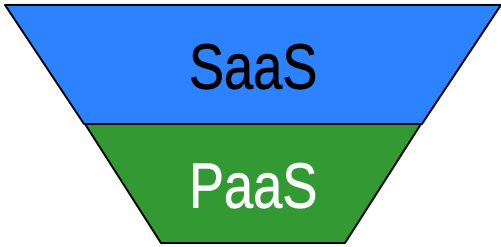


SaaS (3/3)

SaaS

- **Examples**

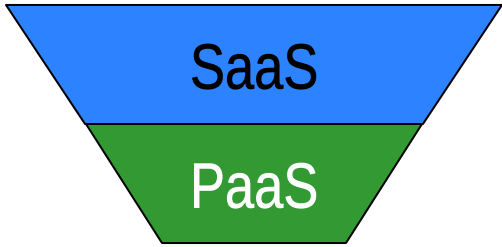
PaaS (1/2)



- **Delivery of an integrated computing platform (to build/test/deploy custom apps) & solution stack as a service.**

- **Deploy your applications & don't worry about buying & managing the underlying hardware and software layers**

PaaS (2/2)



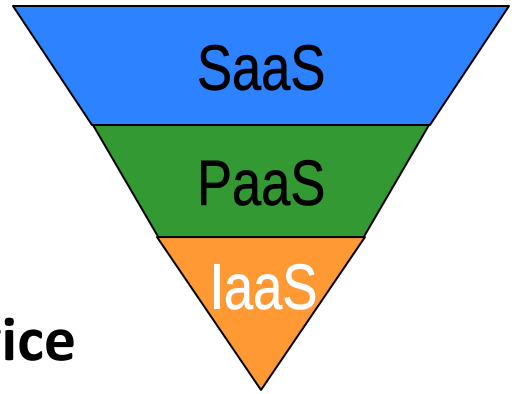
- **Examples**

Google Apps

Etelos

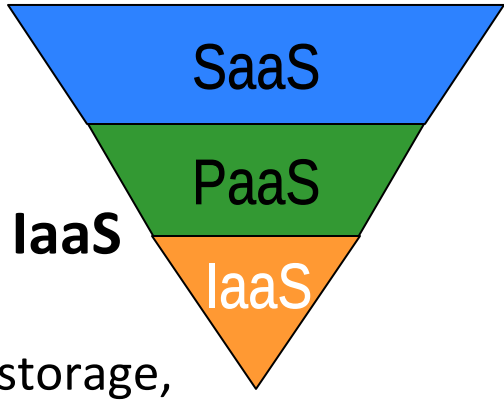
LongJump

IaaS (1/5)

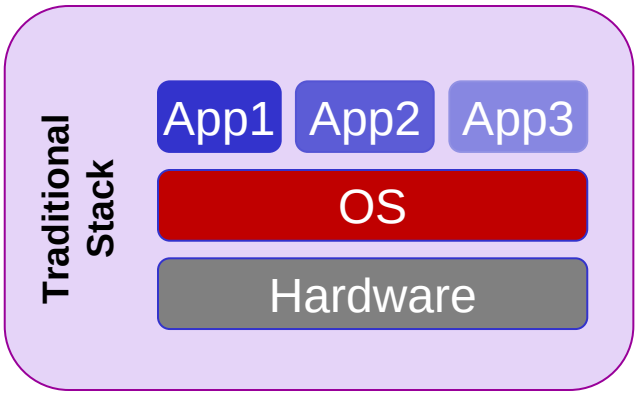


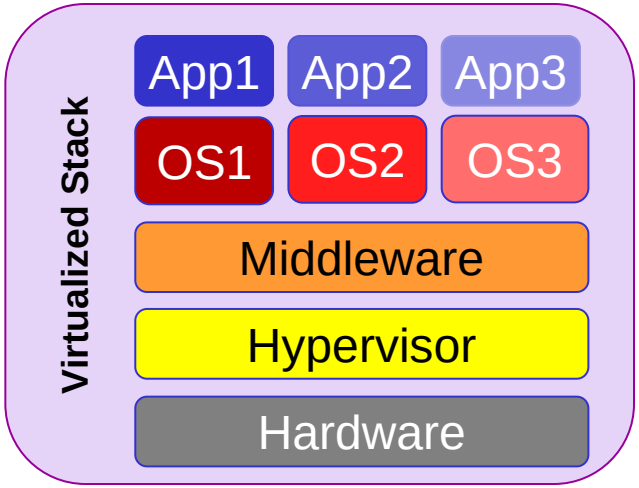
- **Delivery of computer infrastructure (typically platform virtualization environment) as a service**
- **Buy resources**
  - Servers
  - Software
  - Data center space
  - Network equipment as fully outsourced services
- **Example:**

IaaS (2/5)

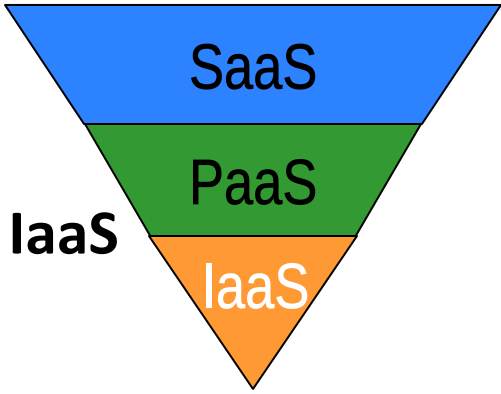


- **Virtualization Technology is a major enabler of IaaS**
  - It's a path to share IT resource pools: Web servers, storage, data, network, software and databases.
  - Higher utilization rates

