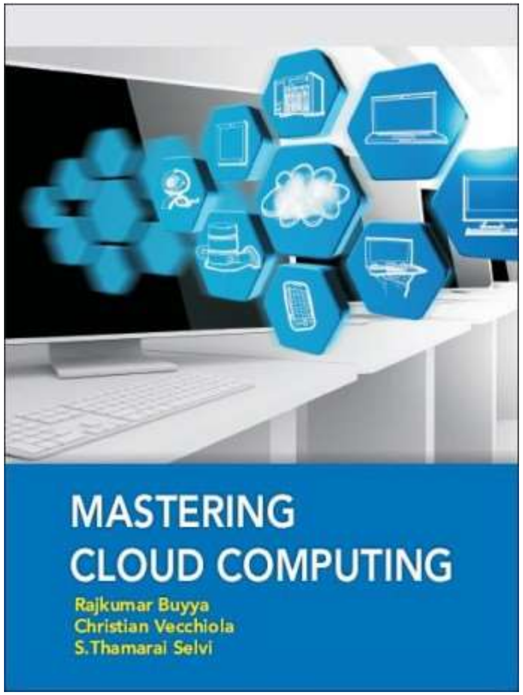
Mastering Cloud Computing

Rajkumar Buyya, Christian Vecchiola, Thamarai Selvi



Chapters

Module I:
Chapter 1 — Introduction
Chapter 3 — Virtualization

Module II
Chapter 4 — Cloud Computing Architecture
Chapter 5 — Aneka: Cloud Application Platform

Module III
Chapter 6 — Concurrent Computing: Thread Programming
Chapter 7 — High-Throughput Computing: Task Programming

Module IV
Chapter 8 — Data Intensive Computing: Map-Reduce Programming

Module V
Chapter 9 — Cloud Platforms in Industry
Chapter 10 — Cloud Applications

Chapter 1 - Introduction

What is Cloud Computing?

LOCATIONS

Public Cloud

Shared resources among a community of users

Private Cloud

Service that is controlled and exclusive to the user

Hybrid Cloud

Ability to move workloads between private and public platforms.

SERVICES

IaaS

Infrastructure as a Service

Outsource the elements of infrastructure like Virtualization, Storage, Networking, Load Balancers

PaaS

Platform as a Service

Core hosting operating system and optional building block services that allow you to run your own applications

SaaS

Software as a Service

Consumed as a service only for the applications needed

Pros +

- Scale and Cost
- Choice and Agility
- Encapsulated Change Management
- Next Generation Architectures

Cons -

- Lock-in to service
- Security (Hacking)
- Lack of control and ownership
- Reliability

The Next Revolution in IT
The Big Switch in IT

- Classical Computing
  - Buy & Own
    - Hardware, System Software, Applications often to meet peak needs.
  - Install, Configure, Test, Verify
  - Manage
  - ..
  - Finally, use it
  - $$$$....$(High CapEx)

Every 18 months?

- Cloud Computing
  - Subscribe
  - Use

  - based on QoS

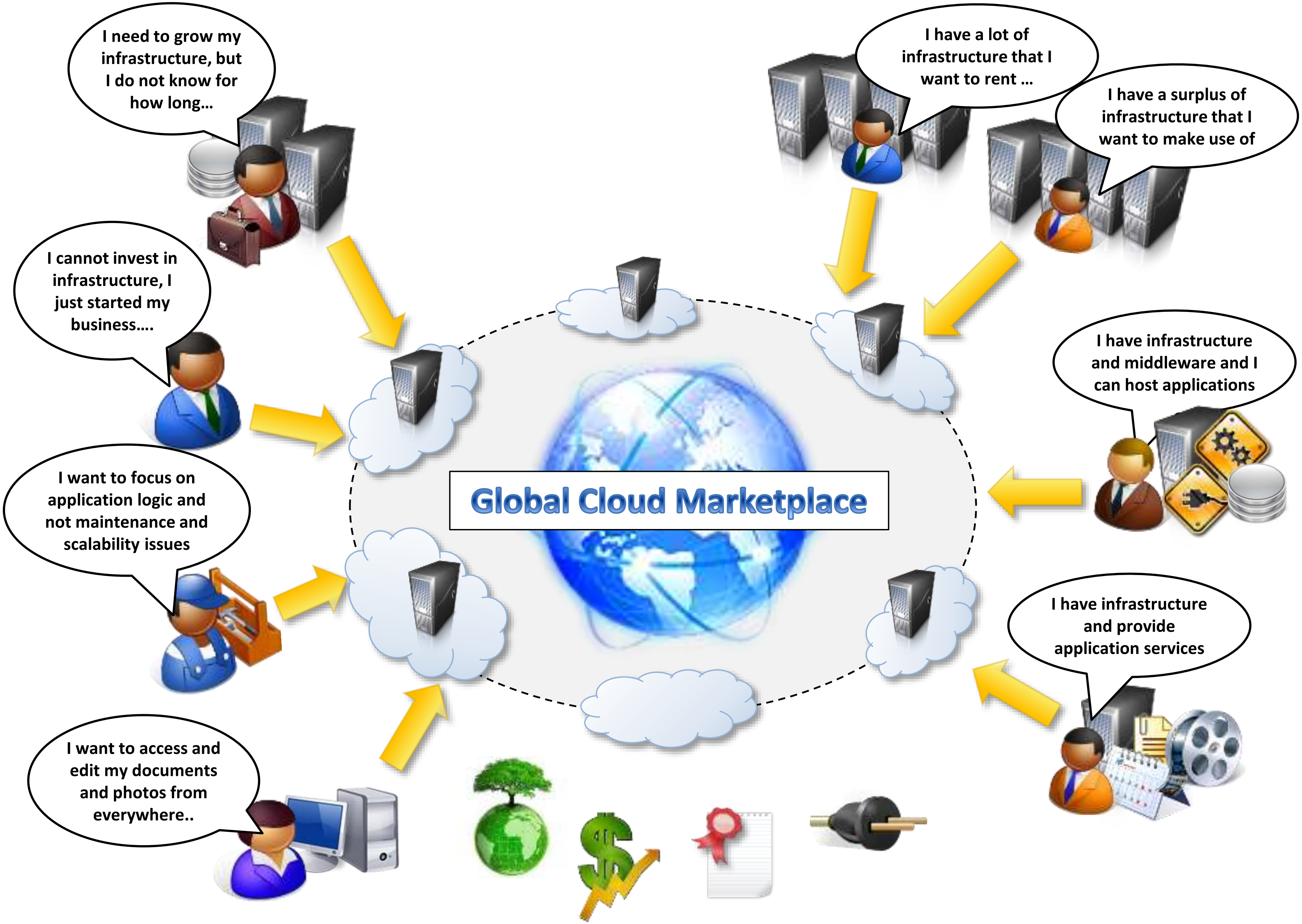
Vision of Cloud Computing

→ Cloud computing provides the facility to provision virtual hardware, runtime environment and services to a person having money.

→ These all things can be used as long as they are needed by the user, there is no requirement for the upfront commitment.

→ The whole collection of computing system is transformed into a collection of utilities, which can be provisioned and composed together to deploy systems in hours rather than days, with no maintenance costs.

→ The long term vision of a cloud computing is that IT services are traded as utilities in an open market without technological and legal barriers.



Global Cloud Marketplace

I need to grow my infrastructure, but I do not know for how long…

I cannot invest in infrastructure, I just started my business….

I want to focus on application logic and not maintenance and scalability issues

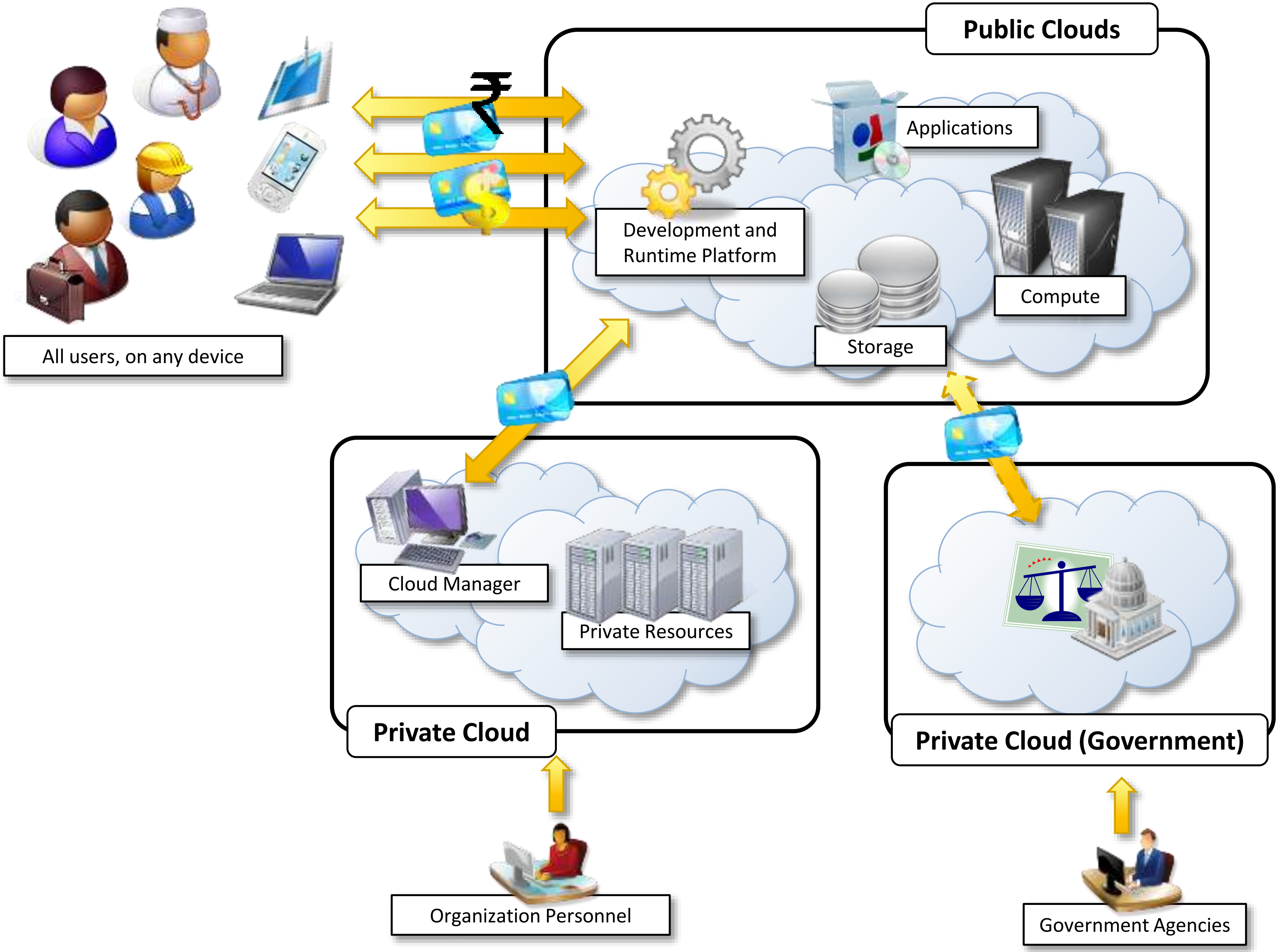
I want to access and edit my documents and photos from everywhere..

I have a lot of infrastructure that I want to rent …

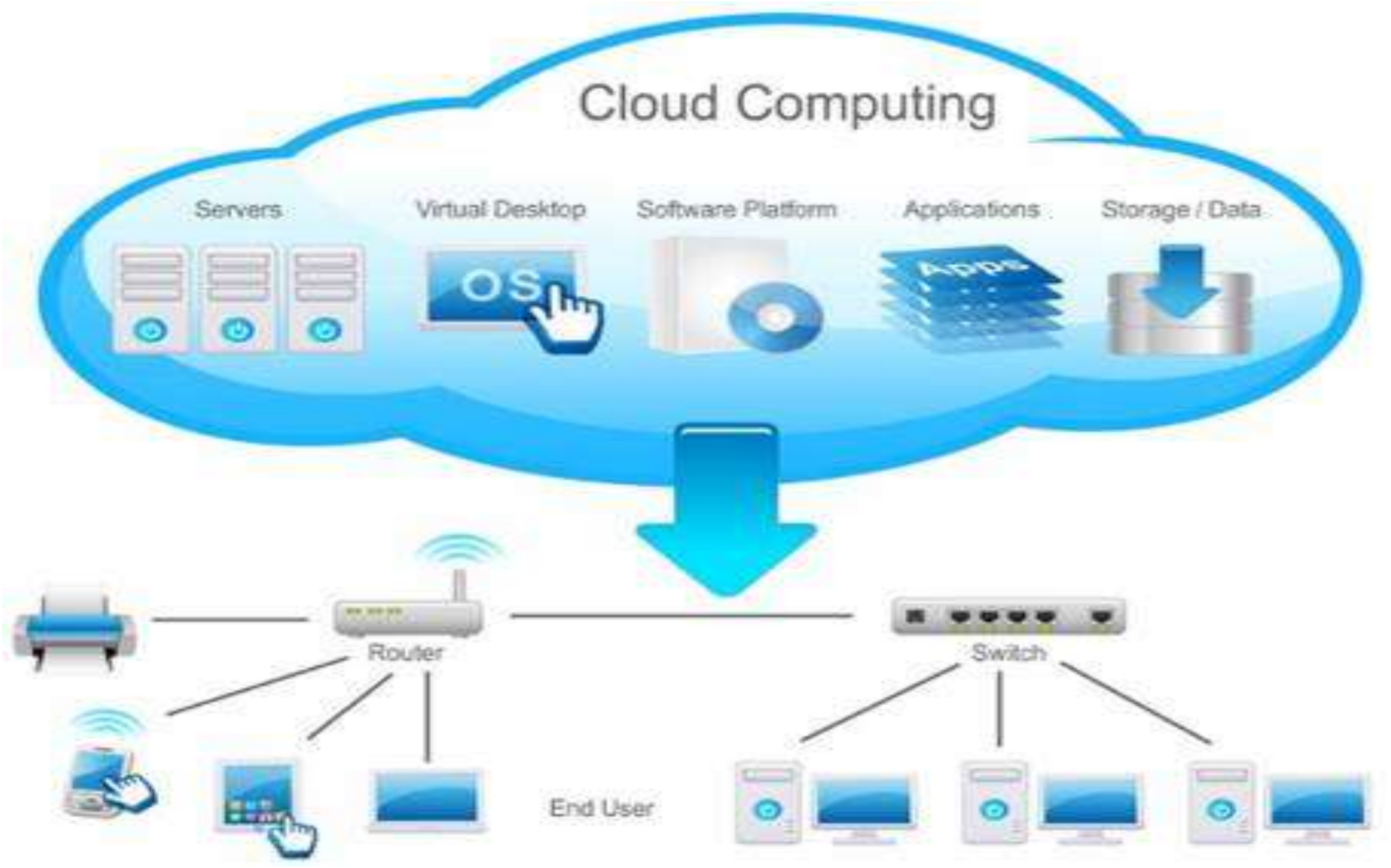
I have a surplus of infrastructure that I want to make use of

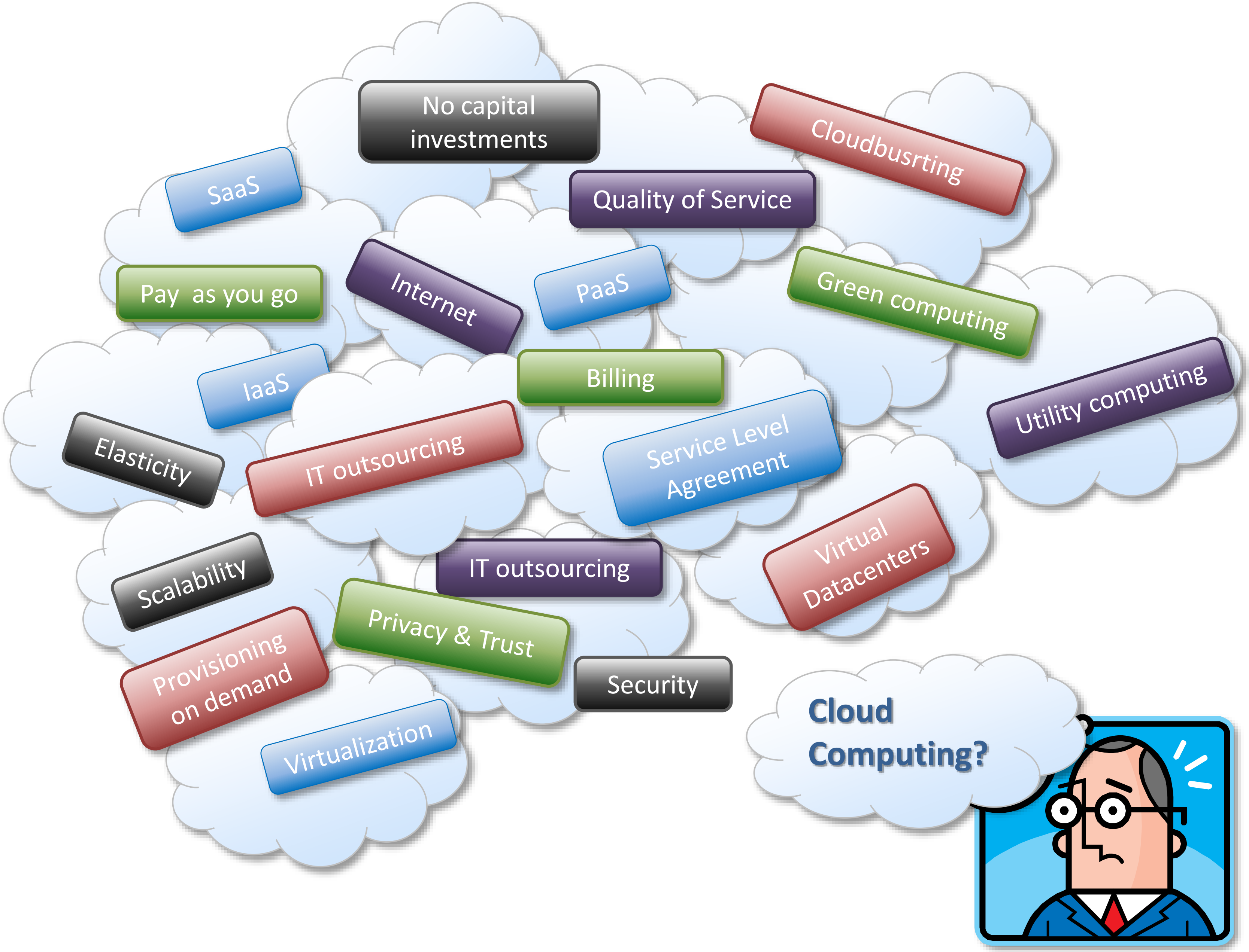
I have infrastructure and middleware and I can host applications

I have infrastructure and provide application services


Public Clouds
Applications
Development and Runtime Platform
Compute
Storage
All users, on any device
₹
$
Cloud Manager
Private Resources
Private Cloud
Private Cloud (Government)
Organization Personnel
Government Agencies

Defining Cloud





No capital investments

Cloudbusrting

SaaS

Quality of Service

Pay as you go

Internet

PaaS

Green computing

IaaS

Billing

Utility computing

Elasticity

IT outsourcing

Service Level Agreement

Scalability

IT outsourcing

Virtual Datacenters

Provisioning on demand

Privacy & Trust

Security

Cloud Computing?

Virtualization

Defining Cloud

A Closer look

- Cloud computing Helping
- Enterprises
- Governments
- Public Institutes
- Private Institutes
- Research Organization

Examples

- Large enterprise can offload some of their activities to Cloud based system.

