Cloud Computing

Course Faculty: Dr. Bheemappa Halavar

(Bheemappa.h@iiits.in)

Course Outcomes

- To describe the different cloud computing models and underlying technologies
- To develop real world applications using cloud computing platforms and containerization technologies
- To implement solutions to complex problems using distributed computing technologies
- To identify and analyze security issues in cloud computing

Tentative List of Topics

- **Introduction to Cloud Computing and its Enabling Technologies :** Evolution of Computing, Grid and Utility Computing, The vision of Cloud Computing, Characteristics and Benefits, Distributed Computing, Virtualization, Web 2.0, Service Oriented Architecture
- **Cloud Computing Architecture** : Introduction, Cloud Service Models, Cloud Deployment Models, Open Challenges
- **Virtualization:** Introduction, Characteristics of virtualized environments, Taxonomy of virtualization techniques, Virtualization and cloud computing, Pros and cons of virtualization, Technology examples, Containers and Applications
- **Cloud Platforms:** Amazon Web Services, Google Cloud Platform, Microsoft Azure, Aneka, OpenStack, Cloud Automation using CHEF/Ansible
- **Introduction to Bigdata:** Bigdata concepts, terminology, NoSQL; Distributed File Systems-Hadoop File System, GFS, Introduction to MapReduce and applications
- **Cloud Security:** Security Issues in Cloud Computing, Hypervisor and VM Security, Data Security in Cloud Environment, Identity and Access Management in Cloud

Tentative Evaluation Policy

- **Examinations: 55%**
  - Mid Semester Exam: 20%
  - End Semester Exam: 35%
- **Research Work / Assignments: 20%**
- **Project: 10%**
- **Scheduled Quizzes: 15%**

So, let us begin…

What is **Computing**?

Pascaline

Charles Babbage

Computing

- **Computing** consists of three things:
  - Managing,
  - Processing, and
  - Communicating information
- Over the years, computing has evolved – passing through a number of computational paradigms
  - More computational capacity
  - More efficiency
  - Less cost
  - Less size

Vacuum Tube

Transistors

Integrated Circuit

Microprocessor

Quantum Computer

1st Generation Computer

2nd Generation Computer

3rd Generation Computer

4th Generation Computer

5th Generation Computer

The Evolution of Commercial Computing

Centralized Era

first, large and expensive Computers

(since 1940s)

Microprocessors & Mass Production

Personal Computers

(since 1970s)

Miniaturization & Mobility needs

Notebook Computers

(since 1980s)

Timesharing Operating Systems & Local Area Networks

dedicated, on-premise Data Centers

(since 1980s)

Internet

Decentralized Era

Public Data Centers

(since late 1990s)

Service-oriented Architecture

Public Cloud Computing

(since early 2000s)

Blockchain

Distributed Computing (with limited use cases)

(in development since 2016)

Artificial Intelligence

Distributed Computing (with flexible use cases)

(in development since 2018)