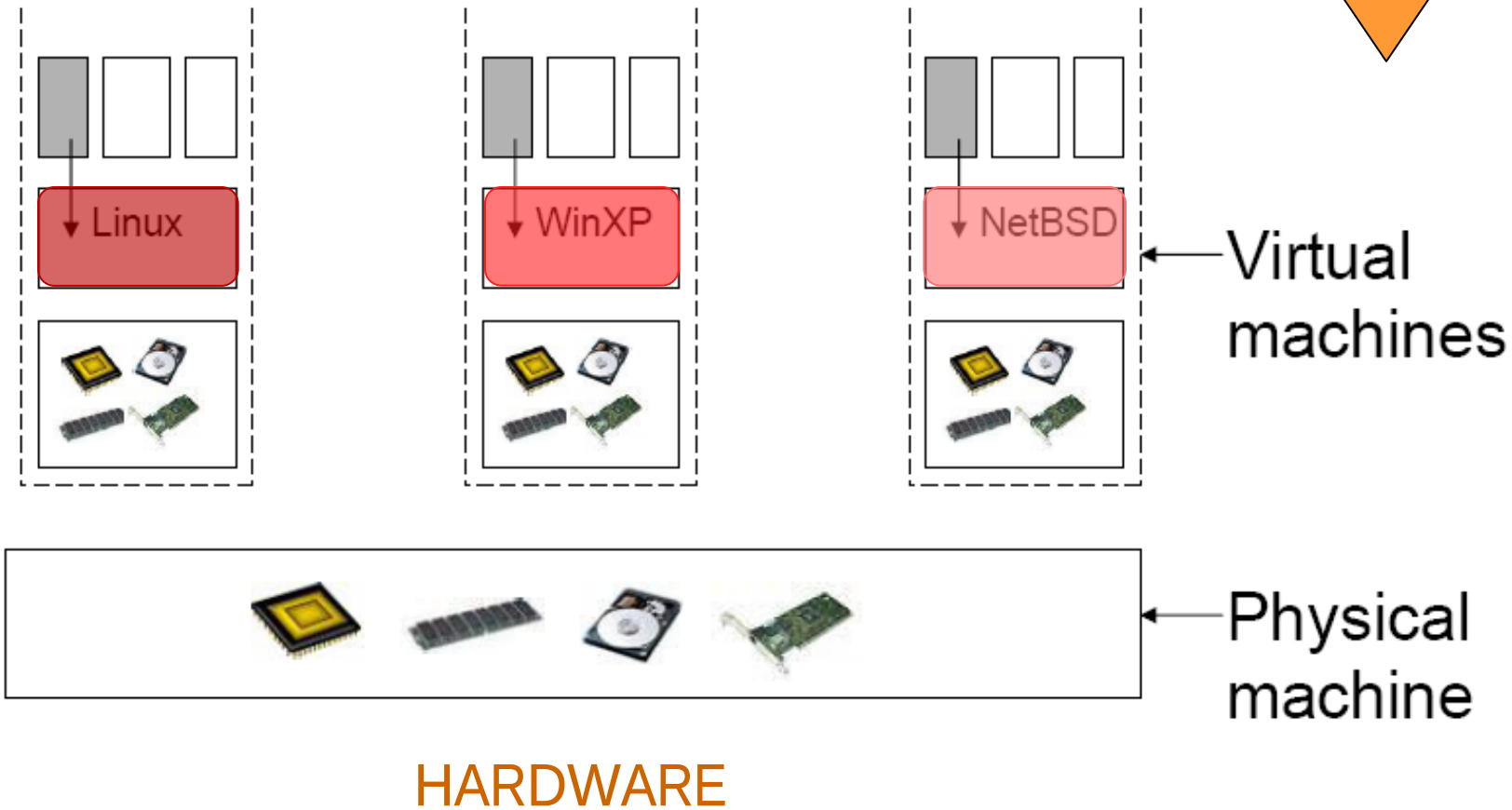




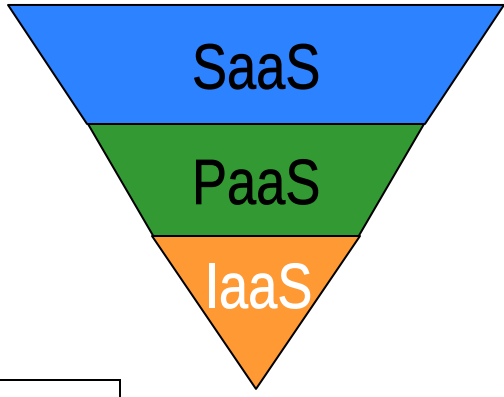
IaaS (3/5)