Example

- Small Enterprises and Start-ups can afford to translate into business results their ideas more quickly without excessive upfront cost

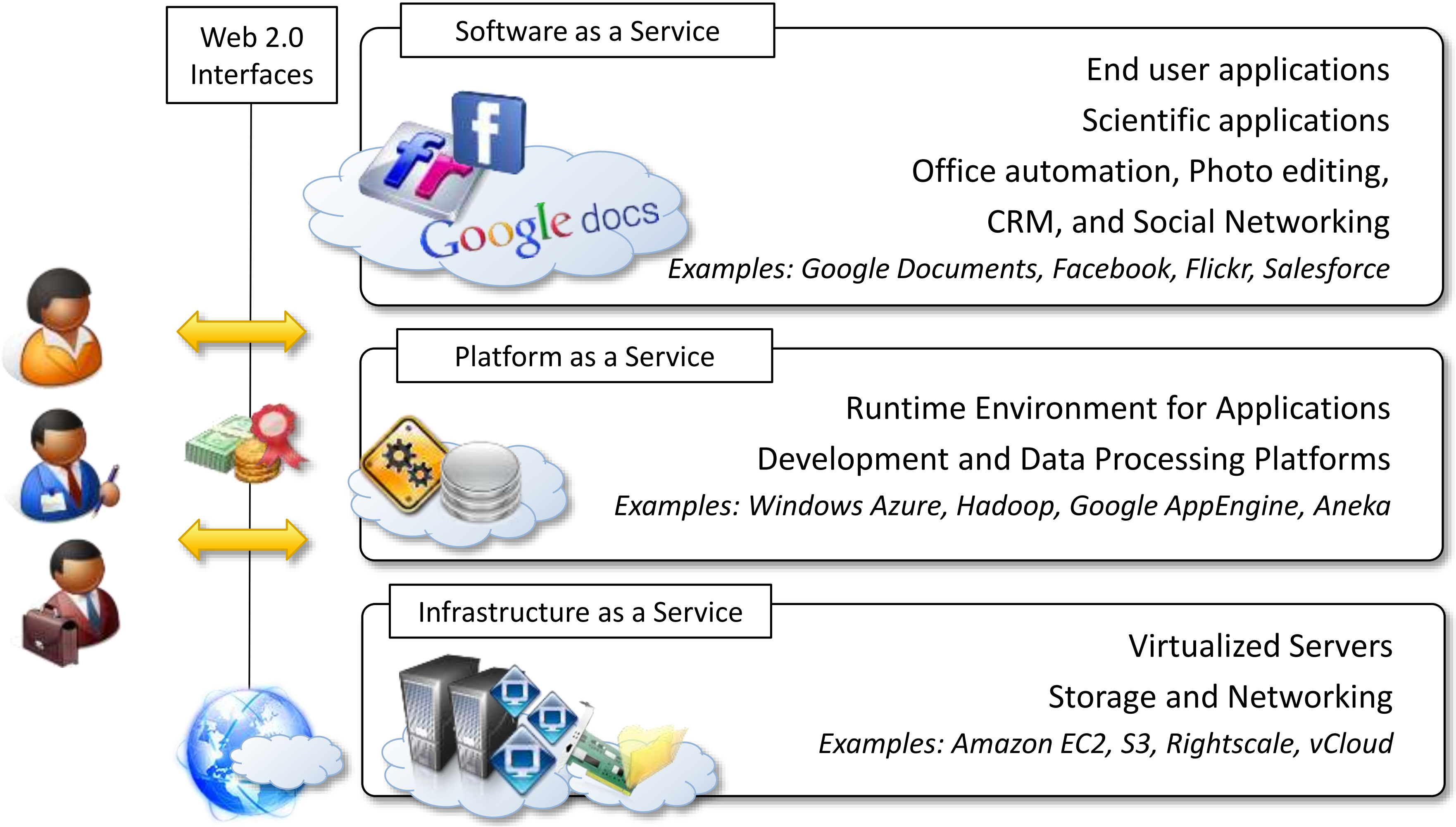
Example

- System Developers can concentrate on business logic rather than dealing with the complexity of infrastructure management and scalability

Example

- End users can have their documents accessible from everywhere and any device

Cloud Computing Reference Model



Web 2.0 Interfaces

Software as a Service

End user applications

Scientific applications

Office automation, Photo editing,

CRM, and Social Networking

Examples: Google Documents, Facebook, Flickr, Salesforce

Platform as a Service

Runtime Environment for Applications

Development and Data Processing Platforms

Examples: Windows Azure, Hadoop, Google AppEngine, Aneka

Infrastructure as a Service

Virtualized Servers

Storage and Networking

Examples: Amazon EC2, S3, Rightscale, vCloud

Characteristics and Benefits

For CSCs and CSPs

- No Upfront Commitments
- On demand access
- Nice pricing
- Simplified application acceleration and scalability
- Efficient resource allocation
- Energy efficiency and seamless creation and use third-party services.

Challenges Ahead

- Dynamic Provisioning of Cloud Computing Services
- Security and Privacy
- Legal issues
- Performance and Bandwidth Cost
- Reliability and Availability

Historical Development of Cloud Computing

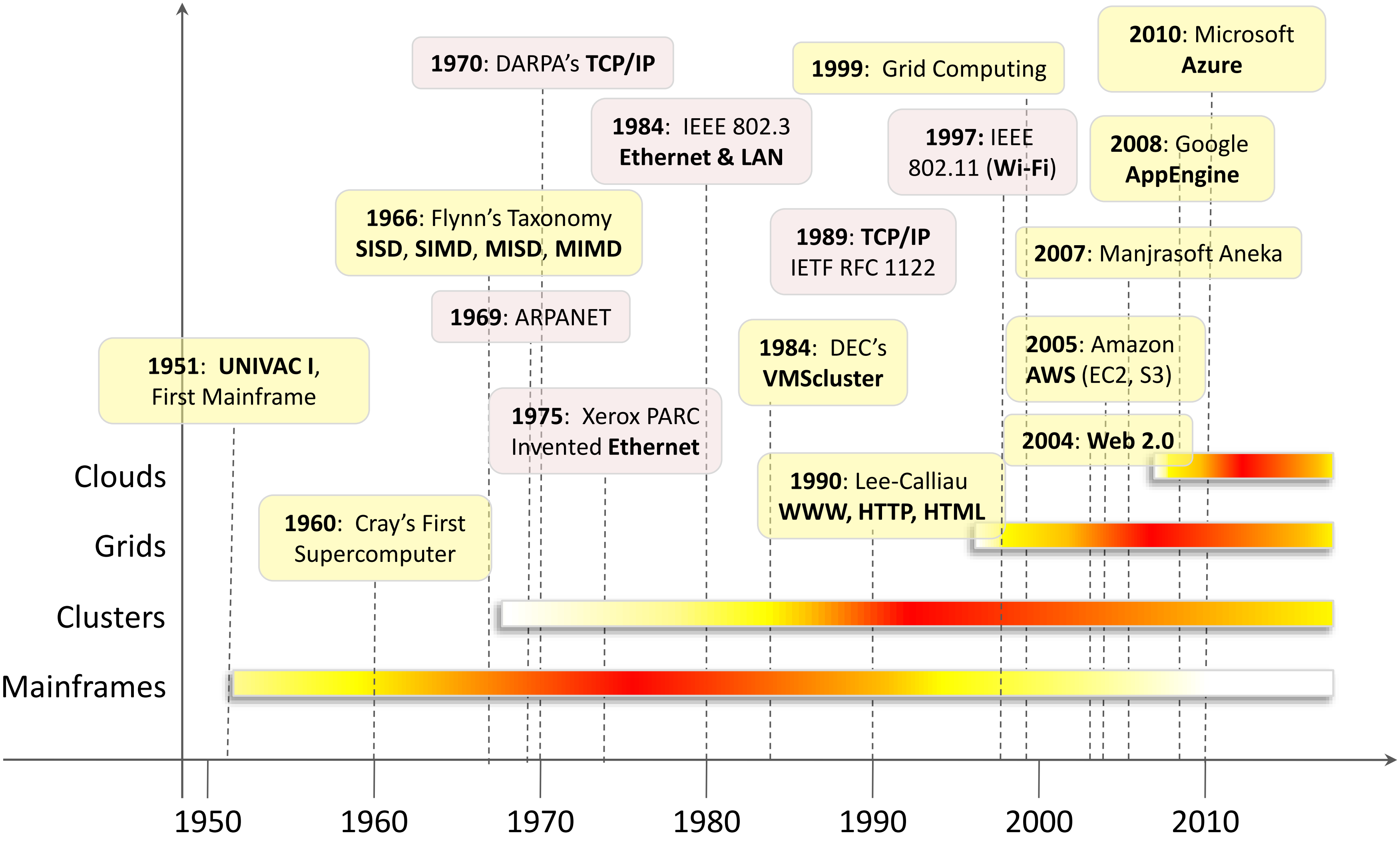
- The history of cloud computing starts from the 1950's and the work done by AT & T in the area of telephone networking

- At that time AT & T had already Begin to develop an architecture and system where data would be located centrally and accessed by business through redesigned telephones and updated telephone network.

- So here we can easily understand that at that time this establishment has been done in the area of telephone networking.

Historical Development of Cloud Computing

- One of the main principles of cloud computing from SAAS (Software as a service) to provide storage on demand, is that the computing capacity varies immediately and transparently with the customer’s need.


1970: DARPA's TCP/IP
2010: Microsoft Azure
1999: Grid Computing
1984: IEEE 802.3 Ethernet & LAN
1997: IEEE 802.11 (Wi-Fi)
2008: Google AppEngine
1966: Flynn's Taxonomy SISD, SIMD, MISD, MIMD
1989: TCP/IP IETF RFC 1122
2007: Manjrasoft Aneka
1969: ARPANET
1951: UNIVAC I, First Mainframe
1984: DEC's VMScluster
2005: Amazon AWS (EC2, S3)
1975: Xerox PARC Invented Ethernet
2004: Web 2.0
1990: Lee-Calliau WWW, HTTP, HTML
1960: Cray's First Supercomputer
Clouds
Grids
Clusters
Mainframes
1950
1960
1970
1980
1990
2000
2010

Evolution of cloud technologies

- *Distributed Systems*

→A distributed system is a collection of independent computers that appears to its users as a single system and also it acts as a single computer.

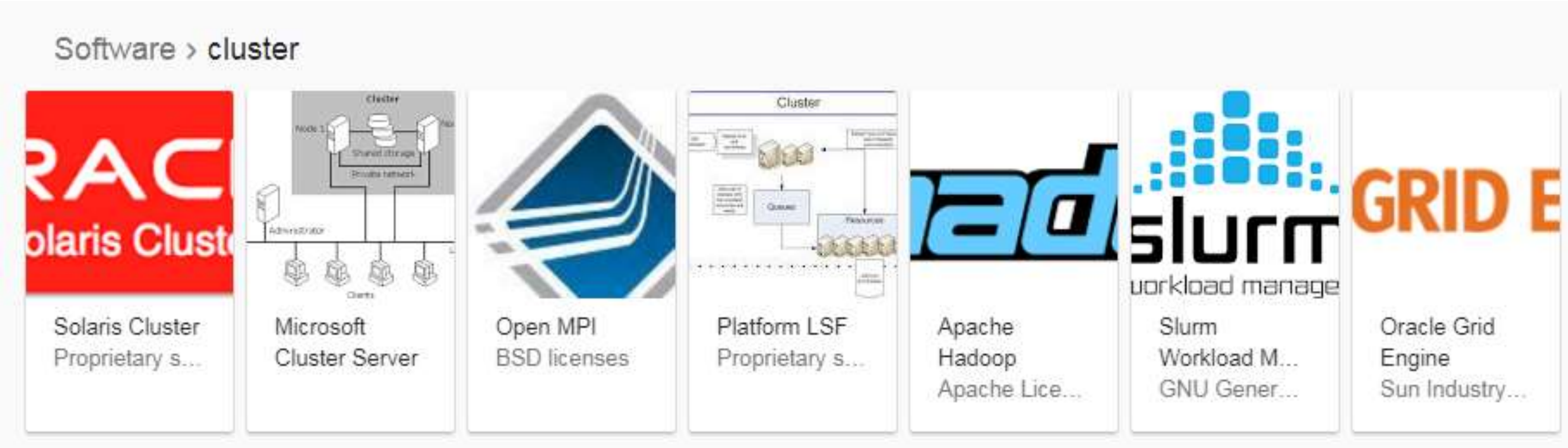
→The main and primary motive of distributed systems is to share resources and to utilize them better.

→This is absolutely true in case of cloud computing because in cloud computing we are sharing the single resource by paying rent.

→The resource is single because the definition of cloud computing clearly states that in cloud computing the single central copy of a particular software is stored in a sever (which is located on a anonymous location) and users are accessing that on PAY PER USE BASIS.

- *Mainframes*
- A large high-speed computer, especially one supporting numerous workstations or peripherals.
- the central processing unit and primary memory of a computer.

- *Clusters*
- A **computer cluster** consists of a set of loosely or tightly connected computers that work together so that, in many respects, they can be viewed as a single system.
- computer clusters have each node set to perform the same task, controlled and scheduled by software.



Software › cluster

Solaris Cluster
Proprietary s...

Microsoft
Cluster Server

Open MPI
BSD licenses

Platform LSF
Proprietary s...

Apache
Hadoop
Apache Lice...

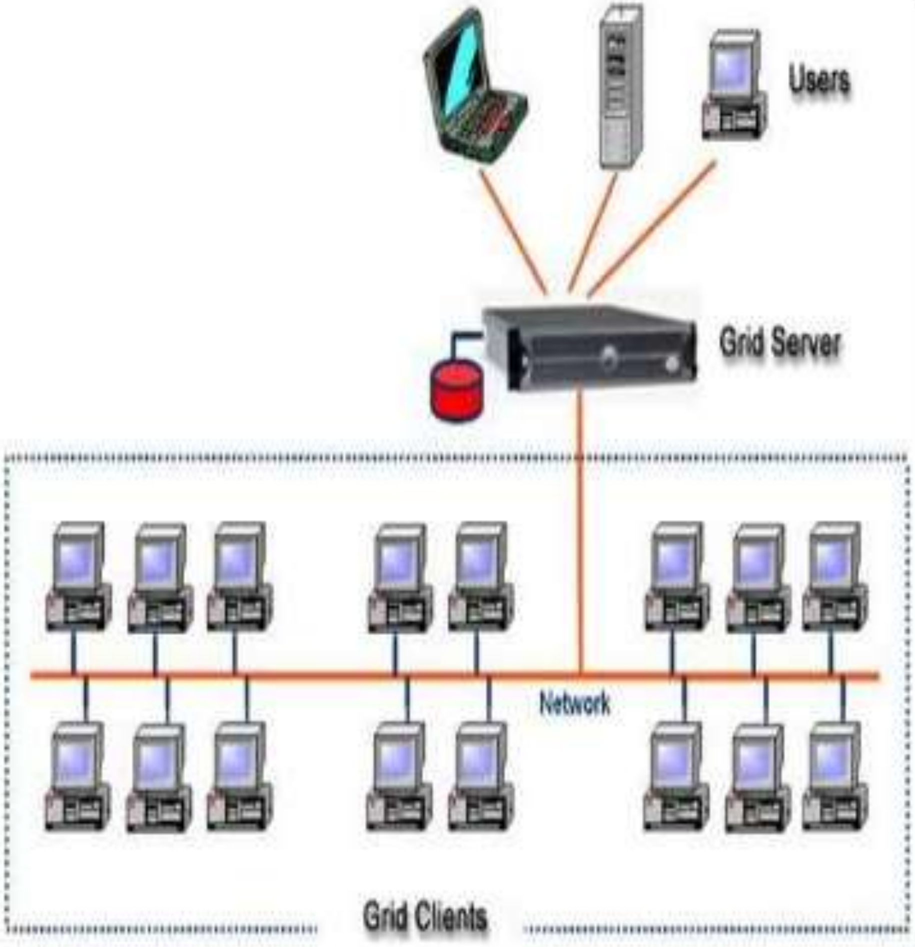
Slurm
Workload M...
GNU Gener...

Oracle Grid
Engine
Sun Industry...

- *Grids*
- **Grid computing** is the collection of **computer** resources from multiple locations to reach a common goal. The **grid** can be thought of as a **distributed** system with non-interactive workloads that involve a large number of files.

- The components of a cluster are usually connected to each other through fast local area networks, with each *node* (computer used as a server) running its own instance of an operating system. In most circumstances,

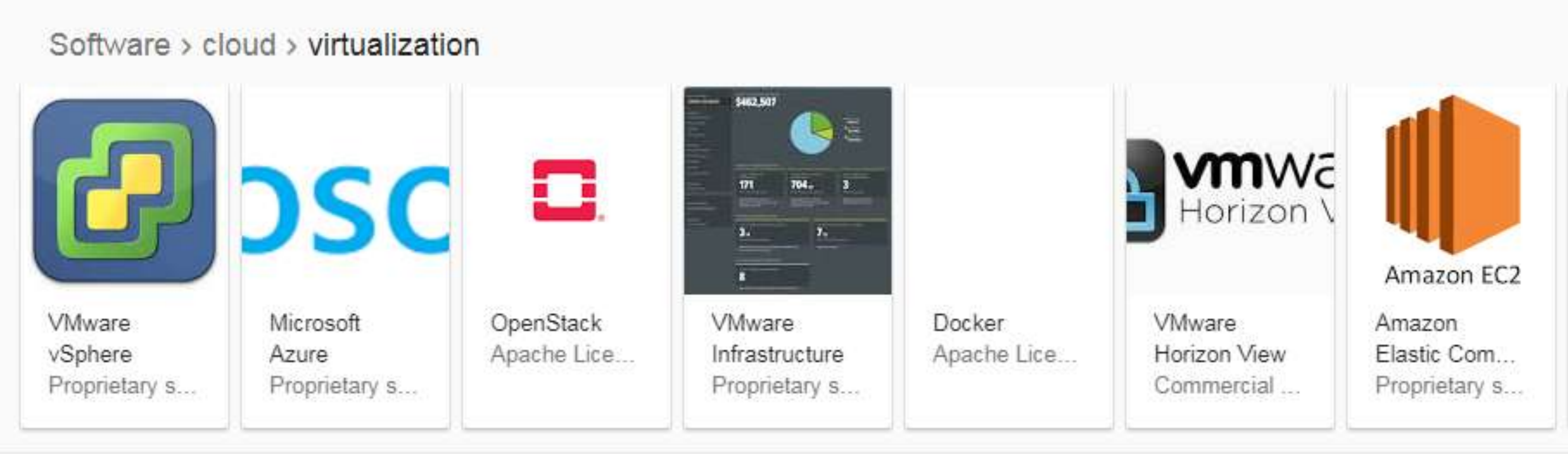
How Grid computing works ?



In general, a grid computing system requires:

- **At least one computer, usually a server, which handles all the administrative duties for the System**
- **A network of computers running special grid computing network software.**
- **A collection of computer software called middleware**

- *Virtualization*
- In computing, **virtualization** refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources.
- Another Cloud Technology of Cloud Computing
- Hardware virtualization
- Storage or Network Virtualization



Software › cloud › virtualization

VMware vSphere
Proprietary s...

Microsoft Azure
Proprietary s...

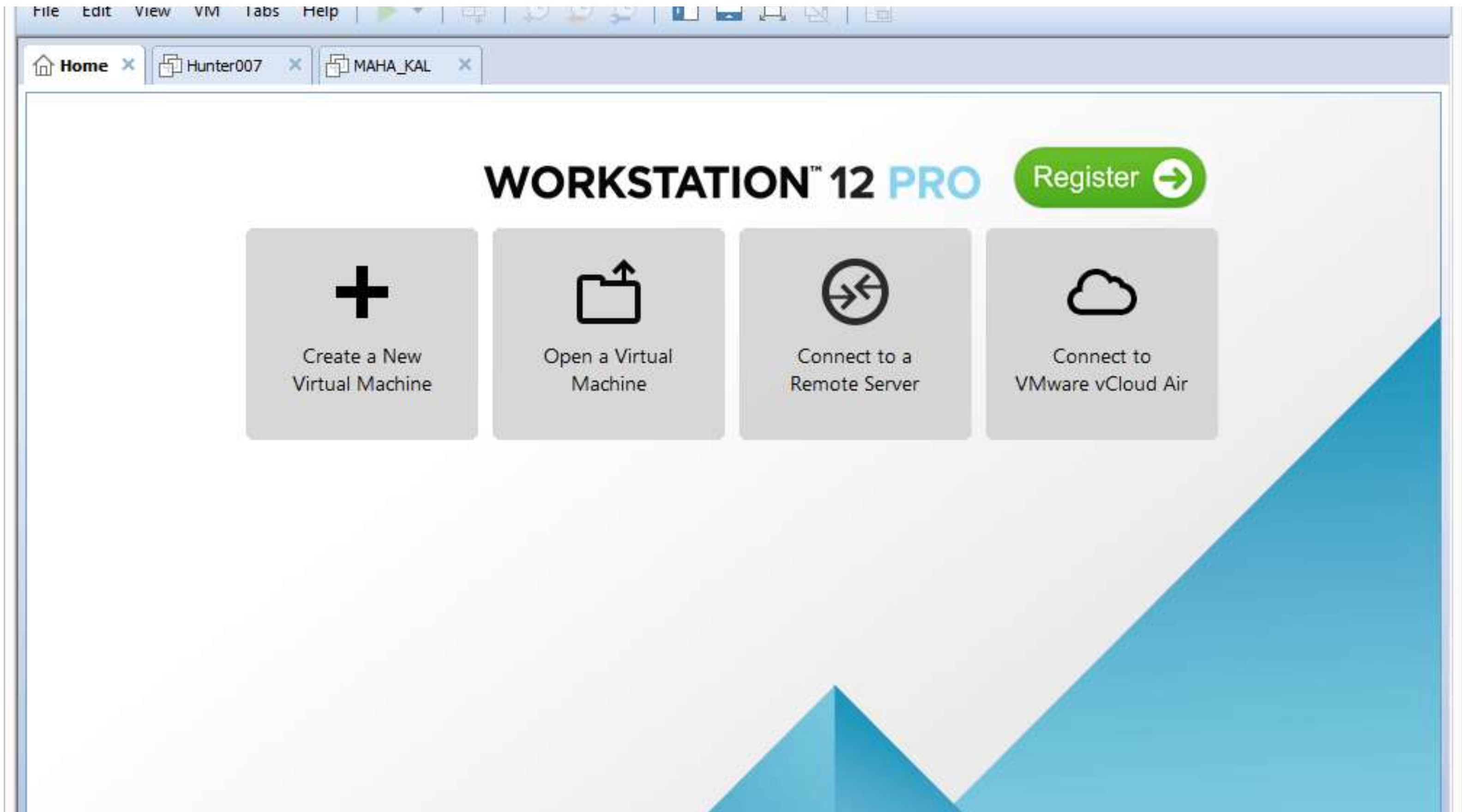
OpenStack
Apache Lice...

VMware Infrastructure
Proprietary s...

Docker
Apache Lice...

VMware Horizon View
Commercial ...