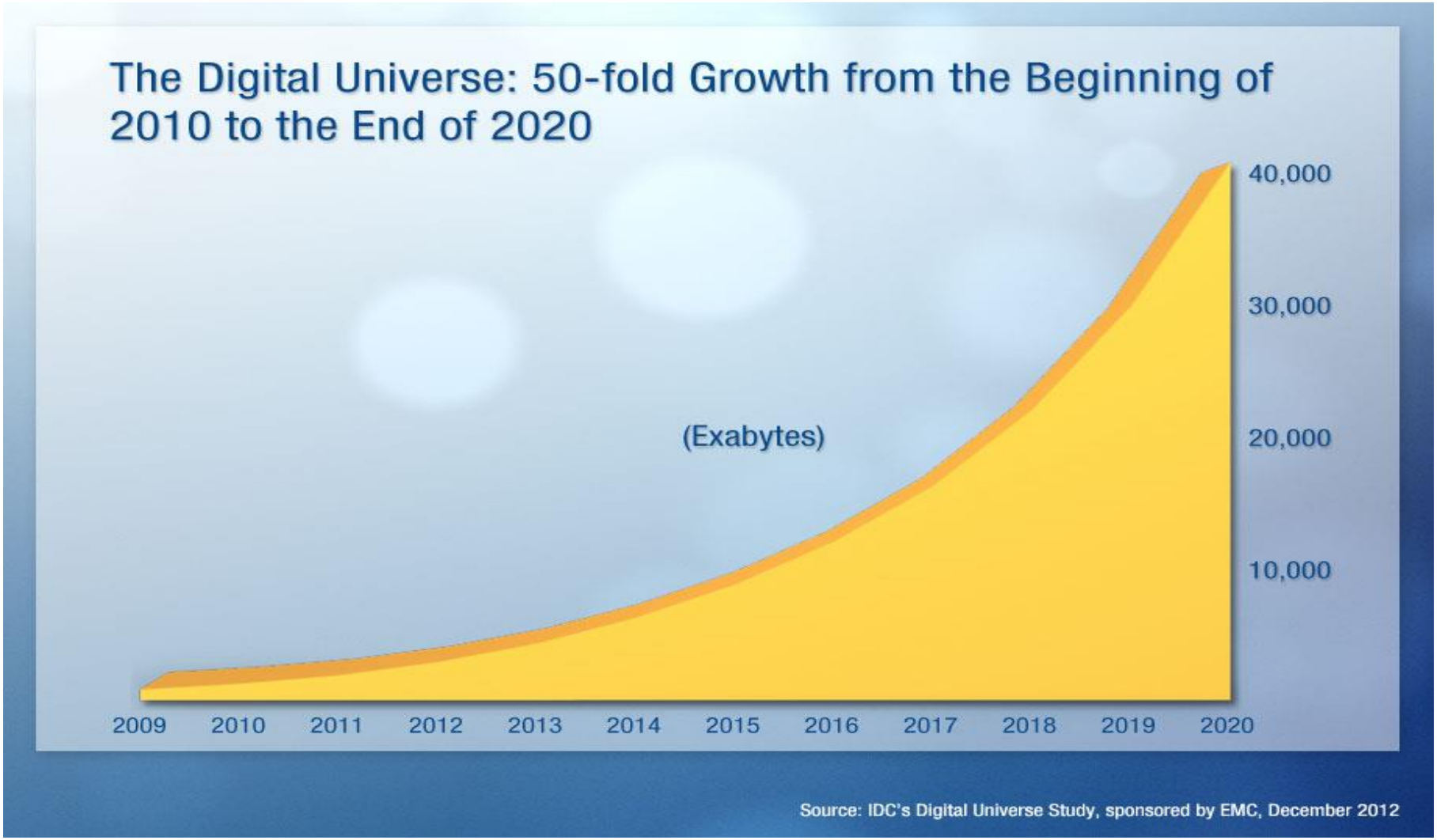
Distributed Era

HiveNet

Networked Computing

- Need for networked computing:
  - Better communication between systems
  - Resource sharing
- Systems and associated resources, such as printers, were interconnected to form local networks (LANs)
- These small networks evolved to form ARPANET, which finally evolved to the Internet

Information Explosion



Identifying Mersenne Primes

- **Mersenne Primes :** Prime numbers of the form 2^n-1
- Exponent *n* needs to be prime
- Used in Elliptic Curve Cryptography (ECC)
- Require enormous amount of computation

The GIMPS Project

- The Great Internet Mersenne Prime Search (**GIMPS**) was launched in 1996
- Allows users to contribute a portion of their unused CPU/GPU for computation
- By the end of 1996, the project could identify the 35th Mersenne Prime ($2^{1,398,269}$-1)
- In 2018, the project identified the 51st known Mersenne Prime ($2^{82,589,933}$-1)

SETI

- SETI – Search for Extra Terrestrial Intelligence
  - Analyzes radio waves in outer space for patterns
- Requires enormous amount of computation
- SETI@home allowed users to dedicate a part of their unused CPU for this purpose
- SETI@home and GIMPS were some of the first instances of volunteer computing projects