- **Virtualization Technology is a major enabler of IaaS**

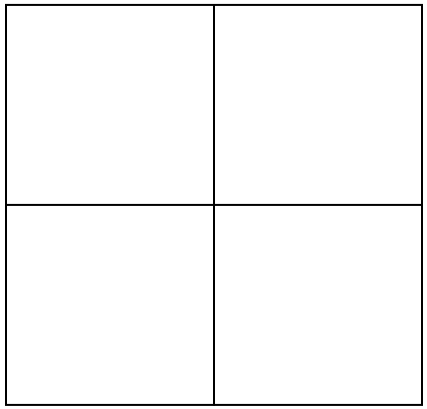


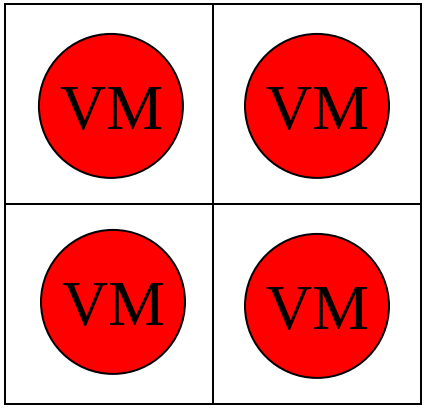
IaaS (4/5)

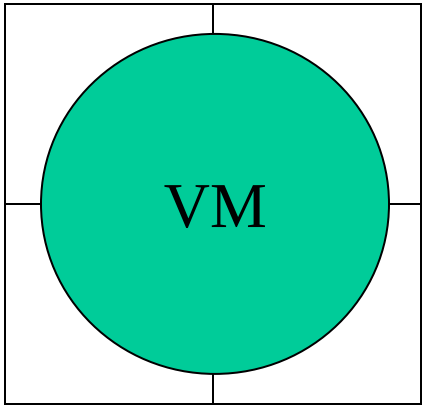


- **Granularity of VMs**
  - Multi-core processors

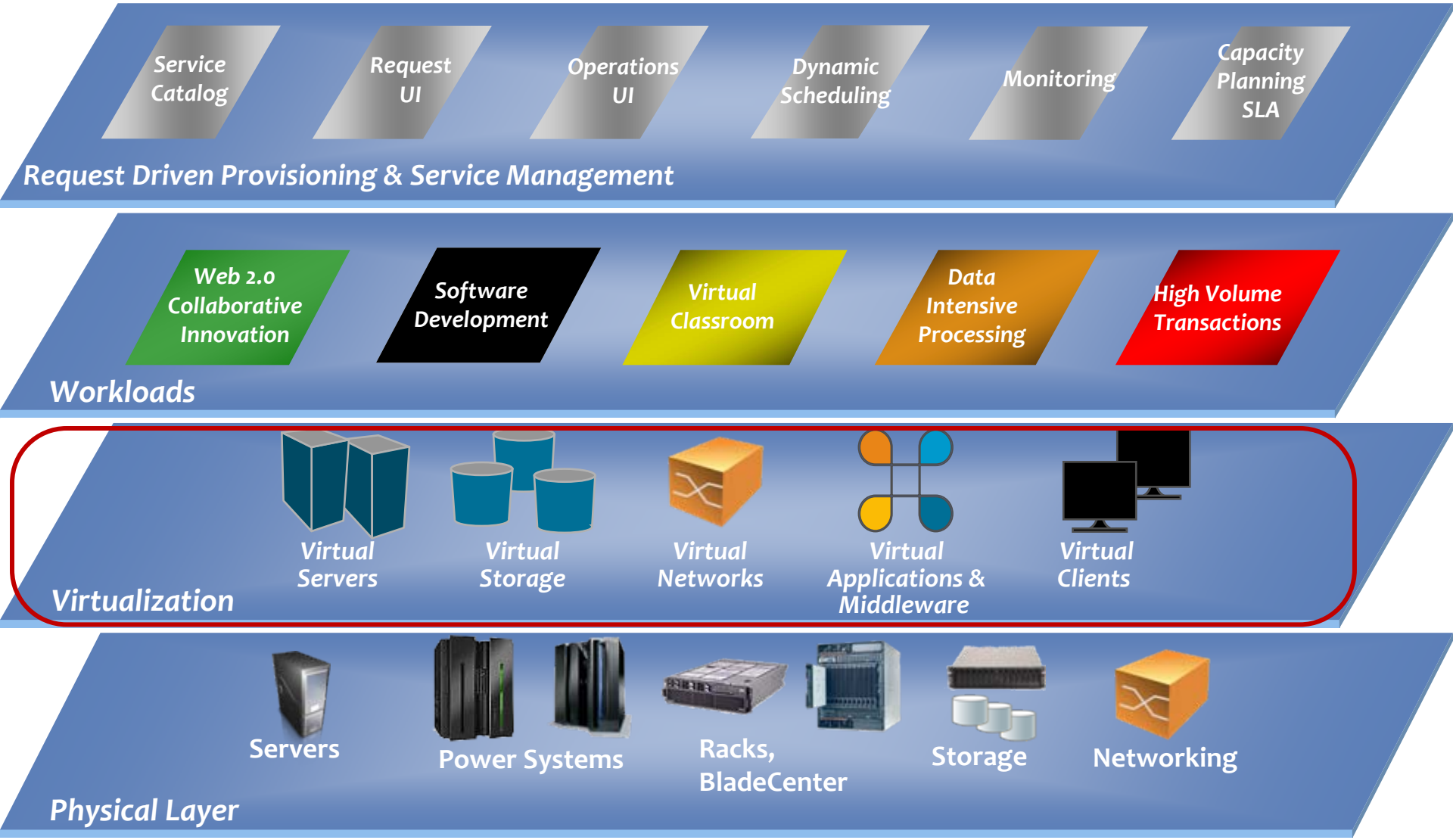
Quad Core:







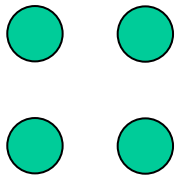
IaaS (5/5)


Service Catalog
Request UI
Operations UI
Dynamic Scheduling
Monitoring
Capacity Planning SLA
Request Driven Provisioning & Service Management
Web 2.0 Collaborative Innovation
Software Development
Virtual Classroom
Data Intensive Processing
High Volume Transactions
Workloads
Virtual Servers
Virtual Storage
Virtual Networks
Virtual Applications & Middleware
Virtual Clients
Virtualization
Servers
Power Systems
Racks, BladeCenter
Storage
Networking
Physical Layer

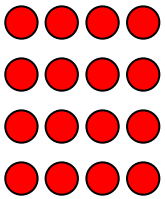
Resource sharing and consolidation

- **Offering computing resources as a service or utility through:**
  - Virtualization
  - Dynamic provisioning

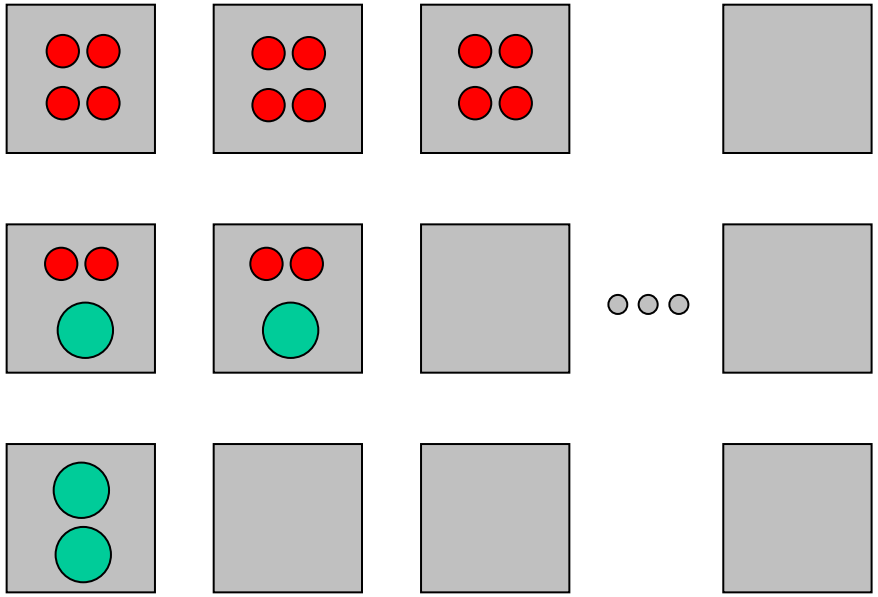
User 1:



User 2:

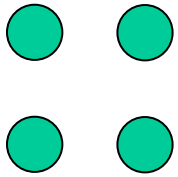


Customizable Shared Resource:

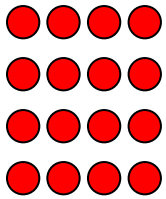


Heterogeneous Physical Resources

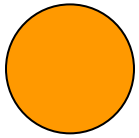
User 1:



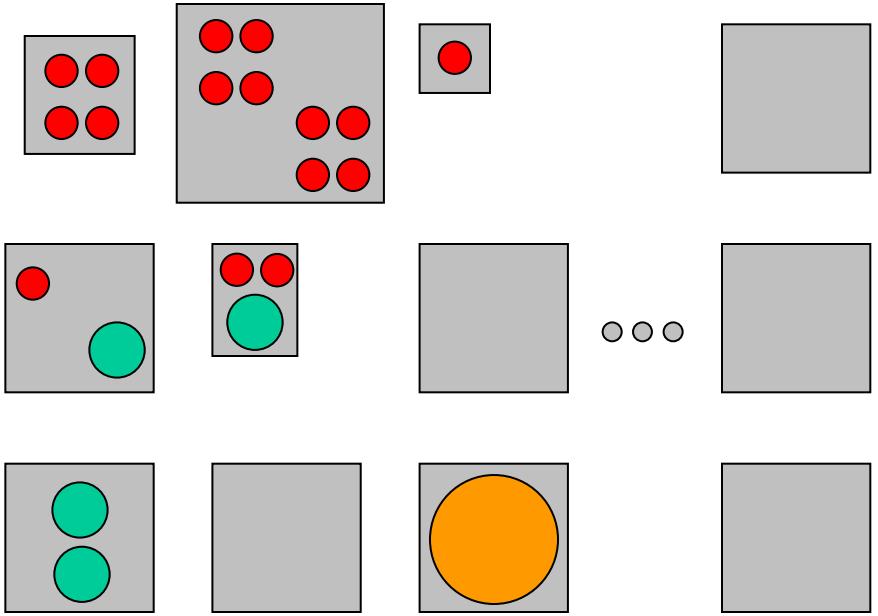
User 2:



User 3:



Customizable Shared
Heterogeneous Resource:

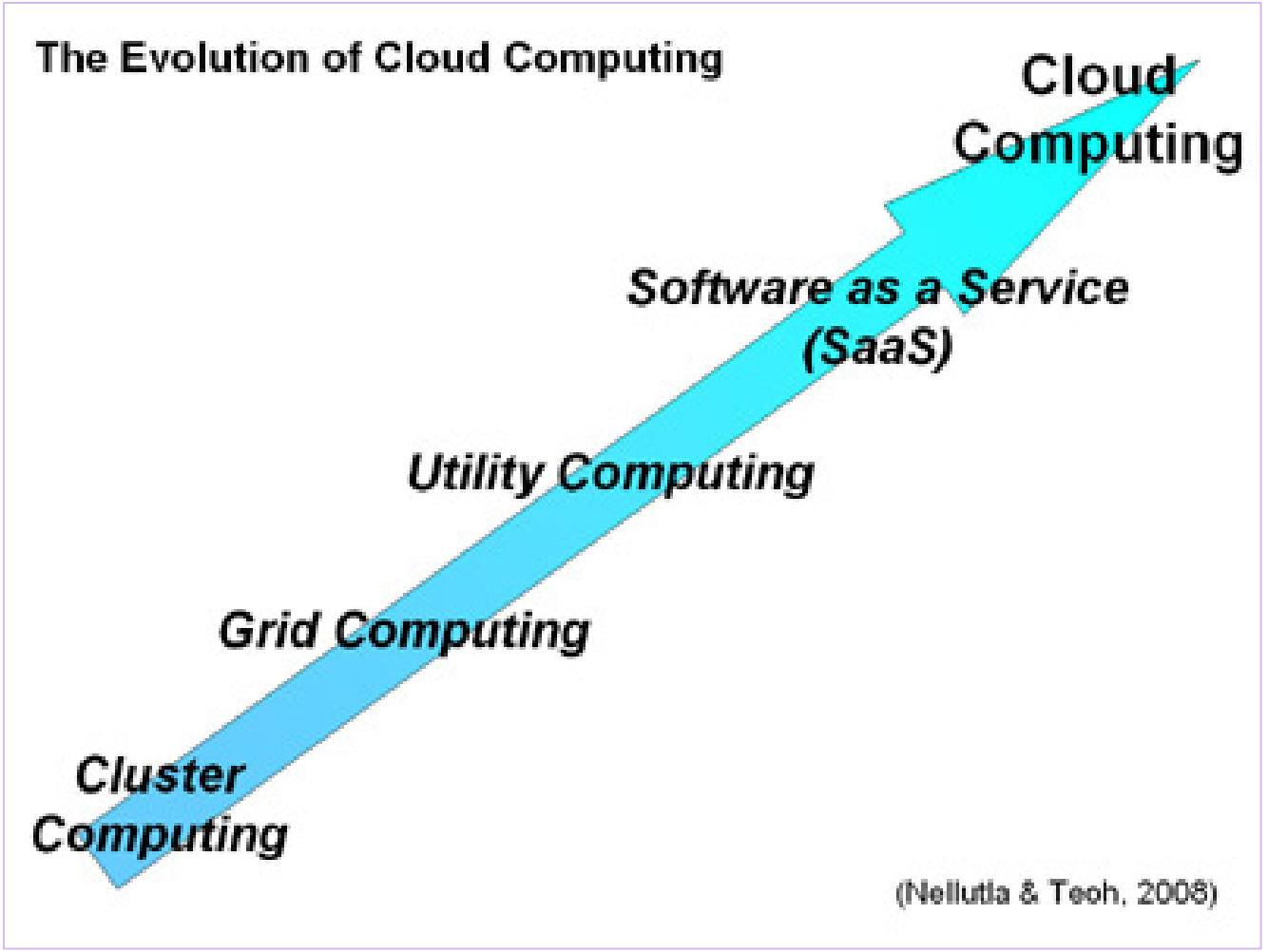


More (XaaS): Everything as a Service EaaS

- **Desktop: DaaS**
  - Use your desktop virtually from anywhere
- **Communication: CaaS**
- **Virtualization: VaaS**
- **Hardware: HaaS**
- **…etc**

Evolution

- Discussed in lecture1



Enabling Technologies

- Virtualization
- Web 2.0
- Distributed Storage
- Distributed Computing
- Utility Computing
- Network Bandwidth & Latency
- Fault-Tolerant Systems

Why Cloud Computing?

- **Large-Scale Data-Intensive Applications**
- **Flexibility**
- **Scalability**
- **Customized to your current needs:**
  - Hardware
  - Software
- **Effect:**
  - Reduce Cost
  - Reduce Maintenance
  - High Utilization
  - High Availability
  - Reduced Carbon Footprint

Why Cloud Computing?

- **Flexibility**
  - Software: Any software platform
  - Access: access resources from any machine connected to the Internet
  - Deploy infrastructure from anywhere at anytime
    - Software controls infrastructure

Why Cloud Computing?

- **Scalability**
  - Instant
  - Control via software
    - Add/cancel/rebuild resources instantly
  - Start small, then scale your resources up/down as you need
  - illusion of infinite resources available on demand

Why Cloud Computing?