Amazon Elastic Com...
Proprietary s...

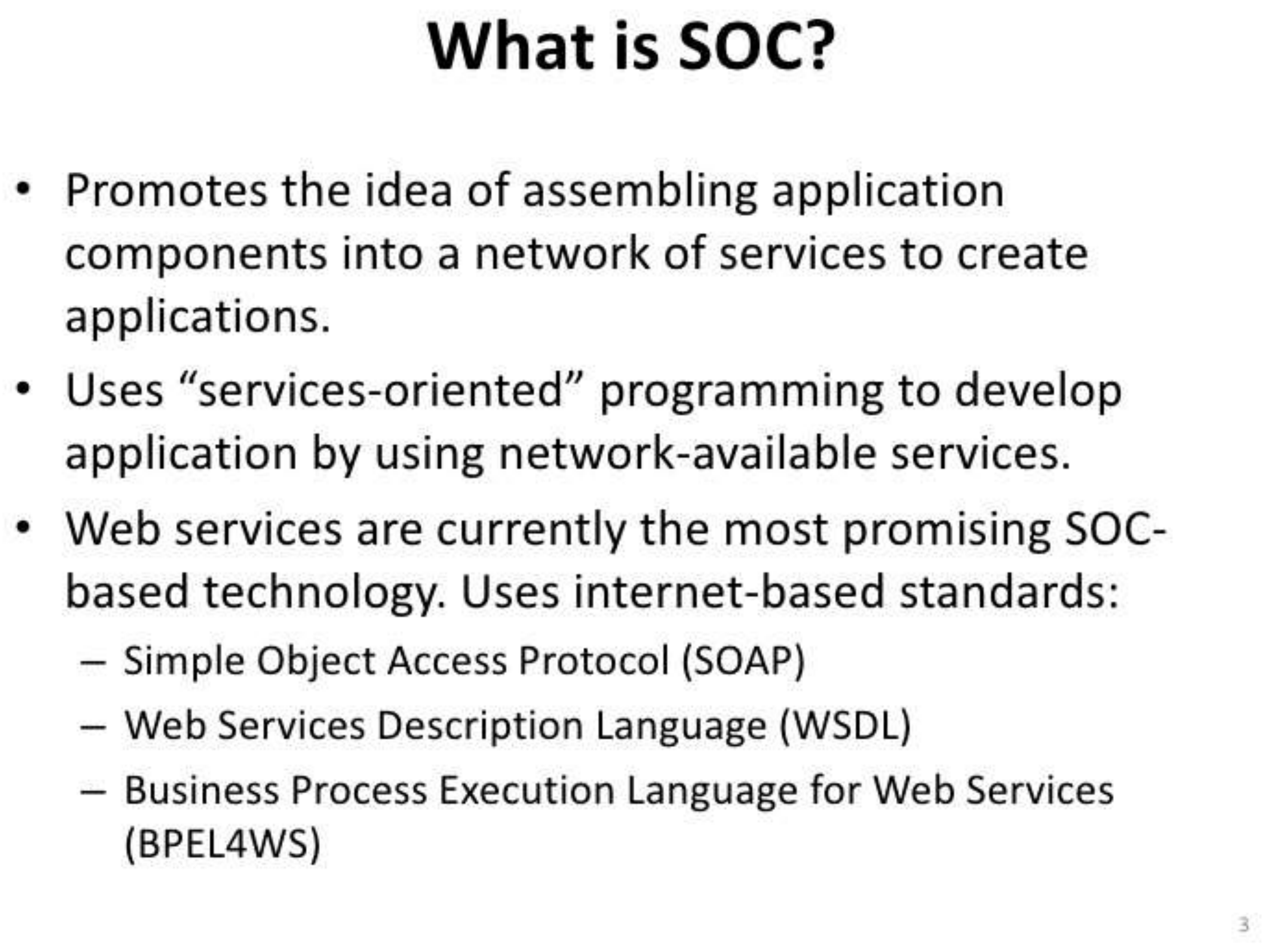


- *Web 2.0*
- the second stage of development of the Internet, characterized especially by the change from static web pages to dynamic or user-generated content and the growth of social media.

JS
HTML
CSS
AJAX
Asynchronous Javascript And XML
JSON
XML
php

TypePad
twitter
Linked in
facebook
zynga
Wikipedia
The Free Encyclopedia
hulu
tumblr.
You Tube
ZOHO
Work. Online
flickr
Google maps
NETFLIX
PANDORA
yelp
Google IO

- *Service-Oriented Computing (SOC)*

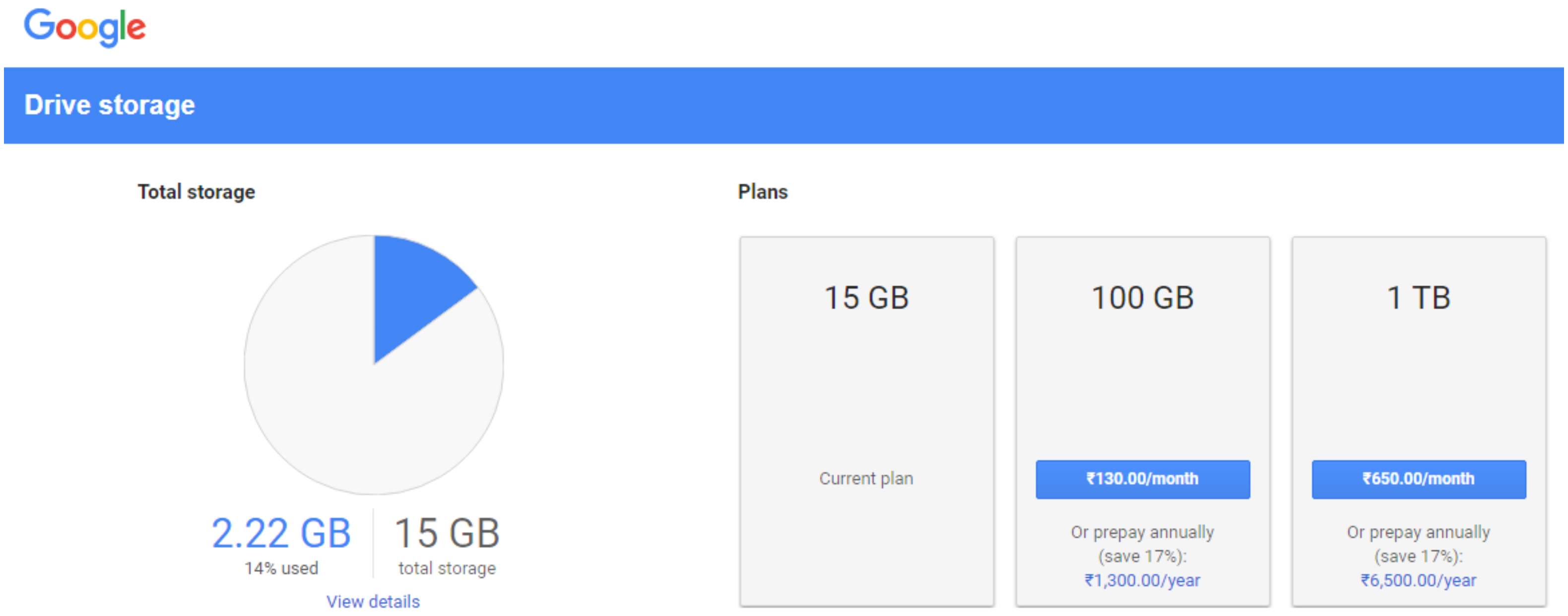


What is SOC?

- Promotes the idea of assembling application components into a network of services to create applications.
- Uses “services-oriented” programming to develop application by using network-available services.
- Web services are currently the most promising SOC-based technology. Uses internet-based standards:
  - Simple Object Access Protocol (SOAP)
  - Web Services Description Language (WSDL)
  - Business Process Execution Language for Web Services (BPEL4WS)

- Core Reference model for Cloud Computing System
- SOC Introduce Two main Concepts
- Quality of Service (QOS)
- Software as Service (SaaS)

- *Utility Oriented Computing*
- **The Computer Utility**, is a service provisioning model in which a service provider makes computing resources and infrastructure management available to the customer as needed, and charges them for specific usage rather than a flat rate.



Google

Drive storage

Total storage

2.22 GB
14% used

15 GB
total storage

View details

Plans

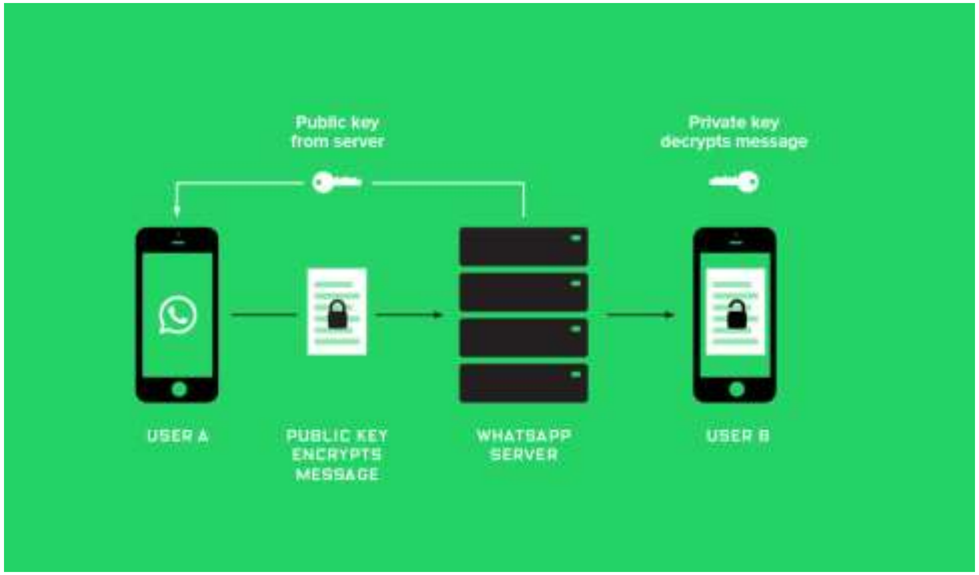
15 GB
Current plan

100 GB
₹130.00/month
Or prepay annually (save 17%):
₹1,300.00/year

1 TB
₹650.00/month
Or prepay annually (save 17%):
₹6,500.00/year

Building Cloud Computing Environment

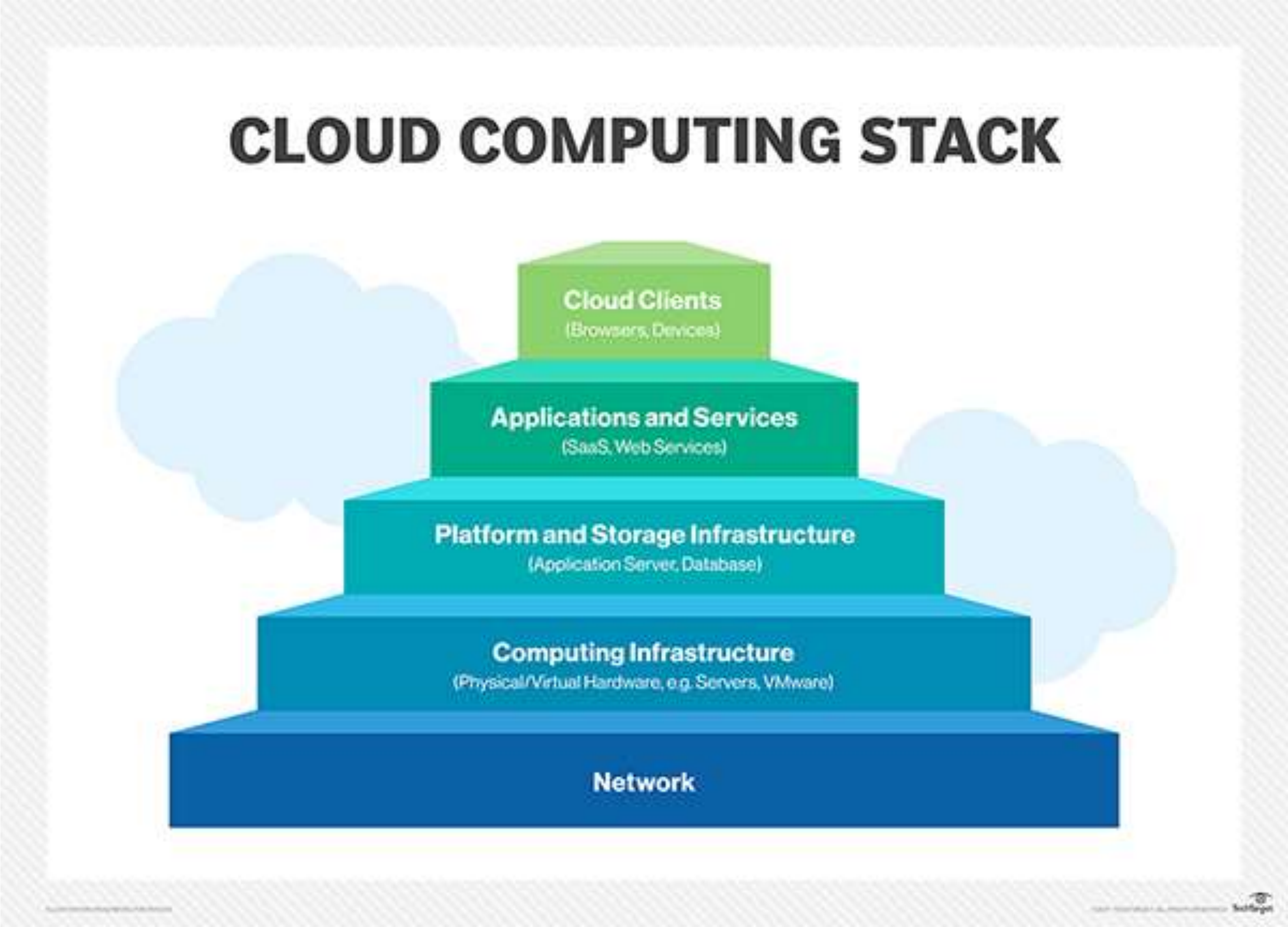
- Application Development



- Enterprise Application

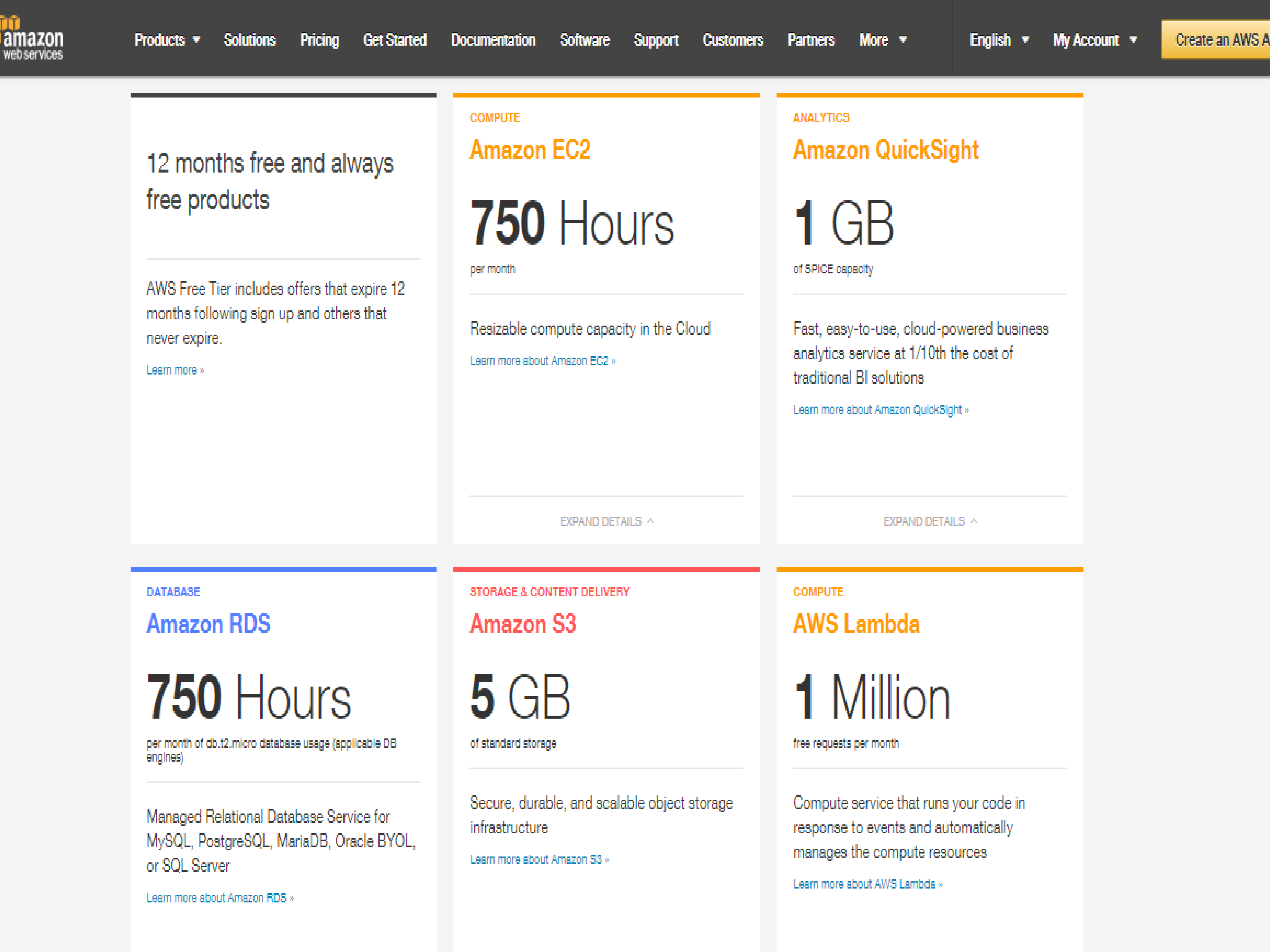
Myntra
skype™
SCALR CLOUD MANAGEMENT
bmcsoftware
Gravitant
abiquo® ENTERPRISE CLOUD
RiGHTSCaLe®
enSTRATUS
ca technologies
servicemesh
kaavo
VMTurbo

Infrastructure and System Development


CLOUD COMPUTING STACK
Cloud Clients
(Browsers, Devices)
Applications and Services
(SaaS, Web Services)
Platform and Storage Infrastructure
(Application Server, Database)
Computing Infrastructure
(Physical/Virtual Hardware, e.g. Servers, VMware)
Network

Computing Platform and Technologies

- AWS – Amazon Web Service



12 months free and always free products

AWS Free Tier includes offers that expire 12 months following sign up and others that never expire.