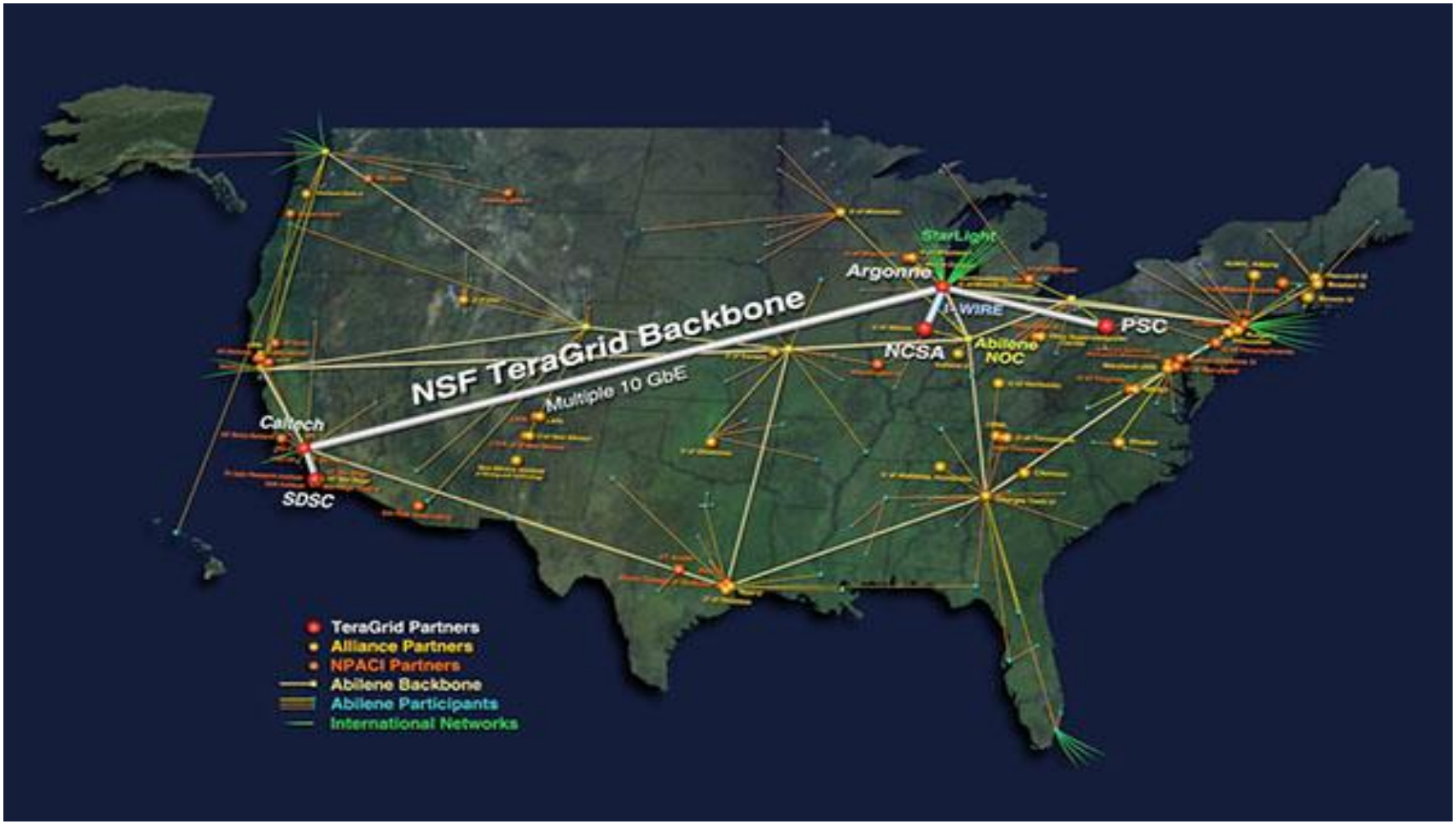
Grid Computing

- The volunteer computing projects are often compared to electric grids
  - Large number of connected nodes which act like a single entity
  - Users do not know which node serves their requests
- This comparison has led to the name “Grid Computing” being used to describe them

Grid Computing

- Multiple physically separated systems share data and resources for **performing a common task**
- Often, the original **task is split and distributed** among different systems or nodes
- Requires **special software** to be installed on the systems (“middleware”)

TERAGRID


StarLight
Argonne
I-WIRE
PSC
NCSA
Abilene NOC
NSF TeraGrid Backbone
Multiple 10 GbE
Caltech
SDSC
TeraGrid Partners
Alliance Partners
NPACI Partners
Abilene Backbone
Abilene Participants
International Networks

TERAGRID

- e-Science grid computing **spread over 11 sites** across USA
- **Petaflops of computing capability** and more than **30 petabytes of online and archival data storage** connected by **high speed optic fibre (10 Gbps)**
- Coordinated TeraGrid Software and Services (CTSS).
  - single-sign on
  - remote job submission
  - workflow support
  - distributed accounting and account management software
  - verification and validation software
  - set of compilers, programming tools etc