- **Customization**
  - Everything in your wish list
    - Software platforms
    - Storage
    - Network bandwidth
    - Speed

Why Cloud Computing?

- Cost
  - Pay-as-you-go model
  - Small/medium size companies can tap the infrastructure of corporate giants.
    - Time to service/market
    - No upfront cost

Why Cloud Computing?

- **Maintenance**
  - Reduce the size of a client's IT department
  - Is the responsibility of the cloud vendor
  - This Includes:
    - Software updates
    - Security patches
    - Monitoring system's health
    - System backup
    - …etc

Why Cloud Computing?

- **Utilization**
  - Consolidation of a large number of resources
    - CPU cycles
    - Storage
    - Network Bandwidth

Why Cloud Computing?

- **Availability**
  - Having access to software, platform, infrastructure from anywhere at any time
  - All you need is a device connected to the internet

- **Reliability**

The system's fault tolerance is managed by the cloud providers and users no longer need to worry about it.

Why Cloud Computing?

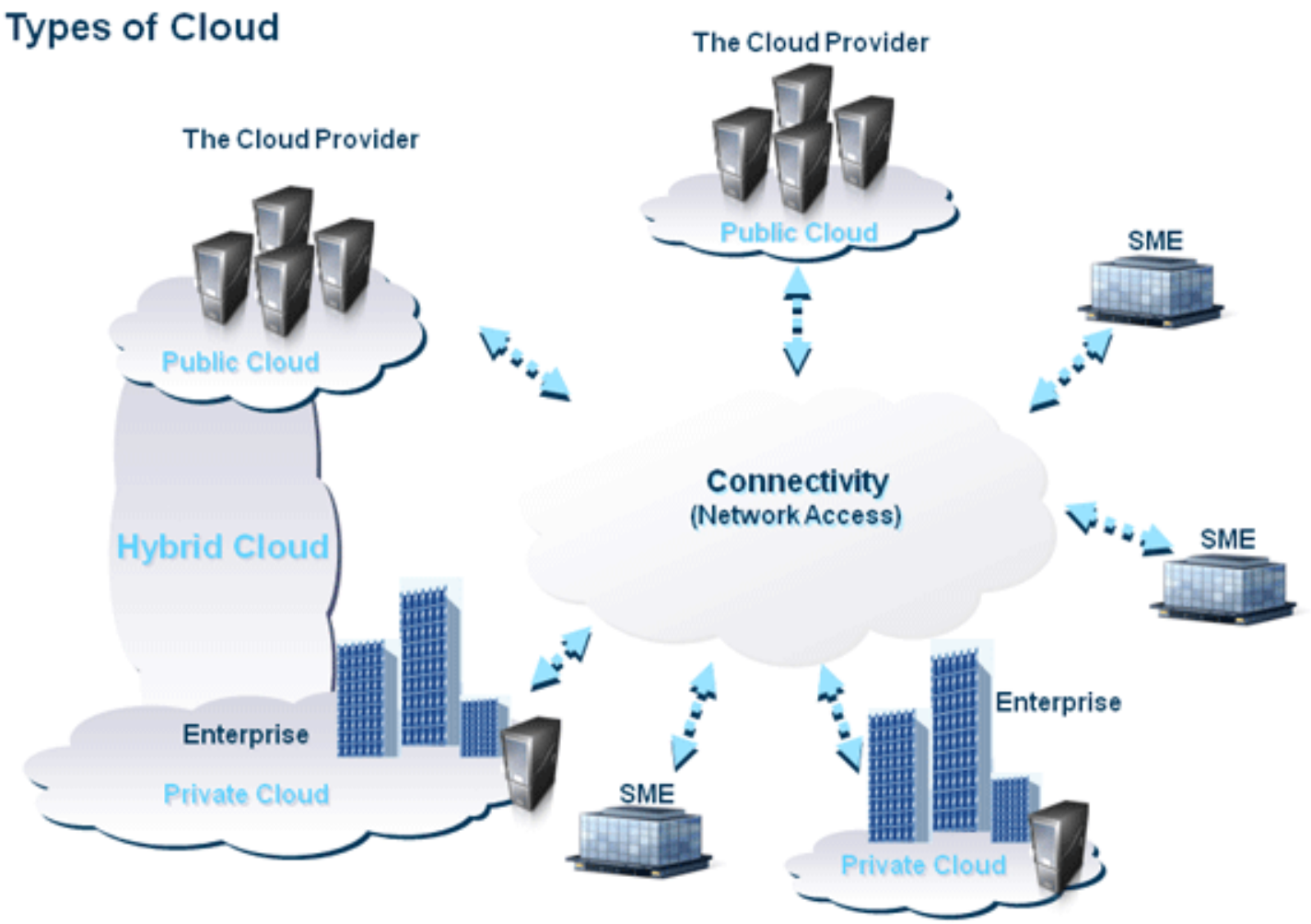
- CO2 Footprint
  - Consolidation of servers
  - Higher utilization
  - Reduced power usage