Learn more »

COMPUTE

Amazon EC2

750 Hours

per month

Resizable compute capacity in the Cloud

Learn more about Amazon EC2 »

EXPAND DETAILS ^

ANALYTICS

Amazon QuickSight

1 GB

of SPICE capacity

Fast, easy-to-use, cloud-powered business analytics service at 1/10th the cost of traditional BI solutions

Learn more about Amazon QuickSight »

EXPAND DETAILS ^

DATABASE

Amazon RDS

750 Hours

per month of db.t2.micro database usage (applicable DB engines)

Managed Relational Database Service for MySQL, PostgreSQL, MariaDB, Oracle BYOL, or SQL Server

Learn more about Amazon RDS »

STORAGE & CONTENT DELIVERY

Amazon S3

5 GB

of standard storage

Secure, durable, and scalable object storage infrastructure

Learn more about Amazon S3 »

COMPUTE

AWS Lambda

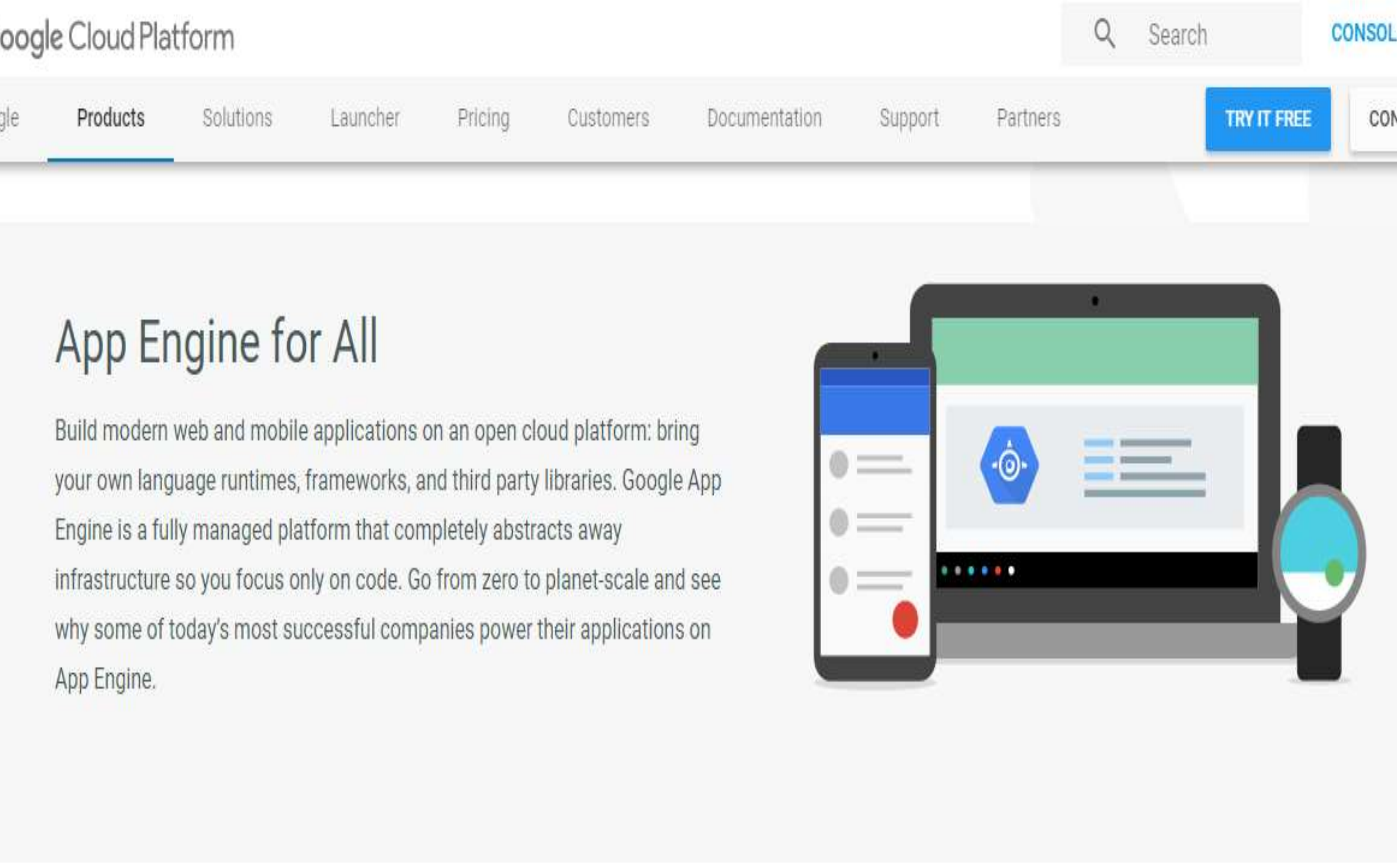
1 Million

free requests per month

Compute service that runs your code in response to events and automatically manages the compute resources

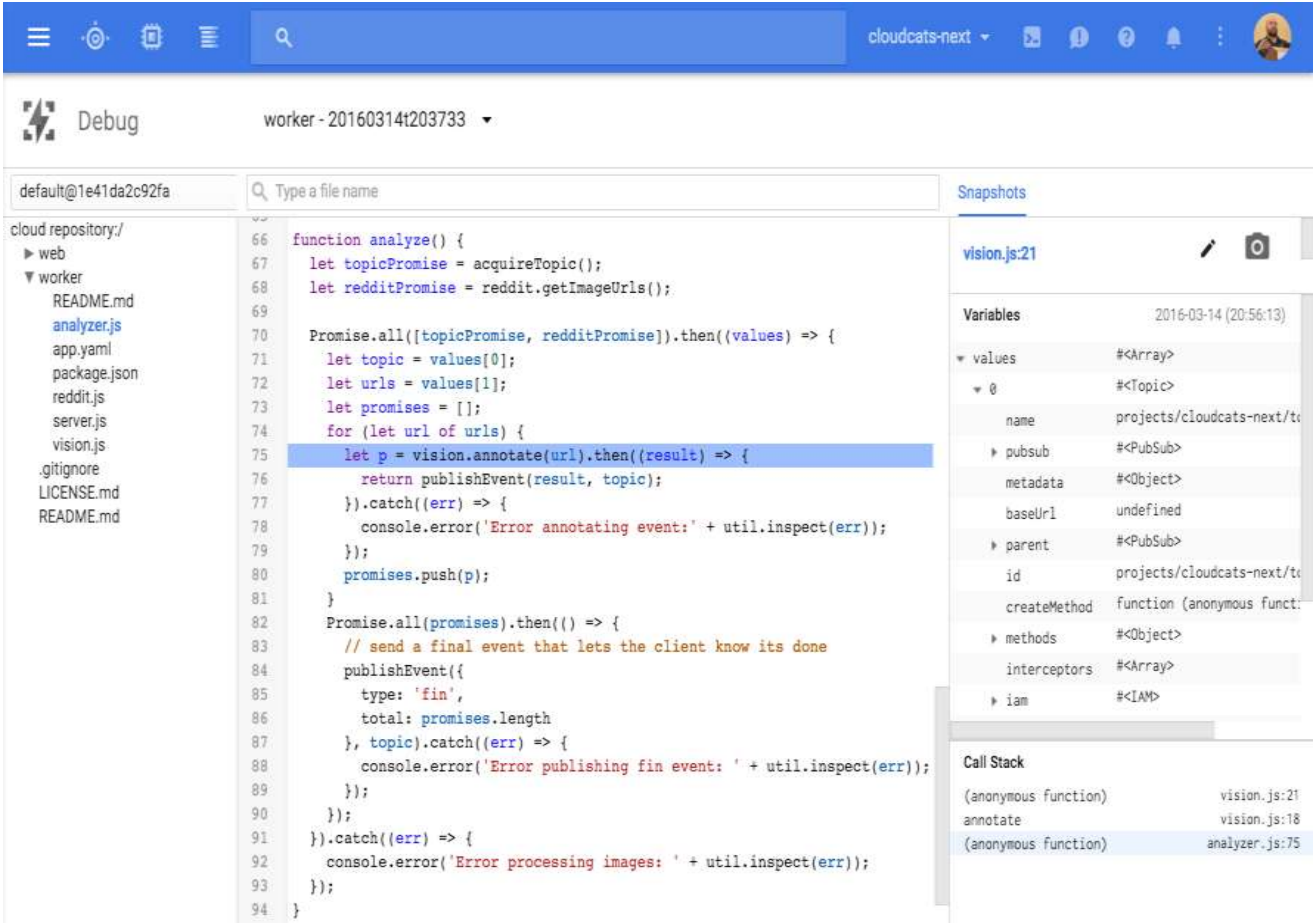
Learn more about AWS Lambda »

- Google App Engine
- Paas
- For Developers



App Engine for All

Build modern web and mobile applications on an open cloud platform: bring your own language runtimes, frameworks, and third party libraries. Google App Engine is a fully managed platform that completely abstracts away infrastructure so you focus only on code. Go from zero to planet-scale and see why some of today's most successful companies power their applications on App Engine.



```
function analyze() {
  let topicPromise = acquireTopic();
  let redditPromise = reddit.getImageUrls();

  Promise.all([topicPromise, redditPromise]).then((values) => {
    let topic = values[0];
    let urls = values[1];
    let promises = [];
    for (let url of urls) {
      let p = vision.annotate(url).then((result) => {
        return publishEvent(result, topic);
      }).catch((err) => {
        console.error('Error annotating event:' + util.inspect(err));
      });
      promises.push(p);
    }
    Promise.all(promises).then(() => {
      // send a final event that lets the client know its done
      publishEvent({
        type: 'fin',
        total: promises.length
      }, topic).catch((err) => {
        console.error('Error publishing fin event: ' + util.inspect(err));
      });
    });
  }).catch((err) => {
    console.error('Error processing images: ' + util.inspect(err));
  });
}
```

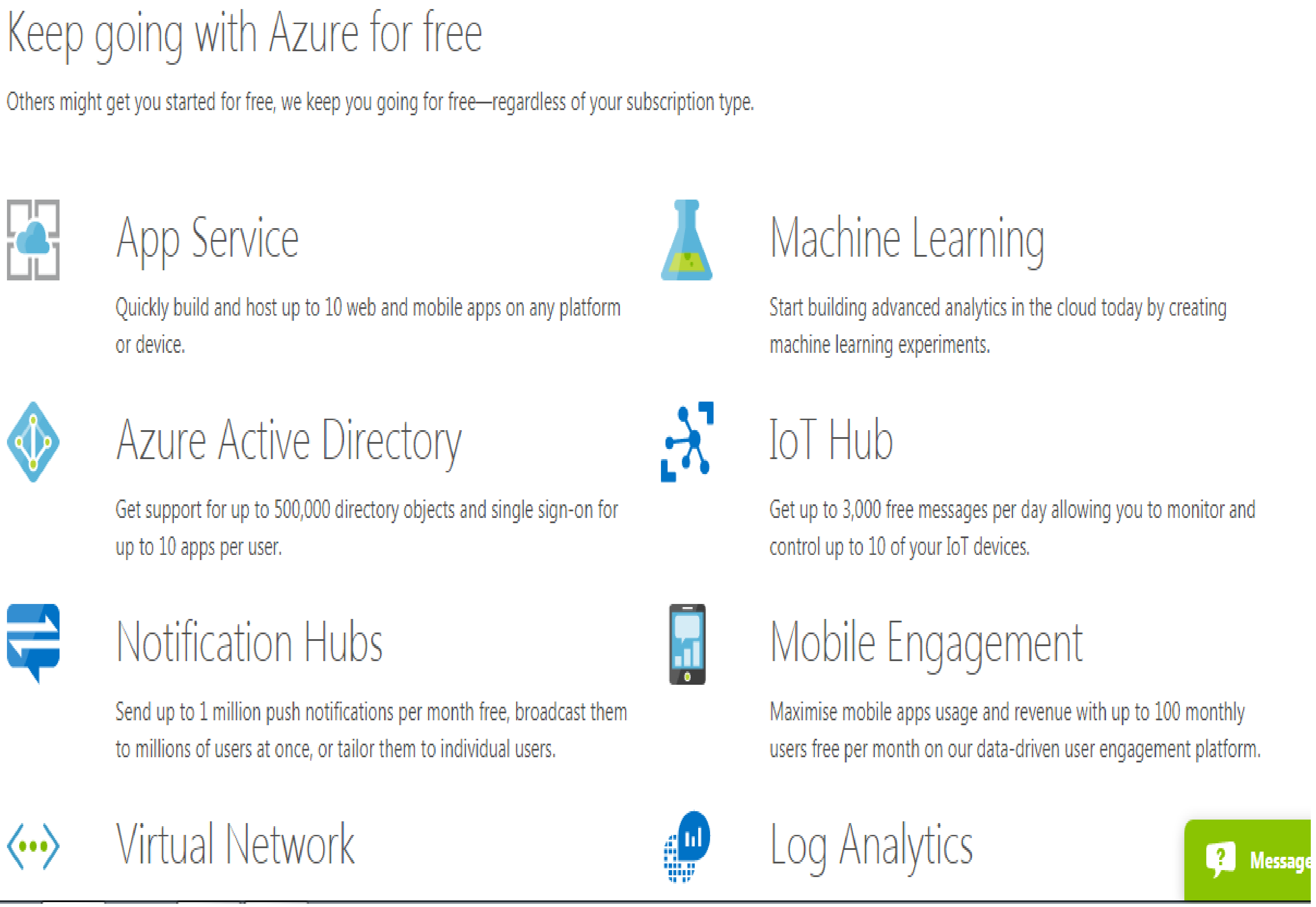
Variables 2016-03-14 (20:56:13)

```
▾ values              #<Array>
  ▾ 0                 #<Topic>
      name            projects/cloudcats-next/t
    ▸ pubsub          #<PubSub>
      metadata        #<Object>
      baseUrl         undefined
    ▸ parent          #<PubSub>
      id              projects/cloudcats-next/t
      createMethod    function (anonymous funct:
    ▸ methods         #<Object>
      interceptors    #<Array>
    ▸ iam             #<IAM>
```

Call Stack

```
(anonymous function)    vision.js:21
annotate                vision.js:18
(anonymous function)    analyzer.js:75
```

- Microsoft Azure
- Paas
- https://azure.microsoft.com



Keep going with Azure for free

Others might get you started for free, we keep you going for free—regardless of your subscription type.

App Service

Quickly build and host up to 10 web and mobile apps on any platform or device.

Machine Learning

Start building advanced analytics in the cloud today by creating machine learning experiments.

Azure Active Directory

Get support for up to 500,000 directory objects and single sign-on for up to 10 apps per user.

IoT Hub

Get up to 3,000 free messages per day allowing you to monitor and control up to 10 of your IoT devices.

Notification Hubs

Send up to 1 million push notifications per month free, broadcast them to millions of users at once, or tailor them to individual users.

Mobile Engagement

Maximise mobile apps usage and revenue with up to 100 monthly users free per month on our data-driven user engagement platform.

Virtual Network

Log Analytics

- Haddop
- **Hadoop** is an open source, Java-based programming framework that supports the processing and storage of extremely large data
- https://cloud.google.com/hadoop
- http://hadoop.apache.org/

hadoop overview

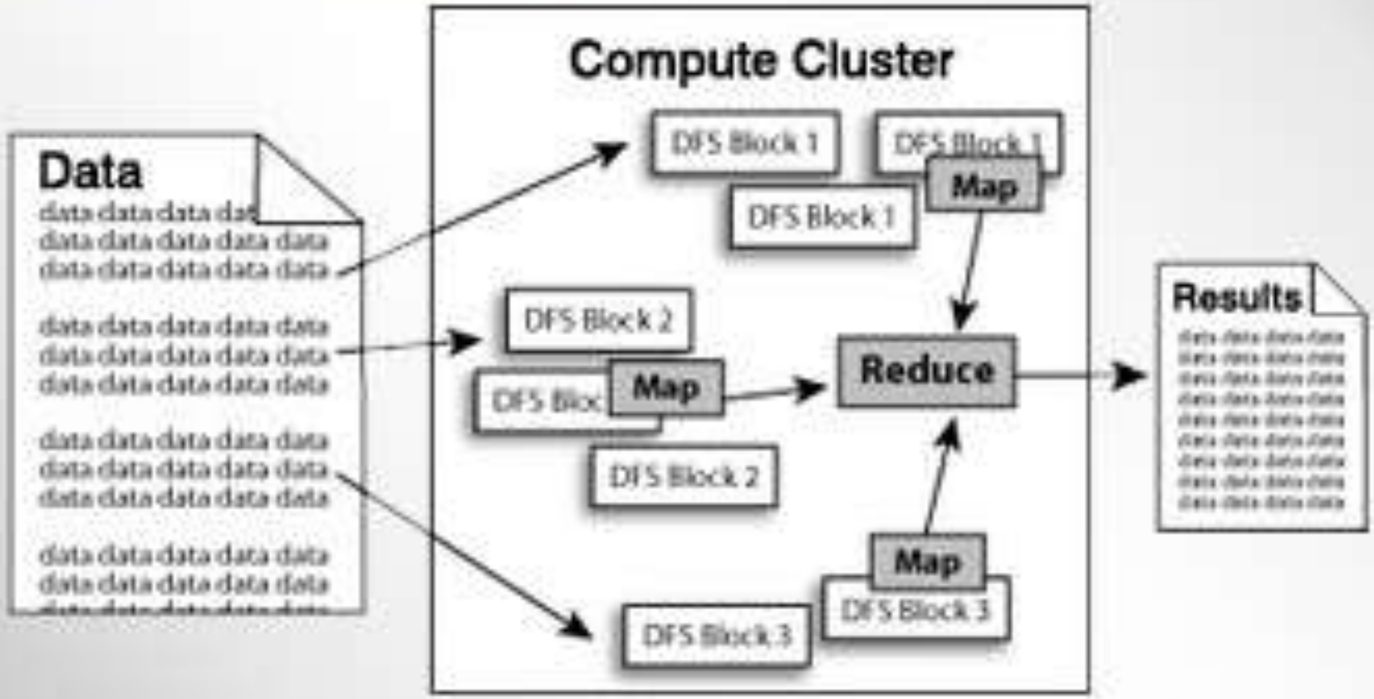
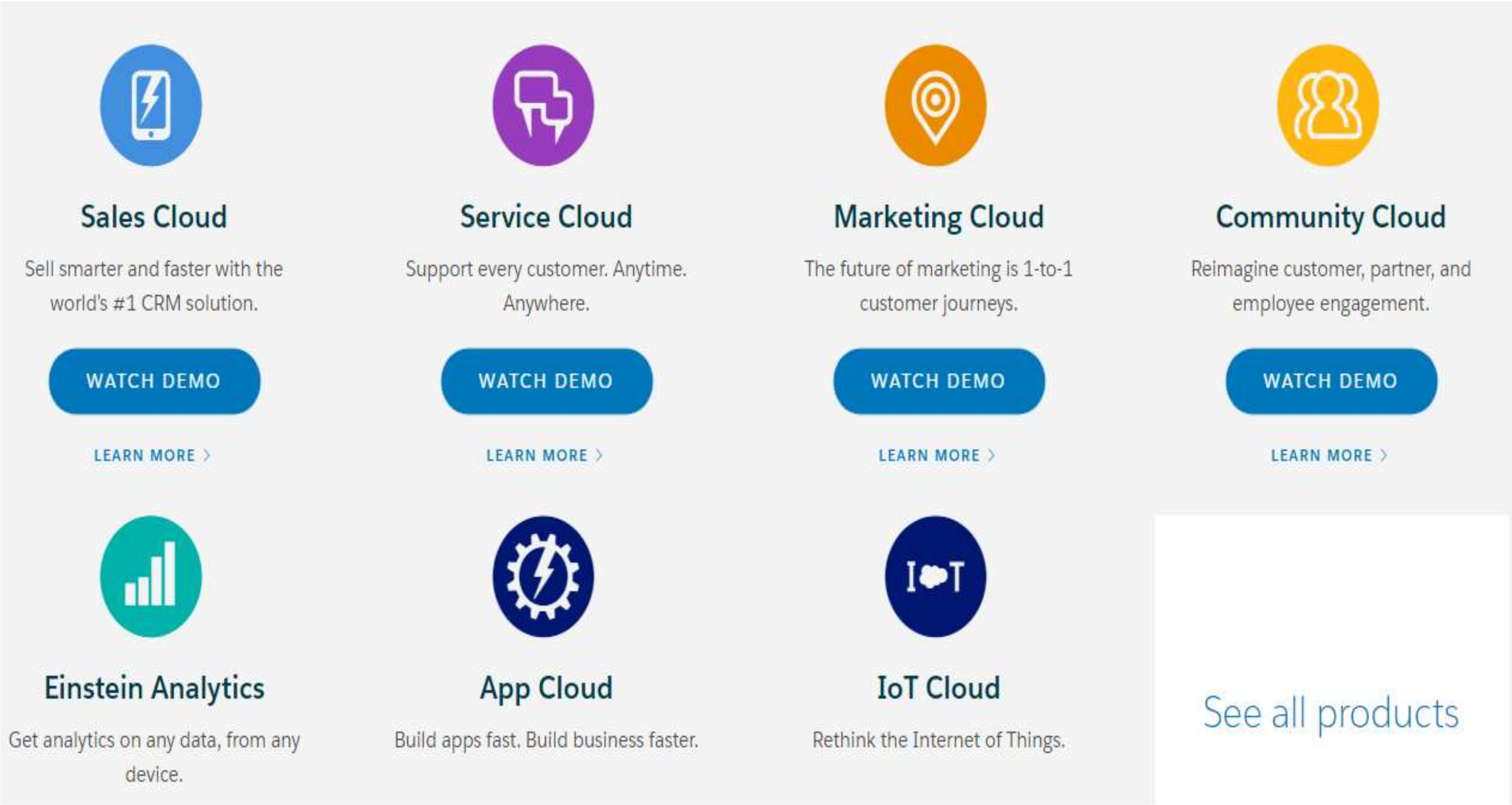


image courtesy of the Apache Software Foundation

- Salesforce
- salesforce.com, inc. is an American cloud computing company headquartered in San Francisco, California.
- https://www.salesforce.com/in/



Sales Cloud

Sell smarter and faster with the world's #1 CRM solution.

WATCH DEMO

LEARN MORE >

Service Cloud

Support every customer. Anytime. Anywhere.

WATCH DEMO

LEARN MORE >

Marketing Cloud

The future of marketing is 1-to-1 customer journeys.

WATCH DEMO

LEARN MORE >

Community Cloud

Reimagine customer, partner, and employee engagement.

WATCH DEMO

LEARN MORE >

Einstein Analytics

Get analytics on any data, from any device.

App Cloud

Build apps fast. Build business faster.

IoT Cloud

Rethink the Internet of Things.