Advantages of Grid Computing

- Improved resource utilization
- General performance increase – parallel processing
- Easier collaboration
- Increased robustness

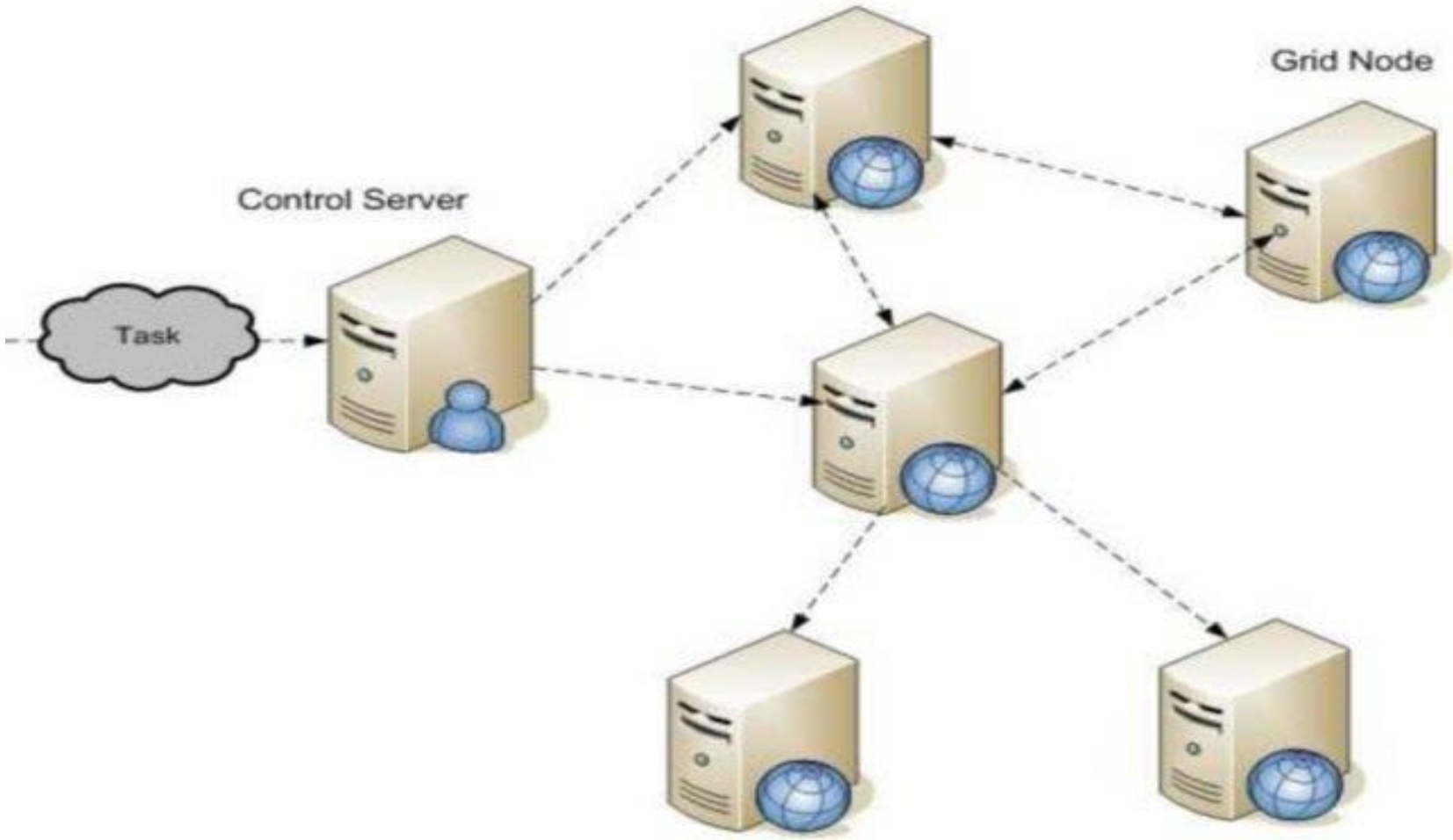
Grid and Utility Computing

A Definition for a Grid

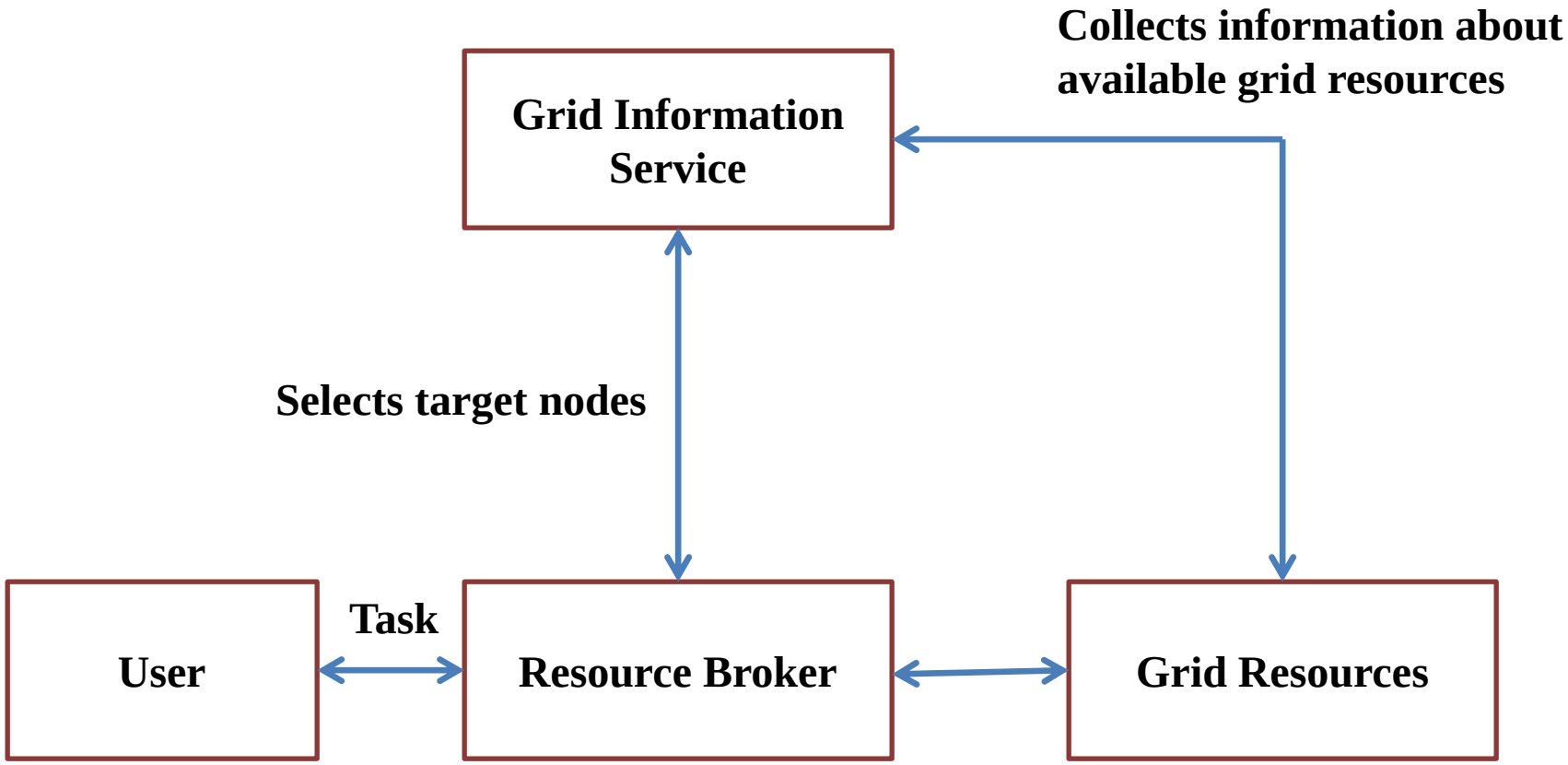
- A Grid is a system that :
  - coordinates resources that are not subject to centralized control
  - uses standard, open, general-purpose protocols and interfaces
  - delivers nontrivial qualities of service

[Ian Foster, What is the Grid? A Three Point Checklist]