Drawbacks

- Security
- Privacy
- Vendor lock-in
- Network-dependent
- Migration

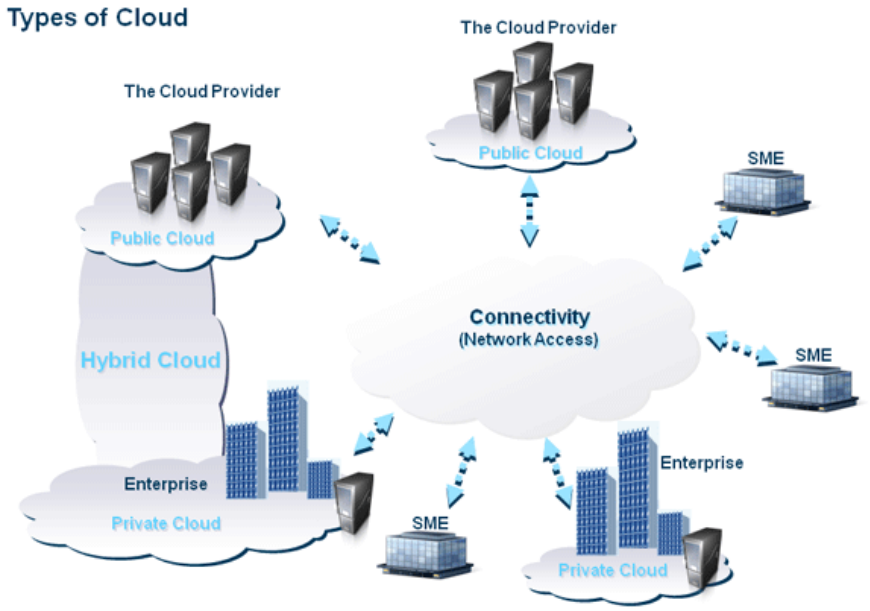
Types of Clouds (1/4)

- Public
- Private
- Hybrid



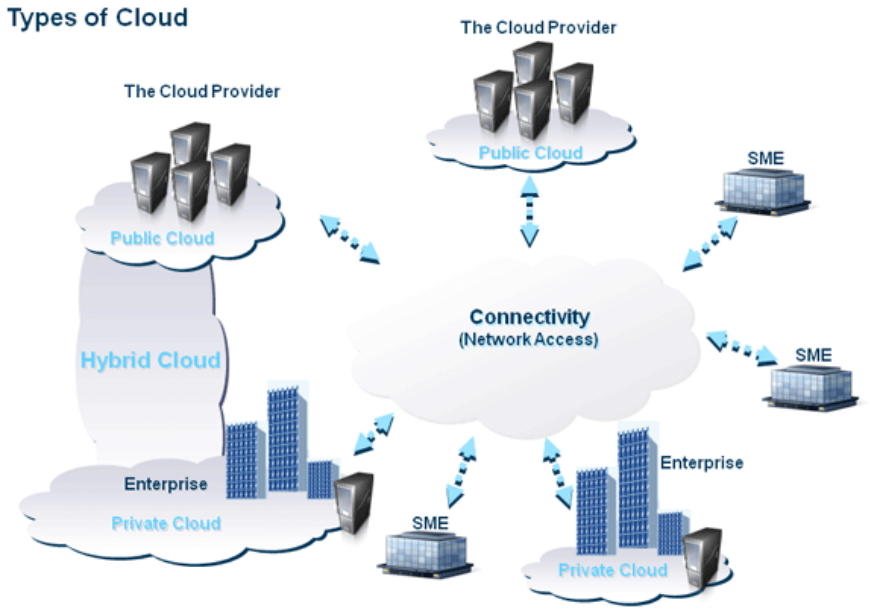
Types of Clouds (2/4)

- **Public (external) cloud**
  - Open Market for on demand computing and IT resources
  - Concerns: Limited SLA, Reliability, Availability, Security, Trust and Confidence
  - Examples: IBM, Google, Amazon, …



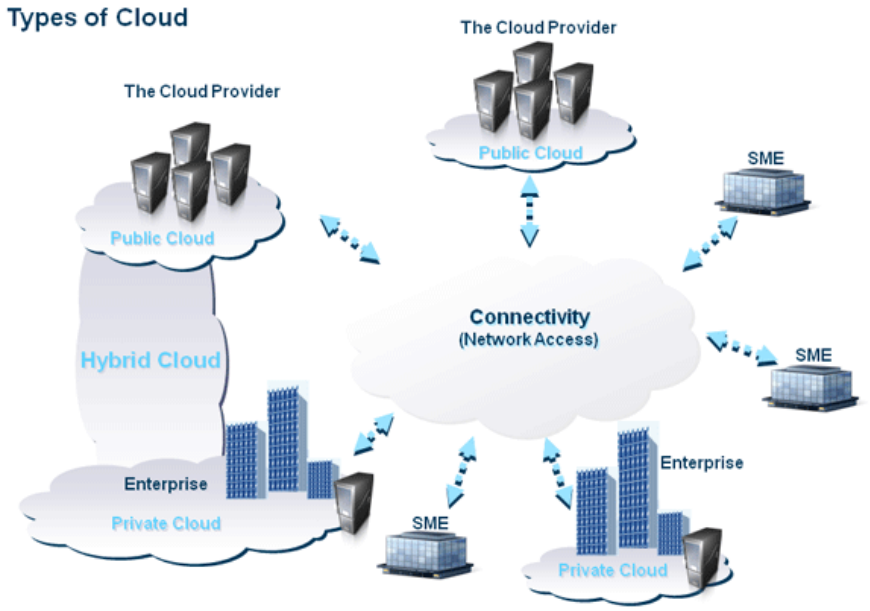
Types of Clouds (3/4)