See all products

- Manjarasoft Aneka
- Aneka is a platform and a framework for developing distributed applications on the Cloud.
- One of the key features of Aneka is the ability of providing different ways for expressing distributed applications by offering different programming models;
- http://www.manjrasoft.com/products.html

Chapter 3 - Virtualization

Virtualization

- **Virtualization** is the creation of a virtual rather than actual version of something, such as an operating system, a server, a storage device or network resources
- One of the fundamental Concepts of Cloud Computing

vmware®

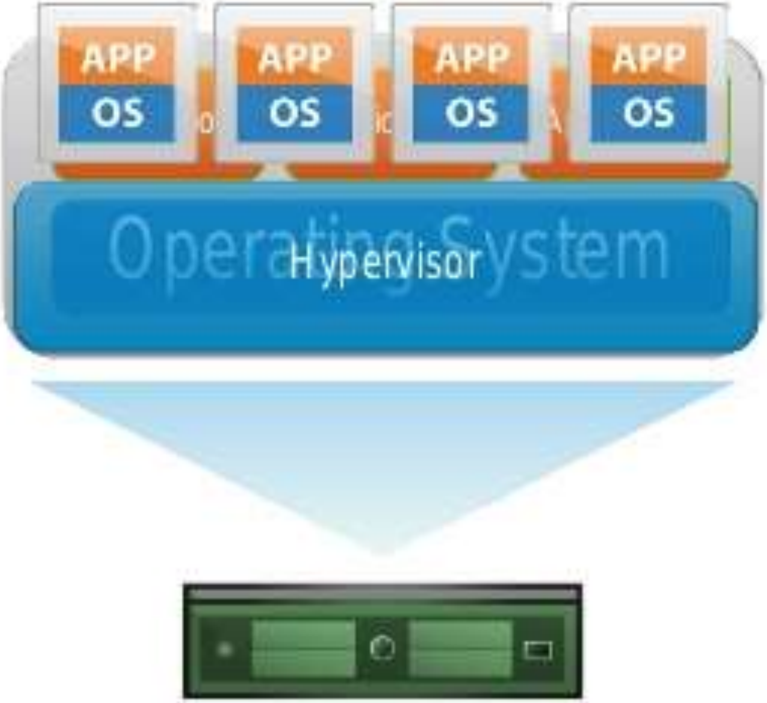
Microsoft® Virtual PC

CITRIX® XenServer

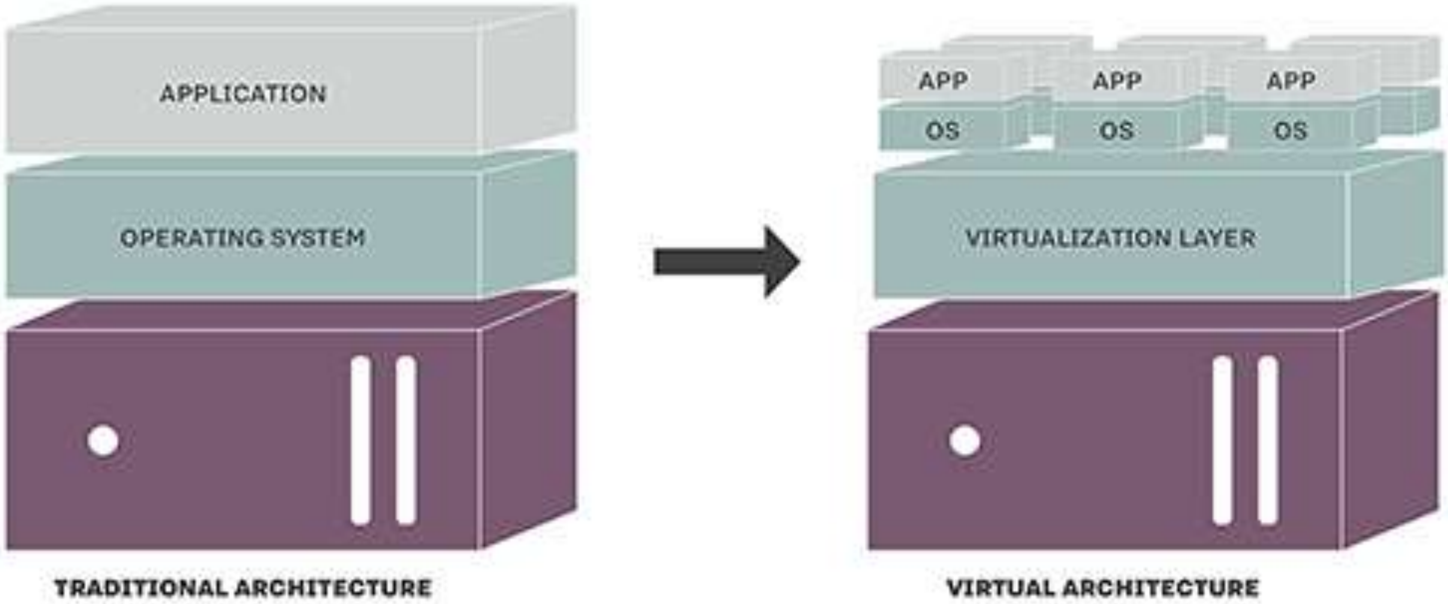
What is Virtualization?

Rockwell Automation

- Traditionally the OS and its applications were tightly coupled to the hardware they were installed on
- Virtualization decouples the operating system from physical hardware
- This allows the ability to change hardware without replacing the OS or applications
- Additionally, multiple instances of an OS with independent applications can now run on the same hardware



TRADITIONAL AND VIRTUAL ARCHITECTURE



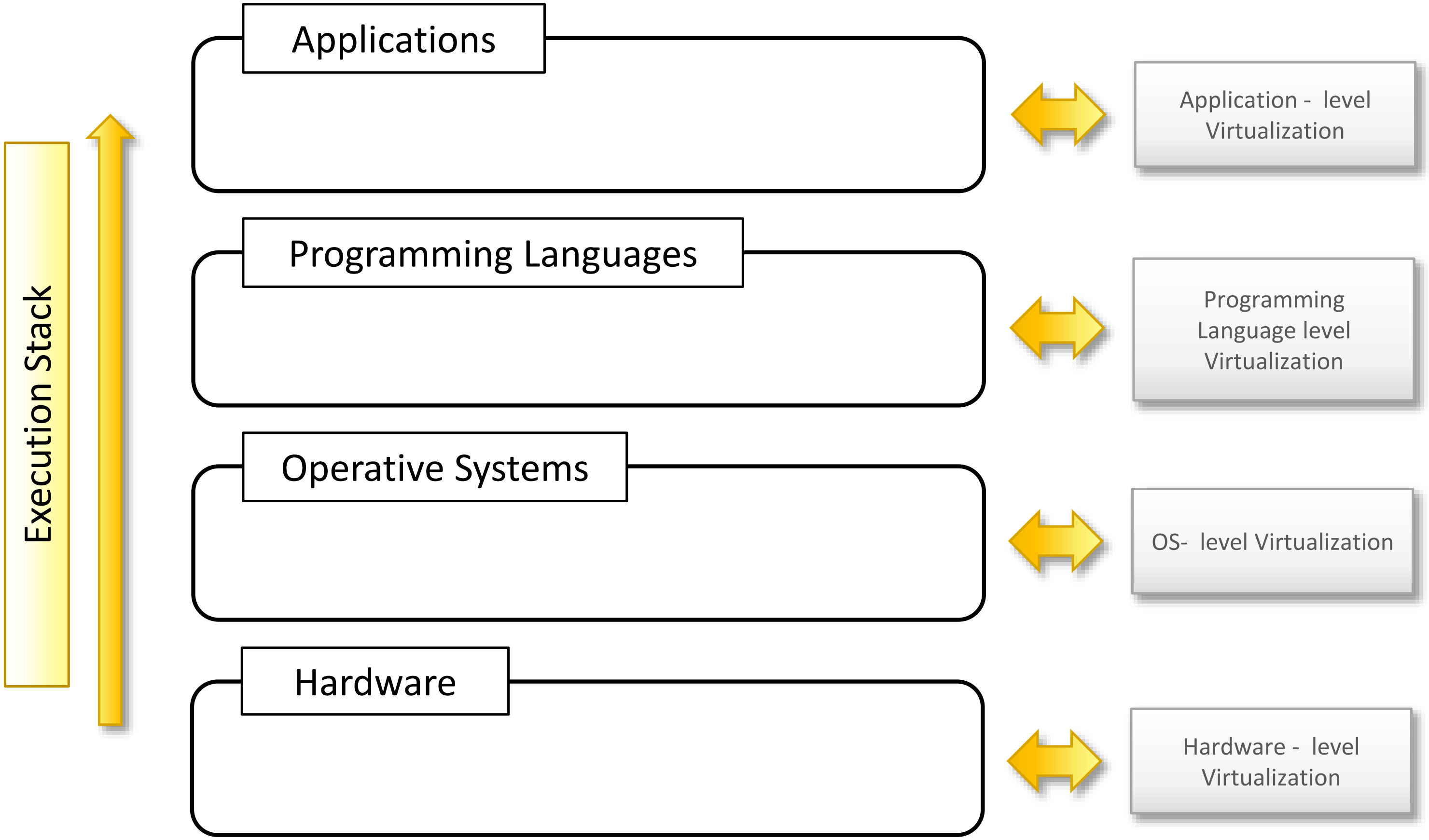
TechTarget

Why are virtualized environments so popular today?

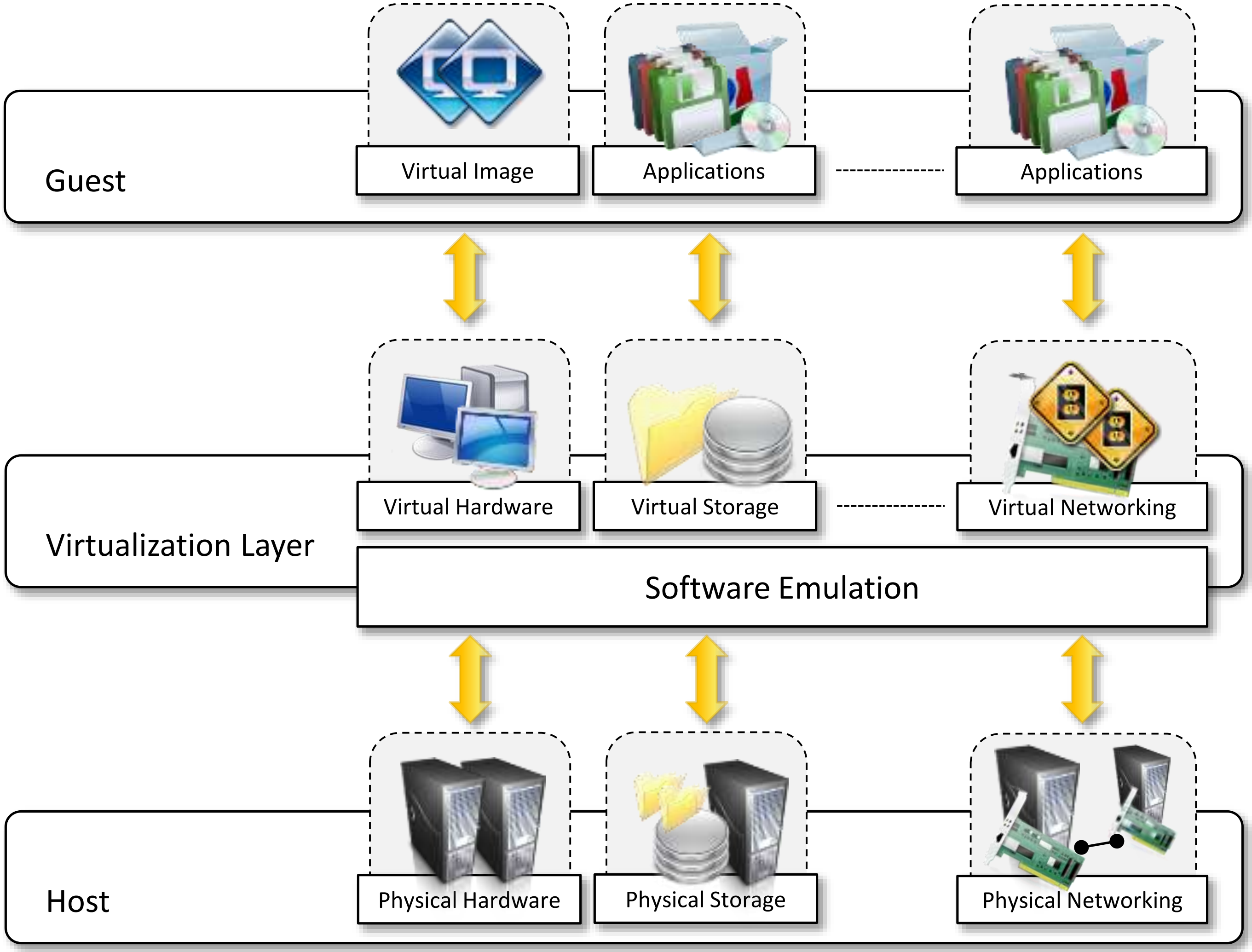
- **Increased performance and computing capacity**
  - PCs are having immense computing power.
- **Underutilized hardware and software resources**
  - Limited use of increased performance & computing capacity.
- **Lack of space**
  - Continuous need for additional capacity.
- **Greening initiatives**
  - Reduce carbon footprints
  - Reducing the number of servers, reduce power consumption.
- **Rise of administrative costs**
  - Power and cooling costs are higher then IT equipments.

Virtualized Environments

- Virtualization is a method of logically dividing the system resources between different applications
- Application Virtualization
- Desktop Virtualization
- Server Virtualization
- Network Virtualization
- Storage Virtualization


Execution Stack
Applications
Programming Languages
Operative Systems
Hardware
Application - level Virtualization
Programming Language level Virtualization
OS- level Virtualization
Hardware - level Virtualization

- Three major components of Virtualized Environments
  - **Guest** – system component that interacts with Virtualization Layer.
  - **Host** – original environment where guest runs.
  - **Virtualization Layer** – recreate the same or different environment where guest will run.



Virtualization Reference Model

Characteristics of VE

- Increased Security
- Managed Execution

✓ - Sharing

✓ - Aggregation

✓ - Emulation

✓ - Isolation

- Portability

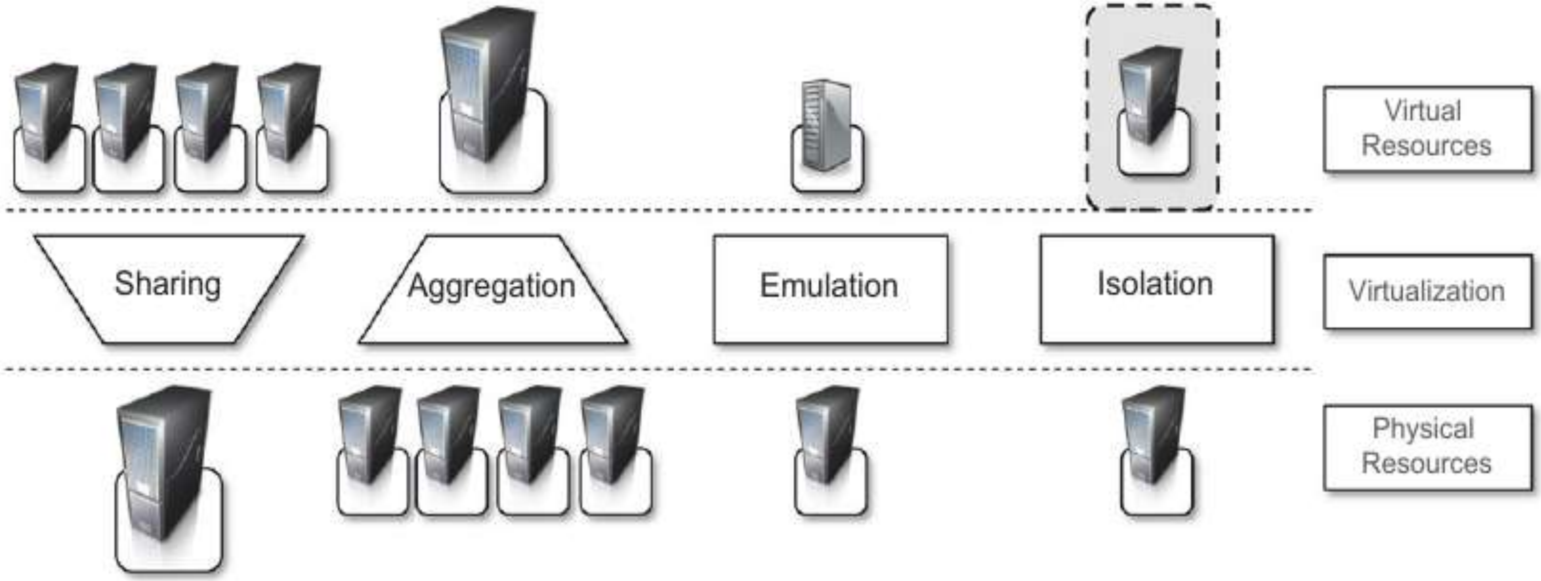
Increased Security

- Ability to control the execution of a guest
- Guest is executed in emulated environment.
- Virtual Machine Manager control and filter the activity of the guest.
- Hiding of resources.
- Having no effect on other users/guest environment.

Managed Execution types

- **Sharing**
  - Creating separate computing environment within the same host.
  - Underline host is fully utilized.
- **Aggregation**
  - A group of separate hosts can be tied together and represented as single virtual host.
- **Emulation**
  - Controlling & Tuning the environment exposed to guest.
- **Isolation**
  - Complete separate environment for guests.

Managed Execution



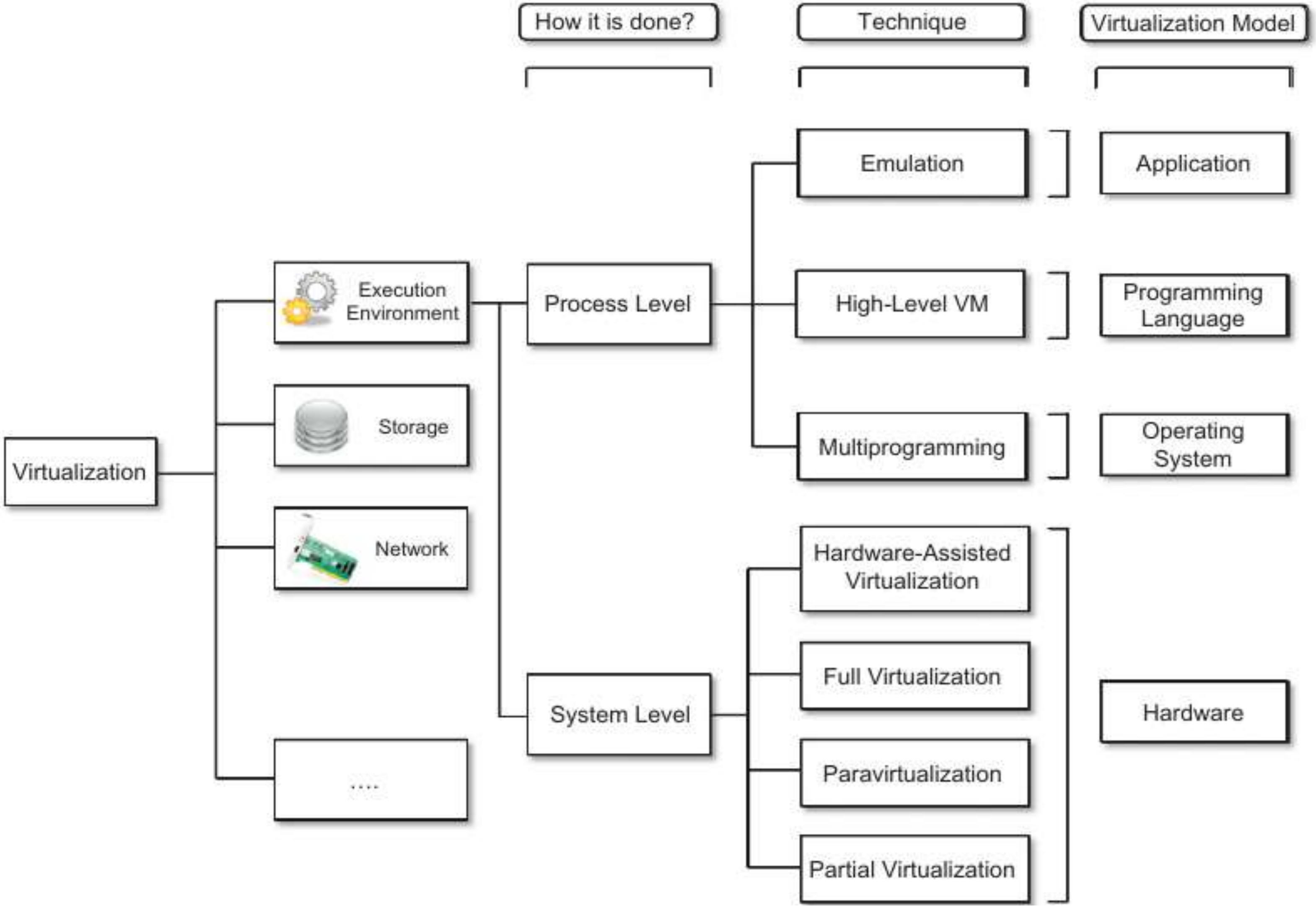
Portability

- safely moved and executed on top of different virtual machine.
- Application Development Cycle more flexible and application deployment very straight forward
- Availability of system is with you.

Taxonomy of Virtualization Techniques

- Virtualization is mainly used to emulate ***execution environment*** , ***storage*** and ***networks***.
- Execution Environment classified into two :-
  - **Process-level** – implemented on top of an existing operating system.
  - **System-level** – implemented directly on hardware and do not or minimum requirement of existing operating system

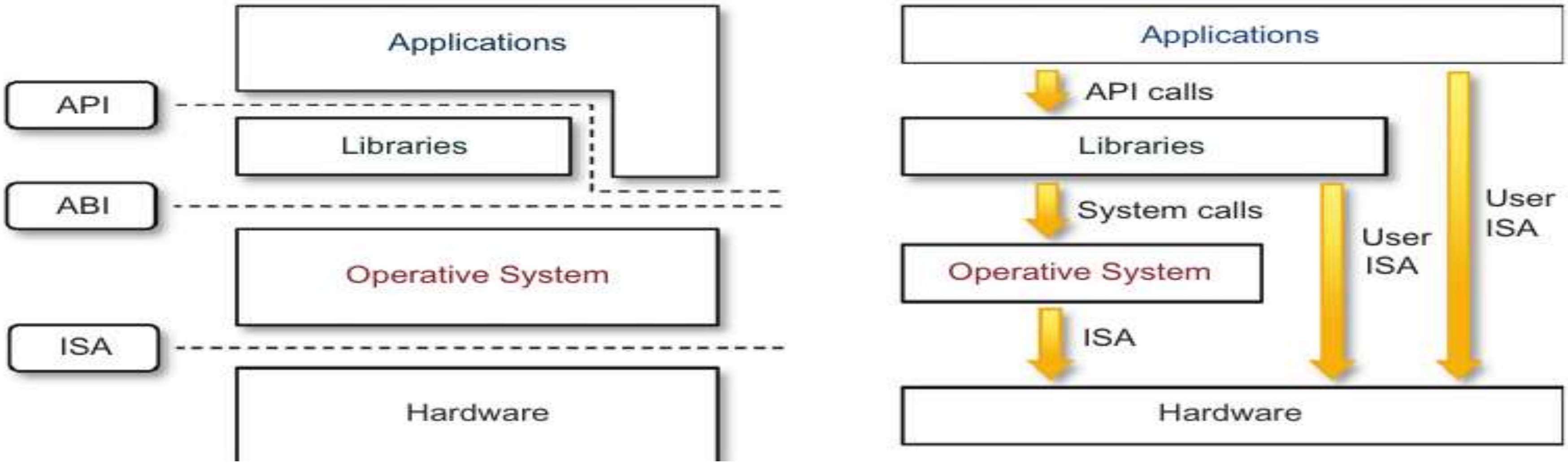
Taxonomy of virtualization



Execution Virtualization

- It defines the ***interfaces between the levels*** of abstractions, which ***hide implementation details***.
- Virtualization techniques actually ***replace one of the layers*** and intercept the calls that are directed towards it.

Machine Reference Model



- Hardware is expressed in terms of the ***Instruction Set Architecture (ISA).***
  - *ISA for processor, registers, memory and the interrupt management.*
- ***Application Binary Interface (ABI***) separates the OS layer from the application and libraries which are managed by the OS.
  - System Calls defined
  - Allows probabilities of applications and libraries across OS.

Machine Reference Model [Cont.]

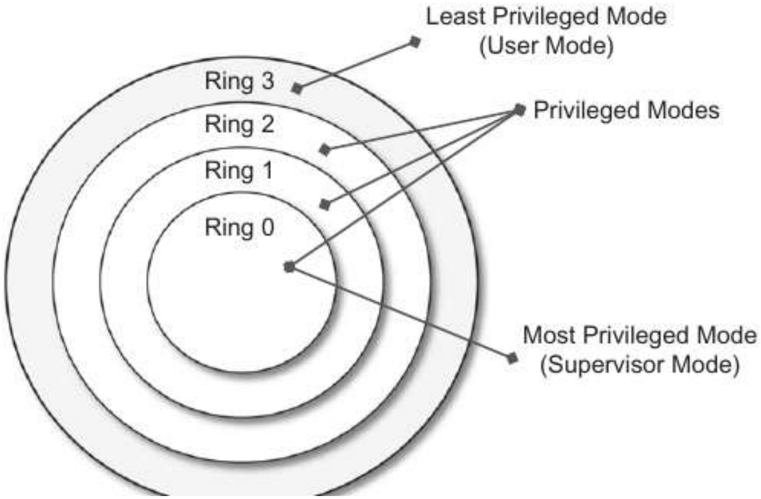
- API – it interfaces applications to libraries and/or the underlying OS.
- Layered approach simplifies the development and implementation of computing system.
- ISA has been divided into two security classes:-
  - ***Privileged Instructions***
  - ***Nonprivileged Instructions***

ISA: Security Classes

- **Nonprivileged instructions**
  - That can be used without interfering with other tasks because they ***do not access shared resources***. Ex. Arithmetic , floating & fixed point.
- **Privileged instructions**
  - That are executed under ***specific restrictions*** and are mostly used for ***sensitive operations***, which expose (*behavior-sensitive*) or modify (*control-sensitive*) the privileged state.
    - **Behavior-sensitive** – operate on the I/O
    - **Control-sensitive** – alter the state of the CPU register.

Privileged Hierarchy: Security Ring

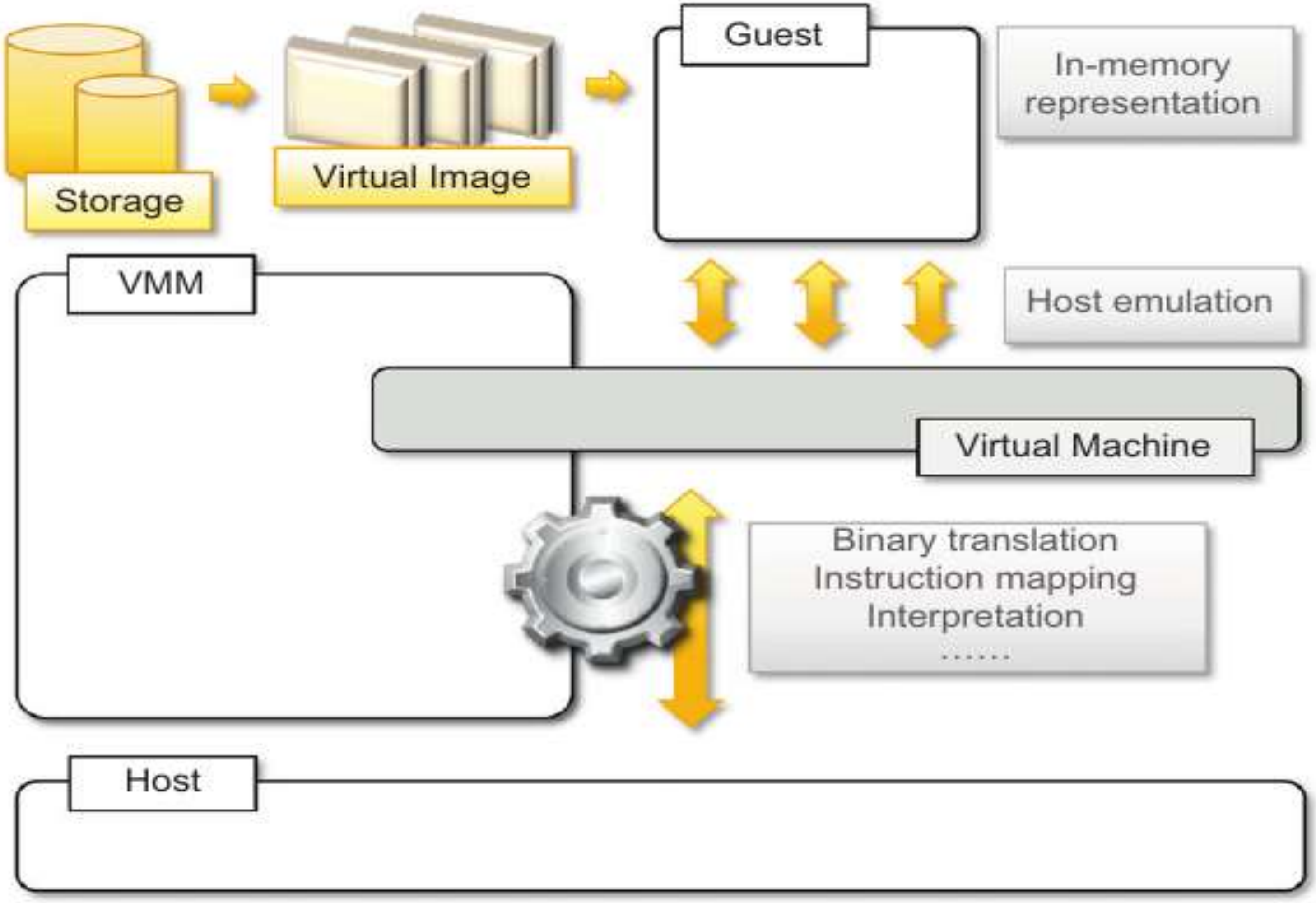
- *Ring-0* is in most privileged level , used by the kernel.
- *Ring-1 & 2* used by the OS-level services
- and , *R3* in the least privileged level , used by the user.
- Recent system support two levels :-
  - **Ring 0 – *supervisor mode***
  - **Ring 3 – *user mode***



Hardware-level virtualization

- It is a virtualization technique that provides an ***abstract execution environmen***t in terms of ***computer hardware*** on top of which a ***guest OS can be run***.
- It is also called as system virtualization.

Hardware-level virtualization

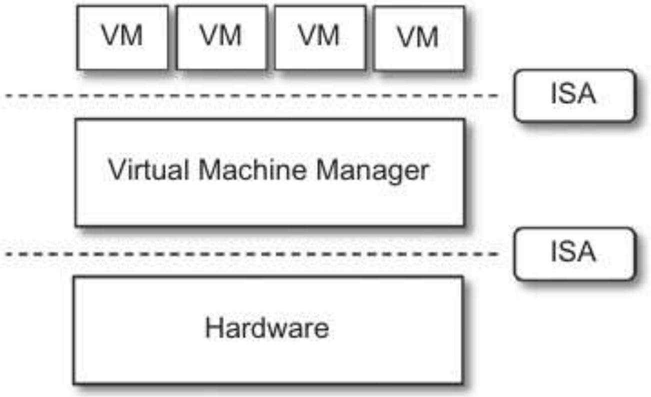


Hypervisor

- Hypervisor runs above the supervisor mode.
- It runs in supervisor mode.
- It recreates a h/w environment.
- It is a piece of s/w that enables us to run one or more VMs on a physical server(host).
- Two major types of hypervisor
  - ***Type -I***
  - ***Type-II***

Type-I Hypervisor

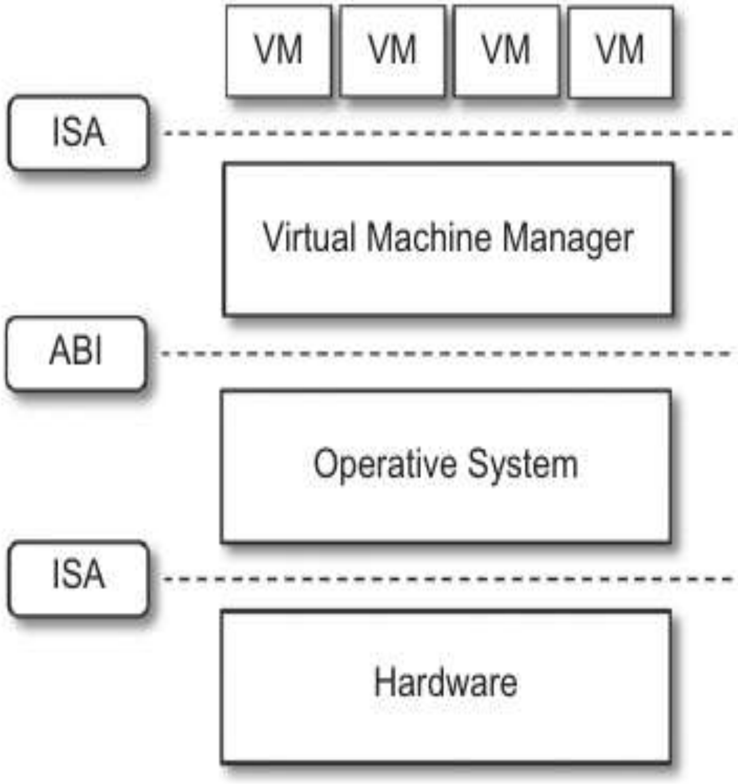
- It runs directly on top of the hardware.
- Takes place of OS.
- Directly interact with the ISA exposed by the underlying hardware.



- Also known as *native virtual machine*.

Type-II Hypervisor

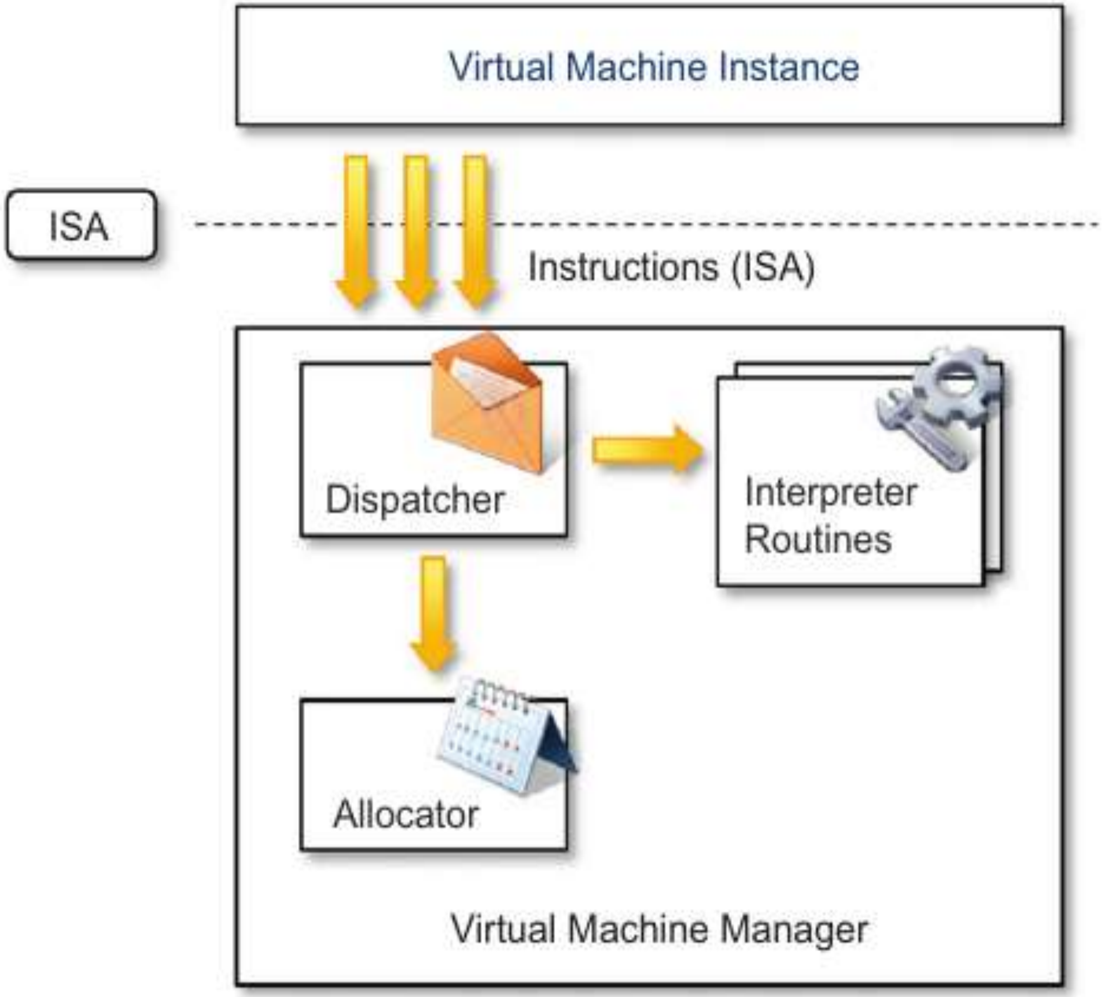
- It require the support of an operating system to provide virtualization services.
- Programs managed by the OS.
- Emulate the ISA of virtual h/w.
- Also called hosted virtual machine.



Virtual Machine Manager (VMM)

- Main Modules :-
  - **Dispatcher**
    - Entry Point of VMM
    - Reroutes the instructions issued by VM instance.
  - **Allocator**
    - Deciding the system resources to be provided to the VM.
    - Invoked by dispatcher
  - **Interpreter**
    - Consists of interpreter routines
    - Executed whenever a VM executes a privileged instruction.
    - Trap is triggered and the corresponding routine is executed.

Virtual Machine Manager (VMM)



Criteria of VMM

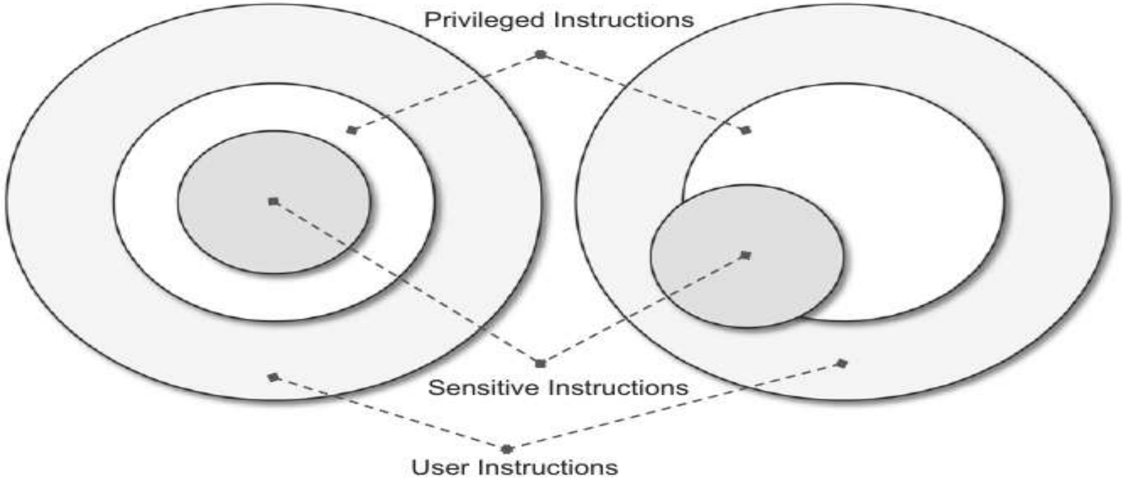
- **Equivalence** – same behavior as when it is ***executed directly*** on the physical host.
- **Resource control** – it should be in ***complete control of virtualized resources***.
- **Efficiency** – a statistically dominant fraction of the machine instructions should be ***executed without intervention*** from the VMM

Theorems

- Popek and Goldberg provided a ***classification of the instruction set*** and proposed three theorems that define the properties that ***hardware instructions need to satisfy*** in order to efficiently support virtualization.
- Classification of IS-
  - Privileged Instructions
    - Trap if the processor is in user mode
  - Control sensitive Instructions

Theorems-1

- Theorems 1
  - For any conventional third-generation computer, a VMM may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions.



Theorems

- Theorems 2
  - A conventional third-generation computers is recursively virtualizable if:
    - It is virtualizable and
    - A VMM without any timing dependencies can be constructed for it.

Theorems

- Theorems 3
  - A hybrid VMM may be constructed third-generation machine in which the set of user-sensitive instructions is a subset of the set of privileged instructions.
  - *In HVM, more instructions are interpreted rather than being executed directly.*

Hardware virtualization Techniques

- CPU installed on the host is only one set, but each VM that runs on the host requires their own CPU.
- It means CPU needs to virtualized, done by hypervisor.

- **Hardware-assisted virtualization**
  - In this hardware provides architectural support for building a VMM able to run a guest OS in complete isolation.
  - ***Intel VT*** and ***AMD V*** extensions.
  - Early products were using ***binary translation to trap some sensitive instructions*** and provide an emulated version

- **Full virtualization**
  - Ability to run program (OS) directly on ***top of a virtual machine*** and without any modification.
  - VMM ***require complete emulation*** of the entire underneath h/w
  - **Advantages**
    - Complete isolation
    - Enhanced security
    - Ease of emulation of different architectures and coexistence
  - ***Key challenge is interception of privileged instructions***

- **Paravirtualization**
  - Not-transparent virtualization
  - Thin VMM
  - Expose software interface to the virtual machine that is slightly modified from the host.
  - Guest OS need to be modified.
  - Simply transfer the execution of instructions which were hard to virtualized, directly to the host.

- **Partial virtualization**
  - Partial emulation of the underlying hardware
  - Not allow complete isolation to guest OS.
  - Address space virtualization is a common feature of comtemporary operating systems.
  - Address space virtualization used in time-sharing system.

Operating system-level virtualization

- It offers the opportunity to create different and ***separated execution environments*** for applications that are managed concurrently.
- No VMM or hypervisor
- Virtualization is in single OS
- OS kernel allows for multiple isolated user space instances
- Good for server consolidation.
- Ex. *chroot , Jails, OpenVZ etc.*

Programming language-level virtualization

- It is mostly used to achieve ***ease of deployment*** of application, ***managed execution*** and ***portability across*** different platform and OS.
- It consists of a virtual machine ***executing the byte code of a program***, which is the result of the ***compilation process***.
- Produce a binary format representing the machine code for an abstract architecture.
- Example
  - Java platform – Java virtual machine (JVM)
  - .NET provides Common Language Infrastructure (CLI)
- They are stack-based virtual machines

Advantage of programming/process-level VM

- Provide ***uniform execution environment*** across different platforms.
- This ***simplifies*** the development and deployment efforts.
- Allow more ***control over the execution*** of programs.
- Security; by filtering the I/O operations
- Easy support for sandboxing

Application-level virtualization

- It is a technique allowing applications to run in ***runtime environments*** that do not ***natively support*** all the features required by such applications.
- In this, applications are not installed in the ***expected runtime environmen***t.
- This technique is most concerned with :-
  - Partial file system
  - Libraries
  - Operating System component emulation

Strategies for Implementation Application-Level Virtualization

- Two techniques:-
  - **Interpretation** -
    - In this every source instruction is ***interpreted*** by an emulator for executing ***native ISA instructions***,
    - ***Minimal*** start up cost but ***huge overhead***.
  - **Binary translation** -
    - In this every source insruction is ***converted to native*** instructions with equivalent functions.
    - Block of instructions ***translated*** , ***cached*** and ***reused***.
    - Large ***overhead cost*** , but over time it is subject to ***better performance***.