- **Private (Internal) cloud**
  - For Enterprises/Corporations with large scale IT



Types of Clouds (4/4)

- **Hybrid cloud**
  - Extend the Private Cloud(s) by connecting it to other external cloud vendors to make use of available cloud services from external vendors
- **Cloud Burst**
  - Use the local cloud, when you need more resources, burst into the public cloud



Types of Applications

- Open discussion

System Infrastructure

- Large-scale Data-centric applications
- Exploit parallelism
- Easy to manage
- Elastic (dynamic?)
- Fault-tolerant

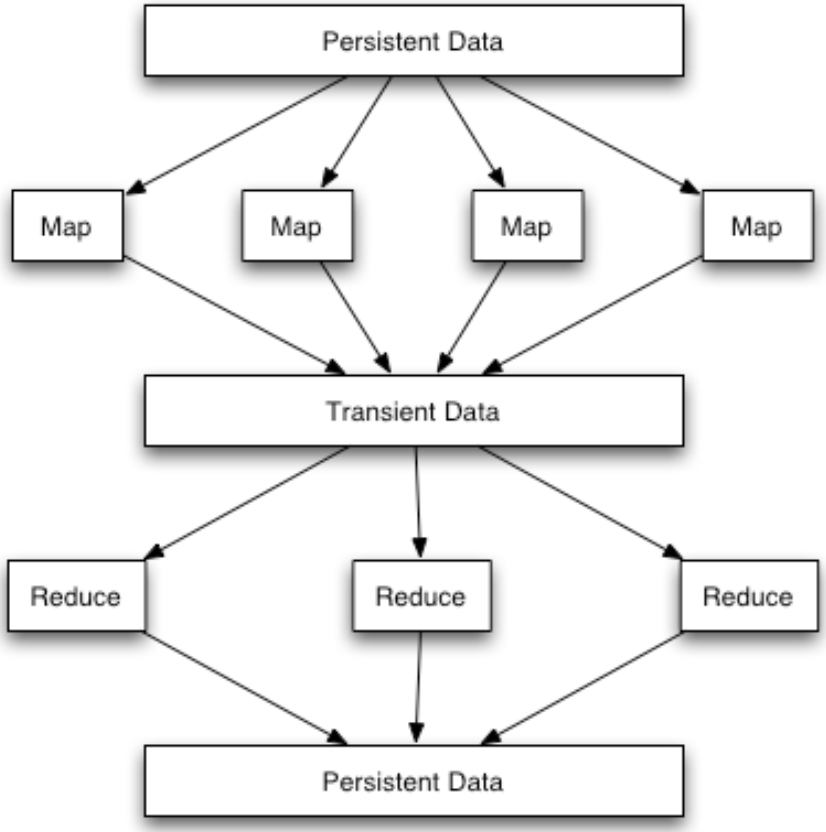
MapReduce and Apache Hadoop

- **MapReduce: Abstraction that simplifies writing applications that access massively distributed data**
- **Hadoop: Open source MapReduce software platform**
- **Distributes data and processing across many nodes**
- **Processes the data locally at each node**
- **Transparent fault tolerance through**
  - Automatic data duplication
  - Automatic detection and restarting of failing nodes

MapReduce Programming Model

- **Functional programming that is easily parallelizable**
- **Split into two phases:**
  - Map – Perform custom function on all items in an array
  - Reduce – Collate map results using custom function
- **Scales well – computation separated from processing dataflow**
- **Illustrative example:**
  - Map that squares the value of numbers in an array

    {1, 2, 3, 4} -> {1, 4, 9, 16}
  - Reduce that sums the squares : 30



Hadoop Map/Reduce

- **The Map-Reduce programming model**
  - **Framework for distributed processing of large data sets**
  - **Pluggable user code runs in generic framework**
- **Example:**
  - **cat * | grep | sort | unique -c | cat > file**
  - **input |** map **| shuffle |** reduce **| output**
- **Natural for unstructured data:**
  - **Log processing**
  - **Web search indexing**
  - **Ad-hoc queries**

Apache Hadoop

- **Open source MapReduce software platform**
- **Automatically provides framework for developing MapReduce applications**
  - Handles mapping and reducing logistics
  - Programmer just provides custom functionality
- **Currently takes custom functionality in Java and Python**
- **Uses an open source Eclipse plug-in to interface with Hadoop**

HDFS

- Very Large Distributed File System
  - 10K nodes, 100 million files, 10 PB
- Assumes Commodity Hardware
  - Files are replication in order to handle hardware failure
  - System detects failures and recovers from them
- Optimized for Batch Processing
  - Data locations exposed so that computations can move to where data resides
  - Provides very high aggregate bandwidth

Distributed File System

- Single Namespace for entire cluster
- Data Coherency
  - Write-once-read-many access model
  - Client can only append to existing files
- Files are broken up into blocks
  - Typically 128 MB block size
  - Each block replicated on multiple DataNodes
- Intelligent Client
  - Client can find location of blocks
  - Client accesses data directly from DataNode