Types: Storage Virtualization

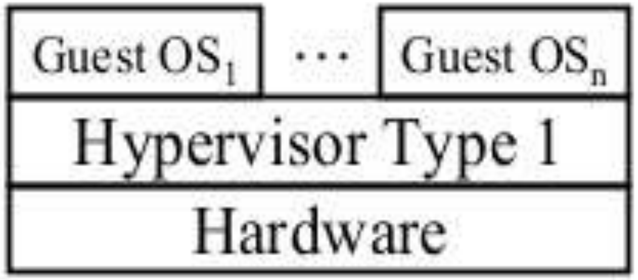
- It allows decoupling the physical organization of the h/w from its logical representation.
- Using Network based virtualization known as ***storage area network*** (SAN).

Network Virtualization

- It combines h/w appliances and specific software for the creation and management of a virtual n/w.
- It can aggregate ***different physical networks*** into a single logical network.

Desktop Virtualization

- A Desktop system with multiple operating systems
- Example: Mac OS X and Windows at the same time Parallels Desktop for Mac
- Hypervisor type 1 similar to server virtualization
- Useful for testing software on multiple OS
- Reduced hardware cost
- This is local desktop virtualization



Guest OS_1 ··· Guest OS_n

Hypervisor Type 1

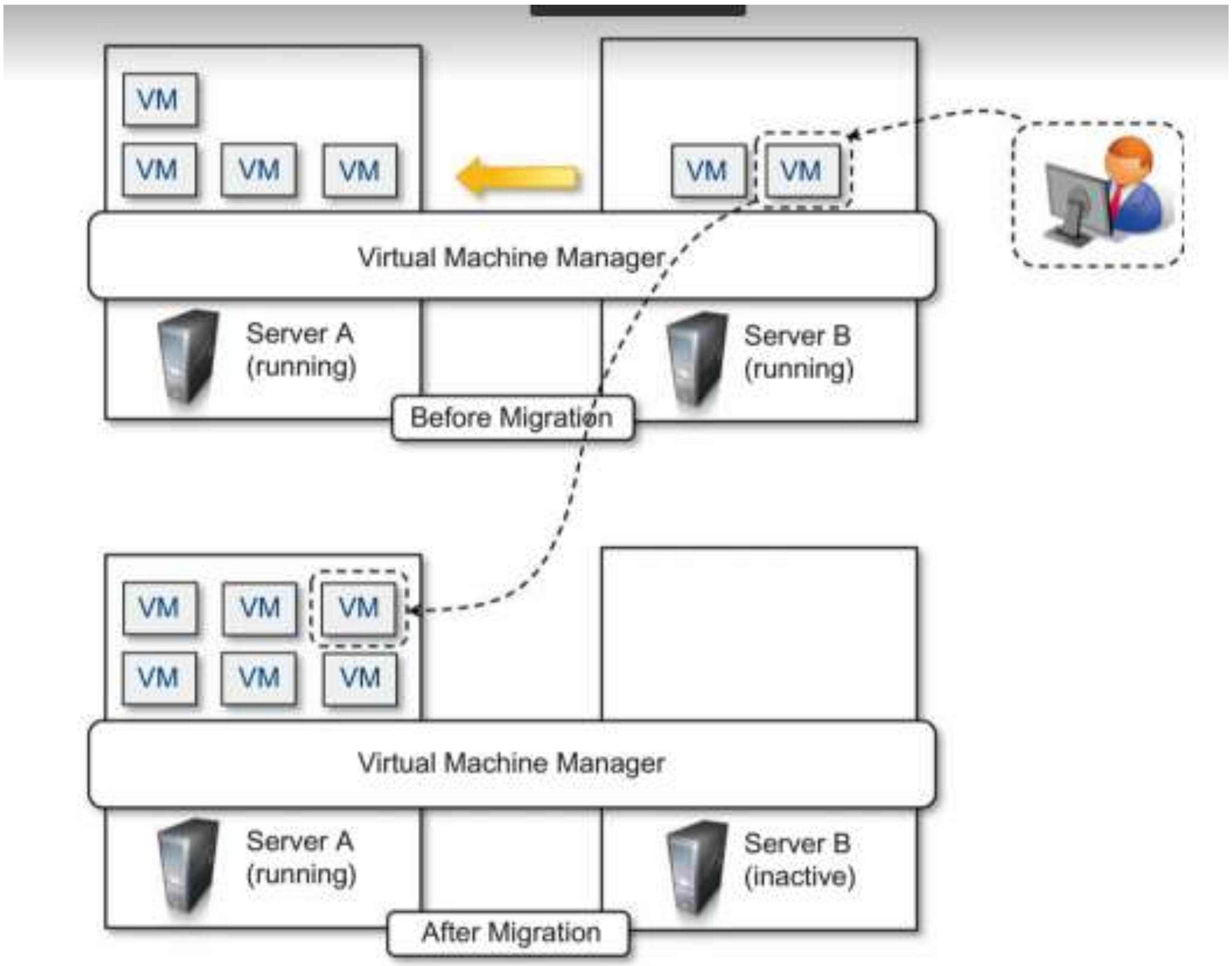
Hardware

Application Server Virtualization

- Application server virtualization abstracts a collection of application servers that provide the same service as a single virtual application server
- Providing better quality of service rather than emulating a different environment

Virtualization and cloud computing

- Virtualization plays an important role in cloud computing
- Virtualization technologies are primarily used to offer configurable computing environments and storage.
- Hardware virtualization is an enabling factor for solutions in the (IaaS) market segment
- programming language virtualization is a technology leveraged in (PaaS) offerings.



Server consolidation and virtual machine migration

Pros and cons of virtualization

- **Advantages of Virtualization**

✓ Reduced spending

✓ Sandbox

✓ Portability

✓ Efficient use of resources.

✓ Easier backup and disaster recovery

✓ Better business continuity

✓ More efficient IT operations

Pros and cons of virtualization

- **Disadvantages of Virtualization**

✓ Upfront costs.

✓ Software licensing considerations

✓ Possible learning curve

✓ Performance degradation

✓ Inefficiency and degraded user experience

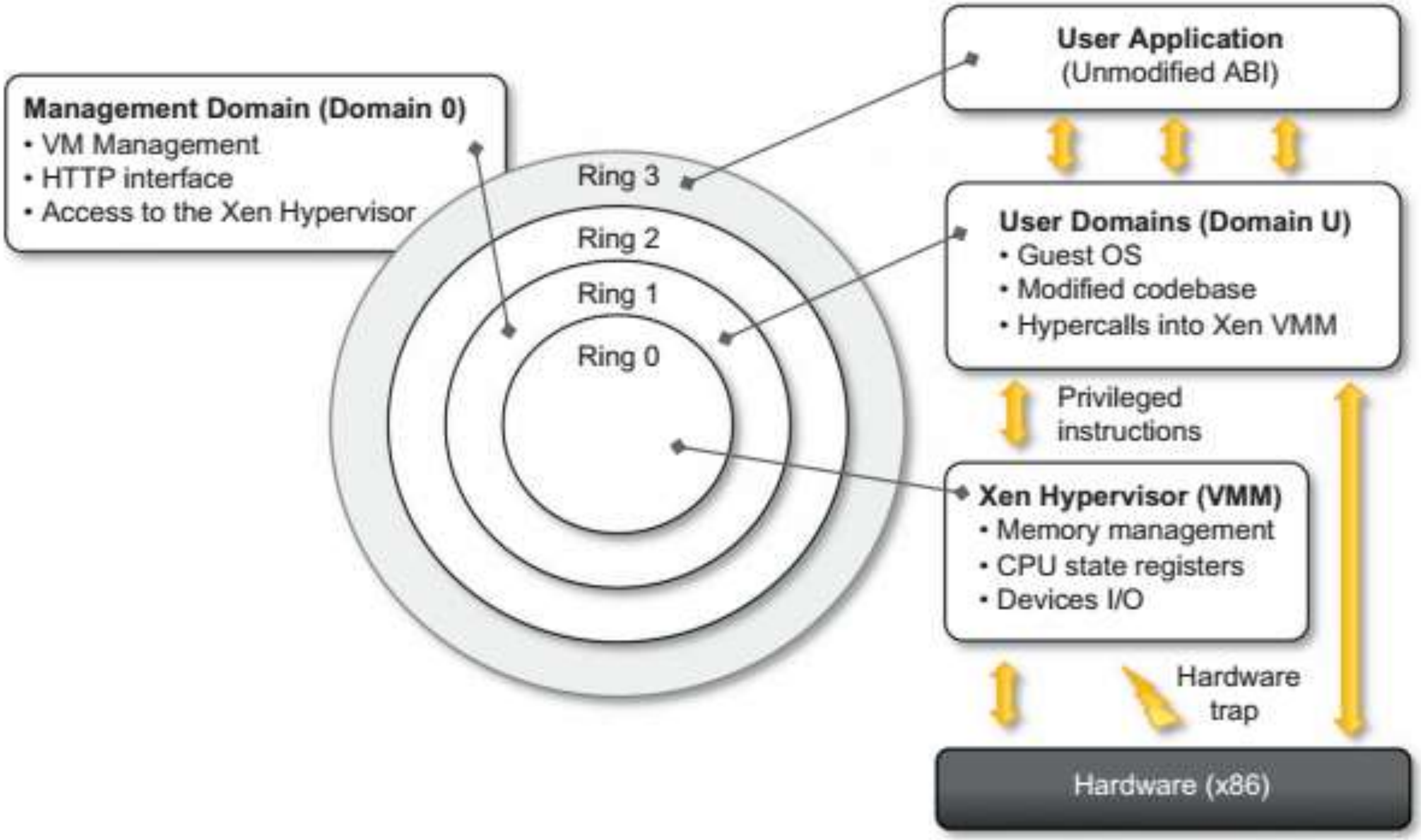
✓ Security holes and new threats

Technology examples

- Xen: paravirtualization
- VMware: full virtualization
- Microsoft Hyper-V

Xen: paravirtualization

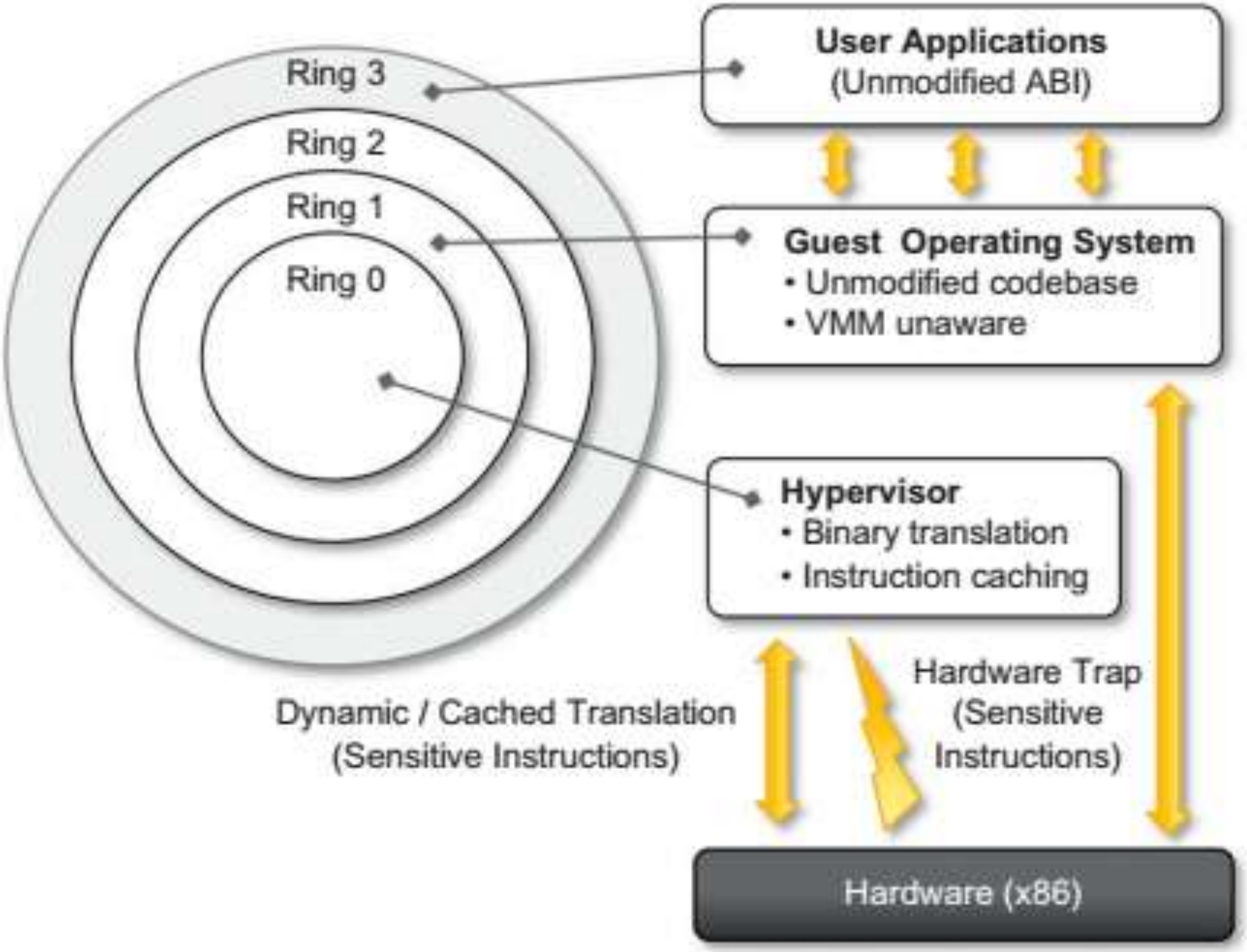
- Xen is an open-source initiative
- Developed by a group of researchers at the University of Cambridge
- XenSource.
- Desktop virtualization or server virtualization
- Xen Cloud Platform (XCP)
- https://www.xenproject.org/



Xen architecture and guest OS management.

VMWare: Full Virtualization

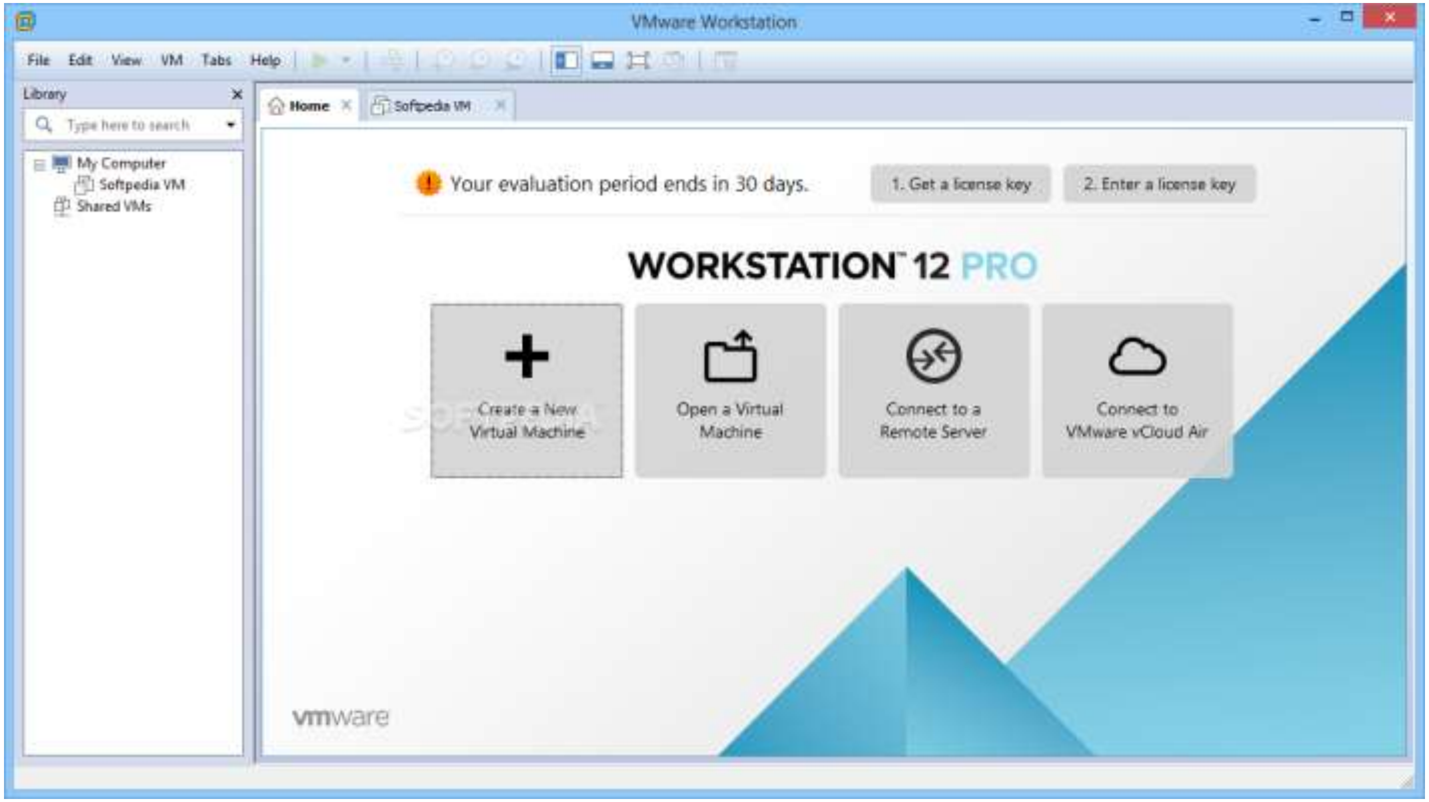
- Underlying hardware is replicated and made available to the guest operating system
- VMware implements full virtualization in the Desktop environments
- Type II hypervisor in Server Environment
- Type I hypervisor in Desktop and Server Environments
- Direct Execution
- Binary Translation



A full virtualization reference model.

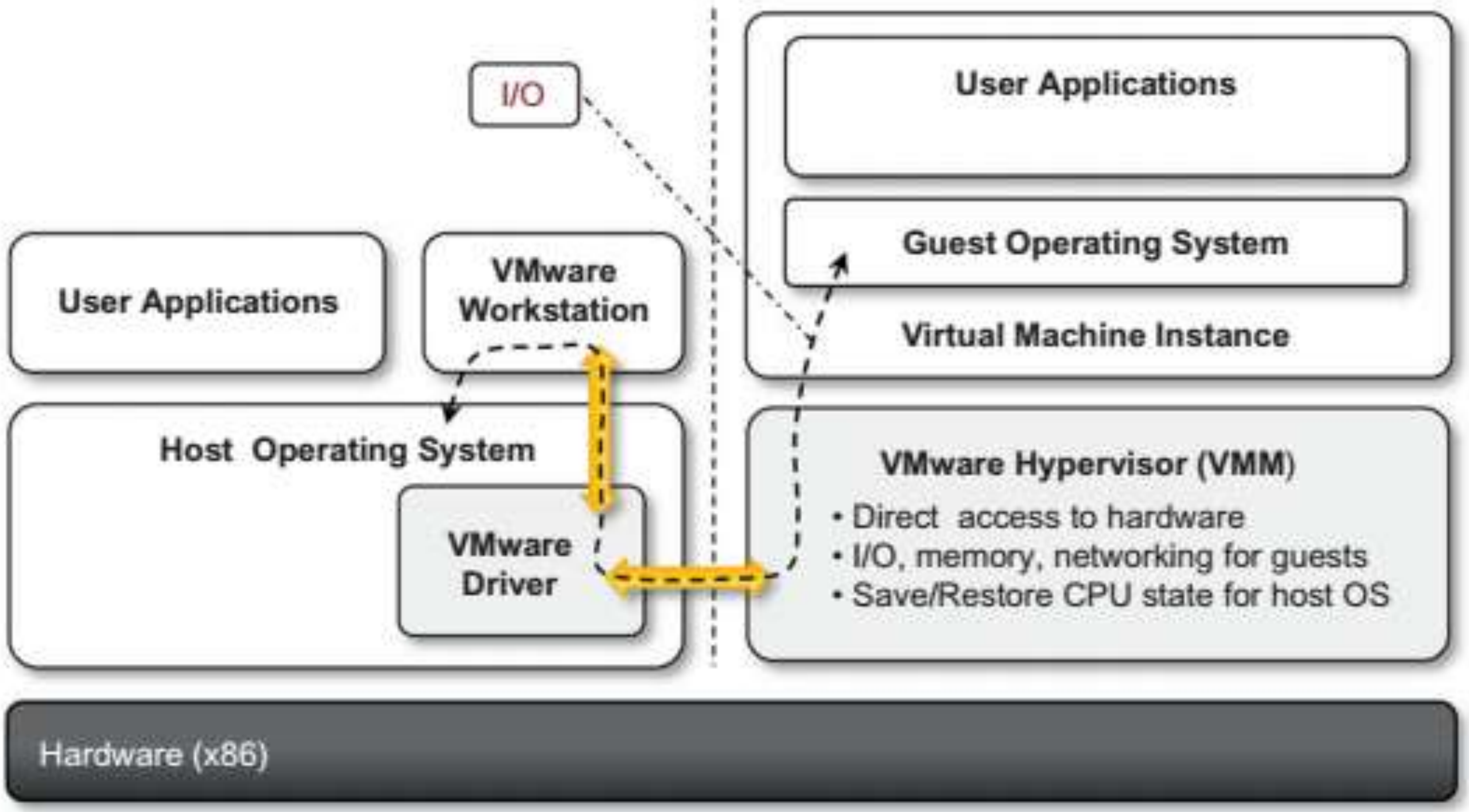
Virtualization solutions by VMware

- End-user (desktop) virtualization



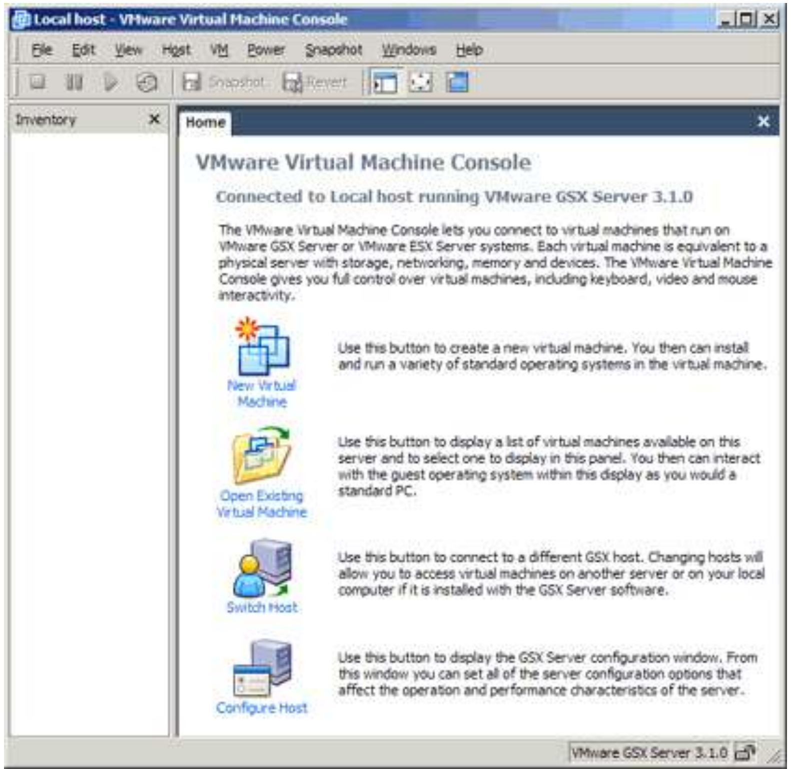
Your evaluation period ends in 30 days.

VMware workstation architecture.



Virtualization solutions by VMware

- Server virtualization
- VMWare GSX
- VMWare ESXi


Local host - VMware Virtual Machine Console
File Edit View Host VM Power Snapshot Windows Help
Snapshot Revert
Inventory
Home
New Virtual Machine
Open Existing Virtual Machine
Switch Host
Configure Host
VMware GSX Server 3.1.0

VMware Virtual Machine Console

Connected to Local host running VMware GSX Server 3.1.0

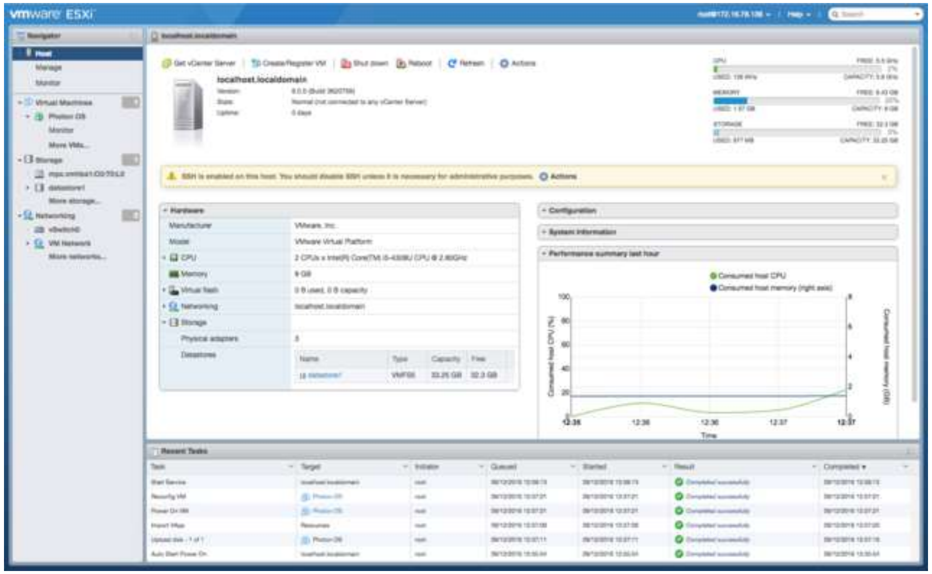
The VMware Virtual Machine Console lets you connect to virtual machines that run on VMware GSX Server or VMware ESX Server systems. Each virtual machine is equivalent to a physical server with storage, networking, memory and devices. The VMware Virtual Machine Console gives you full control over virtual machines, including keyboard, video and mouse interactivity.

Use this button to create a new virtual machine. You then can install and run a variety of standard operating systems in the virtual machine.

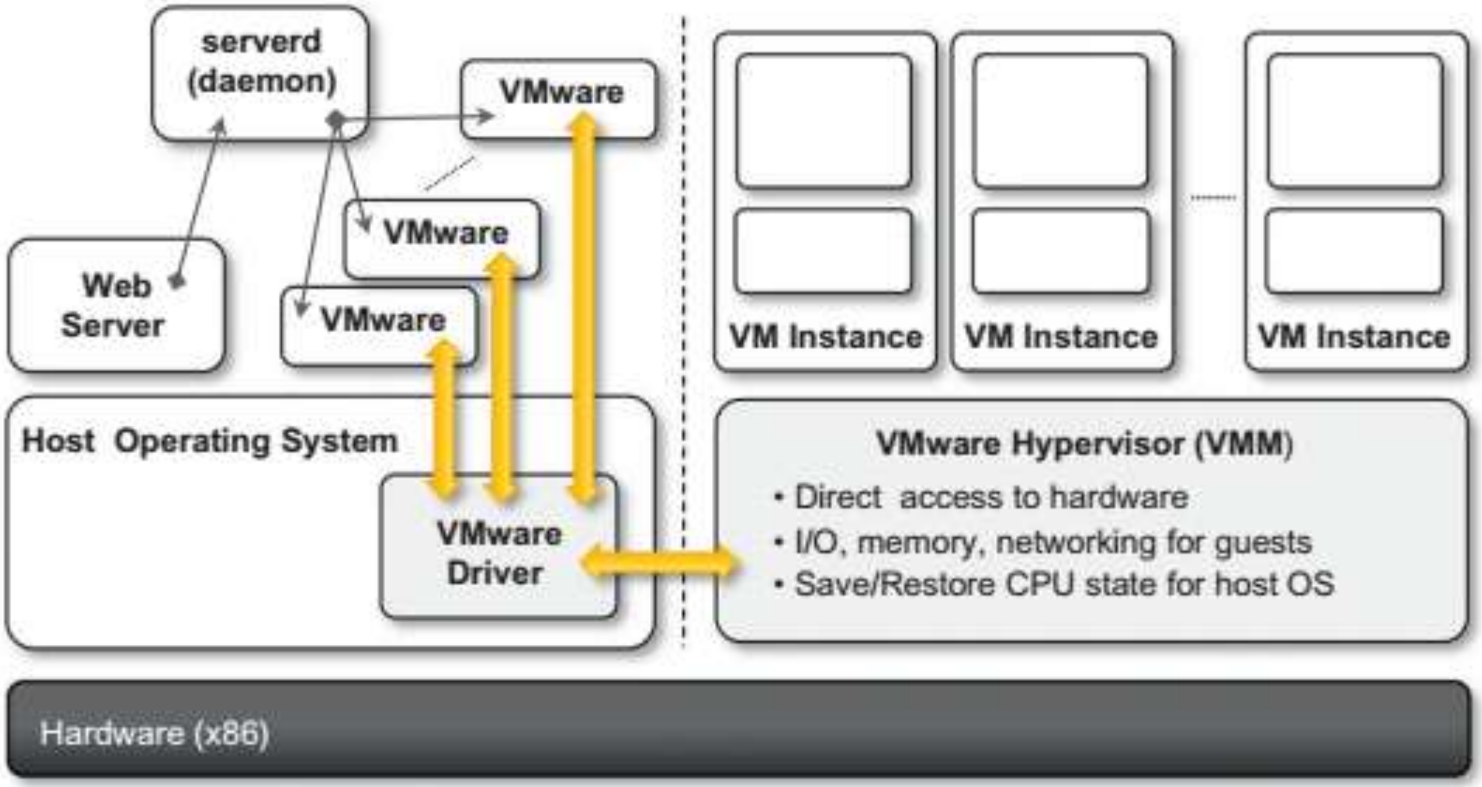
Use this button to display a list of virtual machines available on this server and to select one to display in this panel. You then can interact with the guest operating system within this display as you would a standard PC.

Use this button to connect to a different GSX host. Changing hosts will allow you to access virtual machines on another server or on your local computer if it is installed with the GSX Server software.

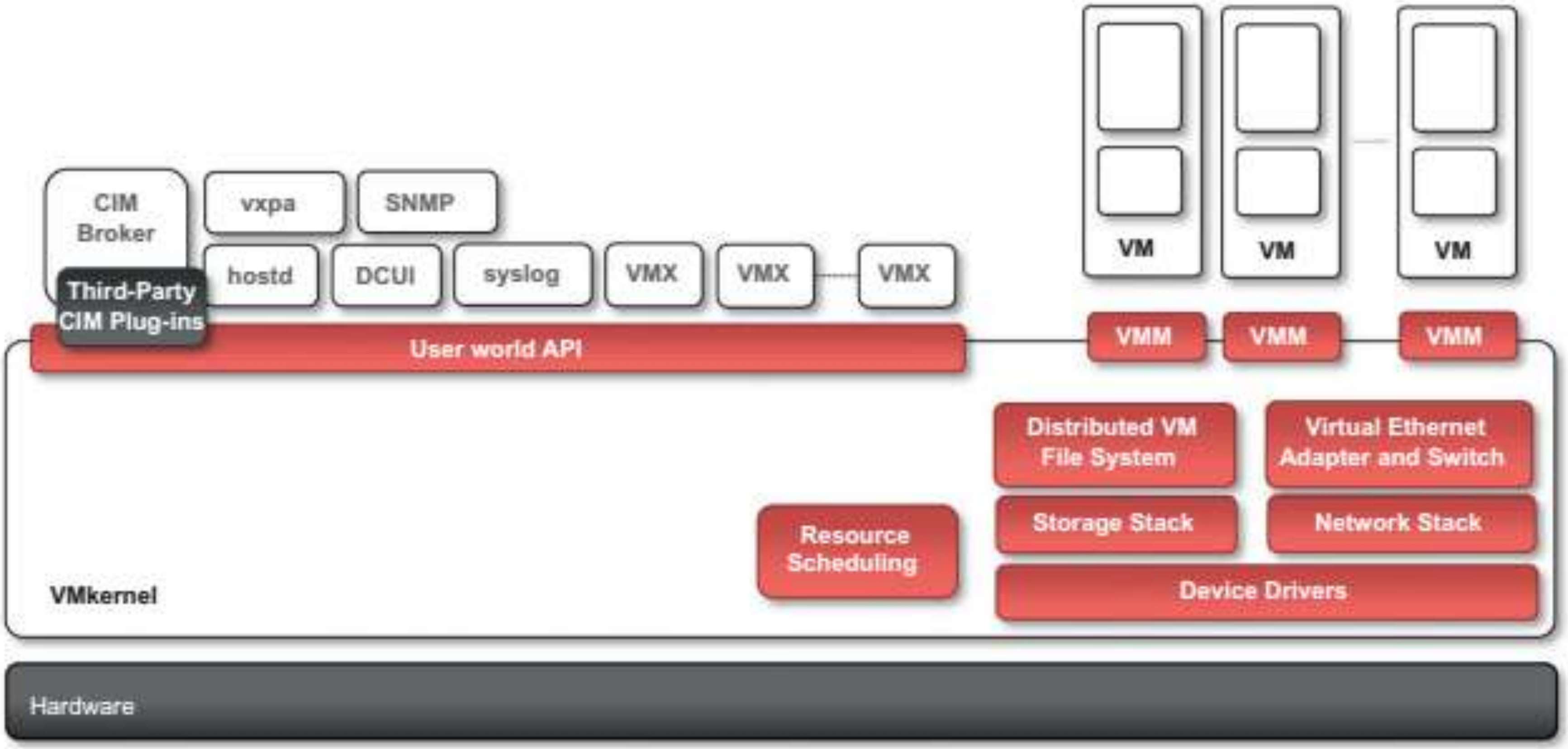
Use this button to display the GSX Server configuration window. From this window you can set all of the server configuration options that affect the operation and performance characteristics of the server.



VMware GSX server architecture.



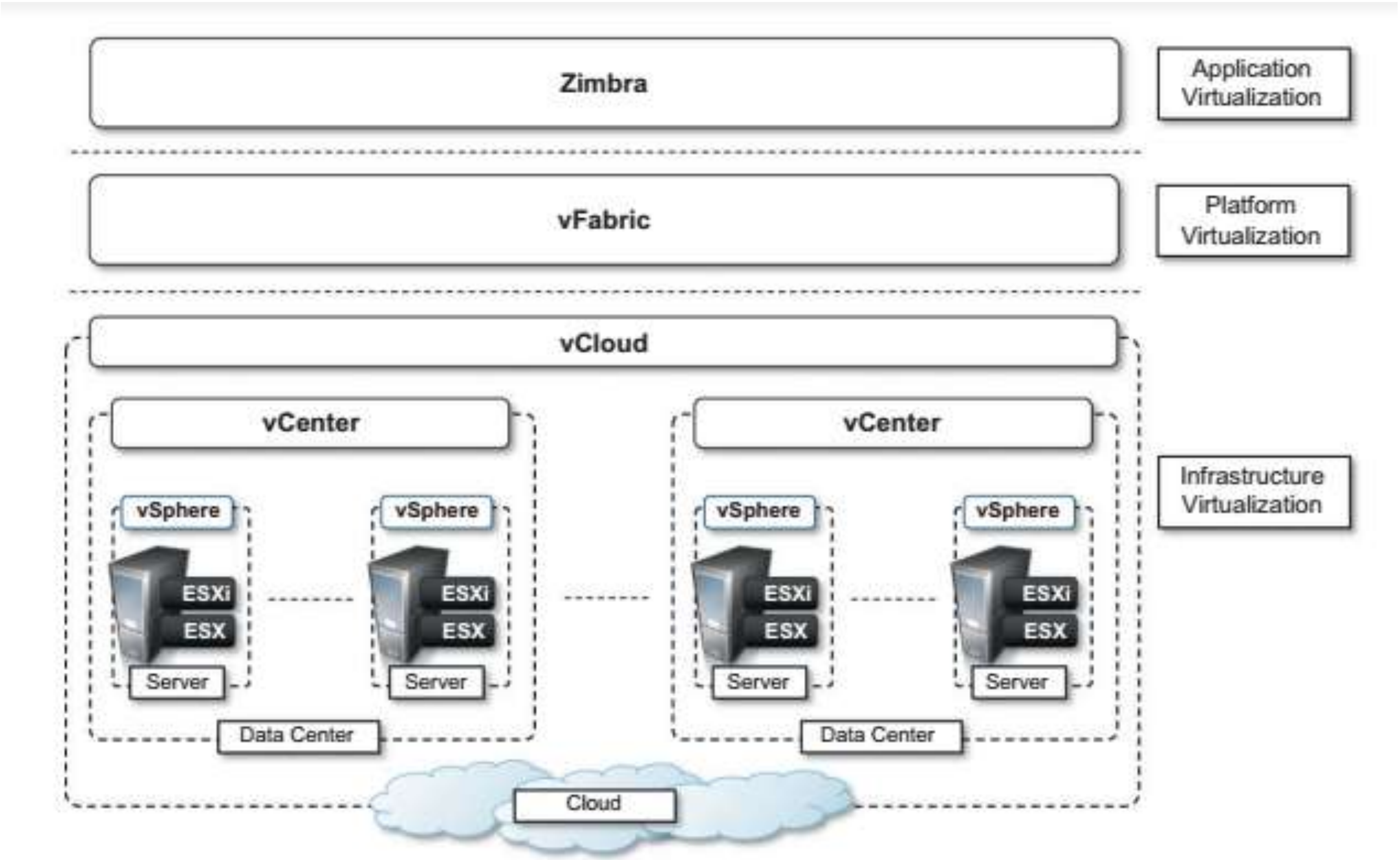
VMware ESXi server architecture.



Virtualization solutions by VMware

- Infrastructure virtualization and cloud computing solutions
- VMware provides a set of products covering the entire stack of cloud computing,

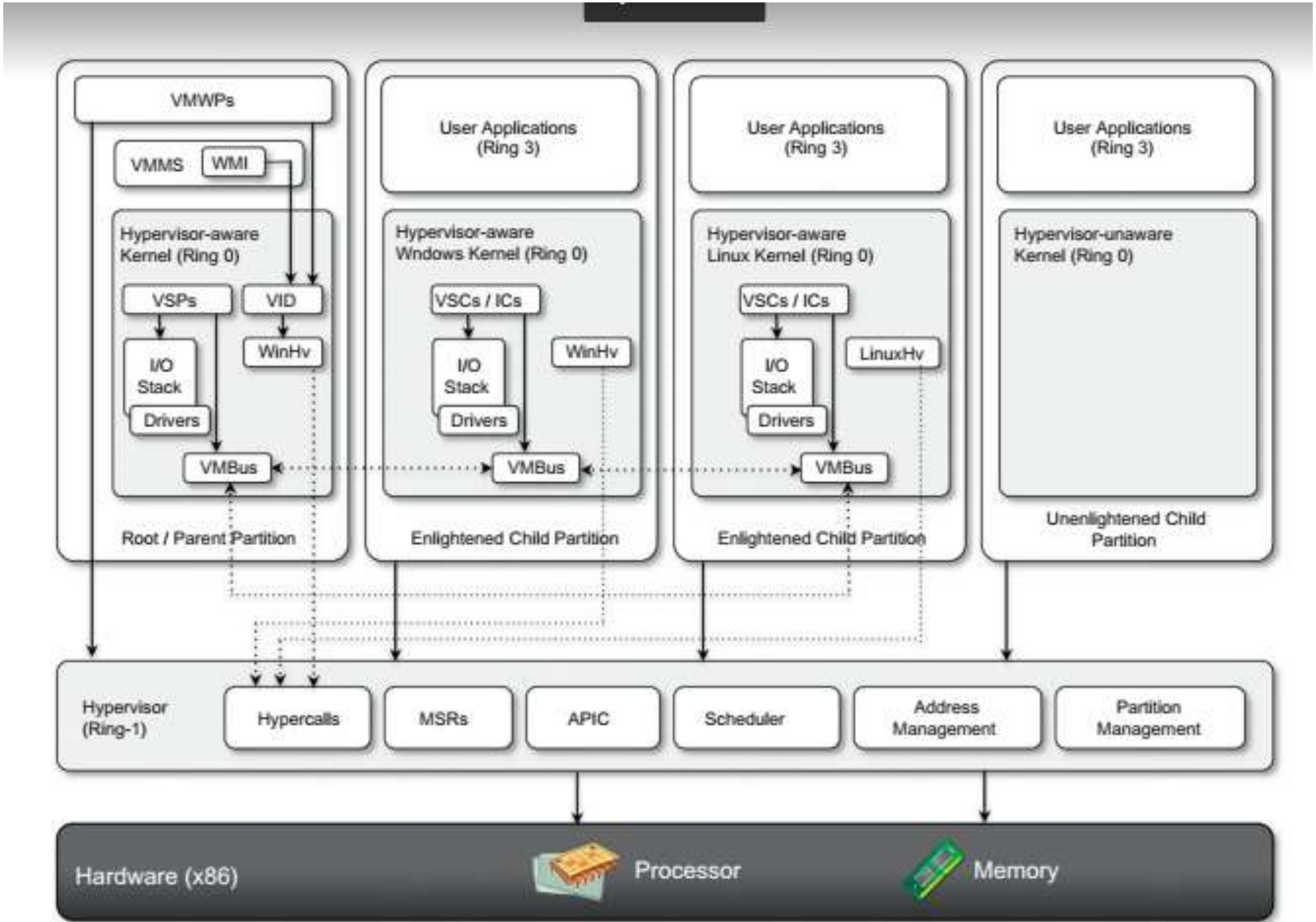
VMware Cloud Solution stack.



Microsoft Hyper-V: Server Virtualization

- formerly known as **Windows Server Virtualization**
- support a variety of guest operating systems.

Microsoft Hyper-V architecture.


VMWPs
VMMS
WMI
Hypervisor-aware Kernel (Ring 0)
VSPs
VID
I/O Stack
WinHv
Drivers
VMBus
Root / Parent Partition
User Applications (Ring 3)
Hypervisor-aware Wndows Kernel (Ring 0)
VSCs / ICs
I/O Stack
WinHv
Drivers
VMBus
Enlightened Child Partition
User Applications (Ring 3)
Hypervisor-aware Linux Kernel (Ring 0)
VSCs / ICs
I/O Stack
LinuxHv
Drivers
VMBus
Enlightened Child Partition
User Applications (Ring 3)
Hypervisor-unaware Kernel (Ring 0)
Unenlightened Child Partition
Hypervisor (Ring-1)
Hypercalls
MSRs
APIC
Scheduler
Address Management
Partition Management
Hardware (x86)
Processor
Memory