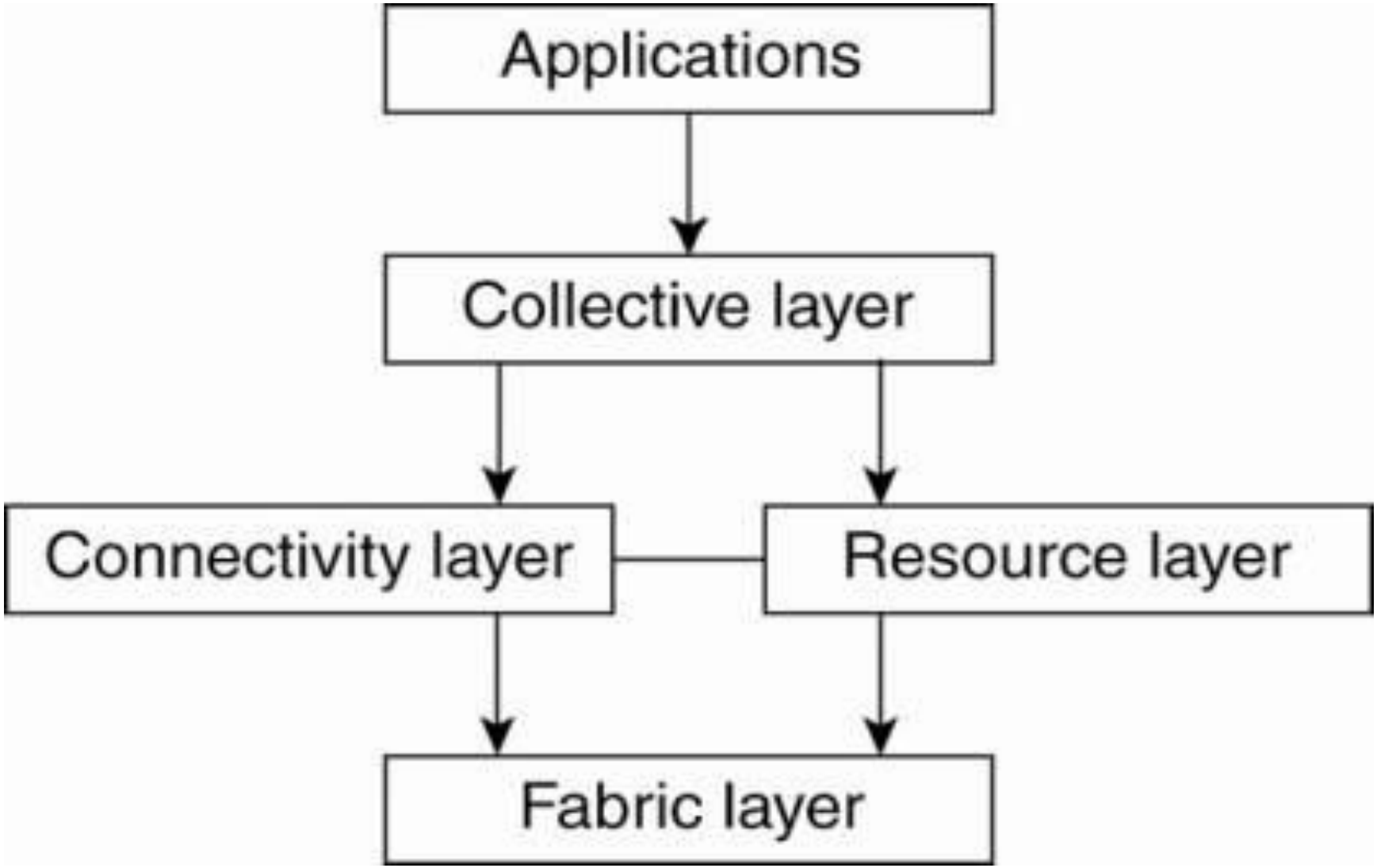
Grid Computing Architecture



How Grid Computing Works



How Grid Computing Works



Types of Grids

- **Computational Grids**
  - Network of high performance servers
- **Scavenging Grids**
  - Network of desktop computers
  - Use idle CPU cycles (“scavenging”) to execute tasks
- **Data Grids**
  - Access, modify or transfer huge amounts of data

Design Issues in Grids

- **Heterogeneity** – Connected systems may run different OS, have different architecture
- **Security** – How do we trust the solution returned by a node?
- **Dynamic Nature** – When nodes come and go dynamically, how do we obtain the result?
- **Network Connectivity** – What happens if nodes cannot periodically connect?
- **Is Grid Computing available for the masses?**

A Look Back at the "Grid" Analogy

- An electric grid possesses the following features:
  - Interconnected nodes acting like a single entity
  - Users do not know which node serves their requests
  - Users pay for what they use
- Grid computing is a collaborative effort, and users usually do not pay for the services they receive

Utility Computing

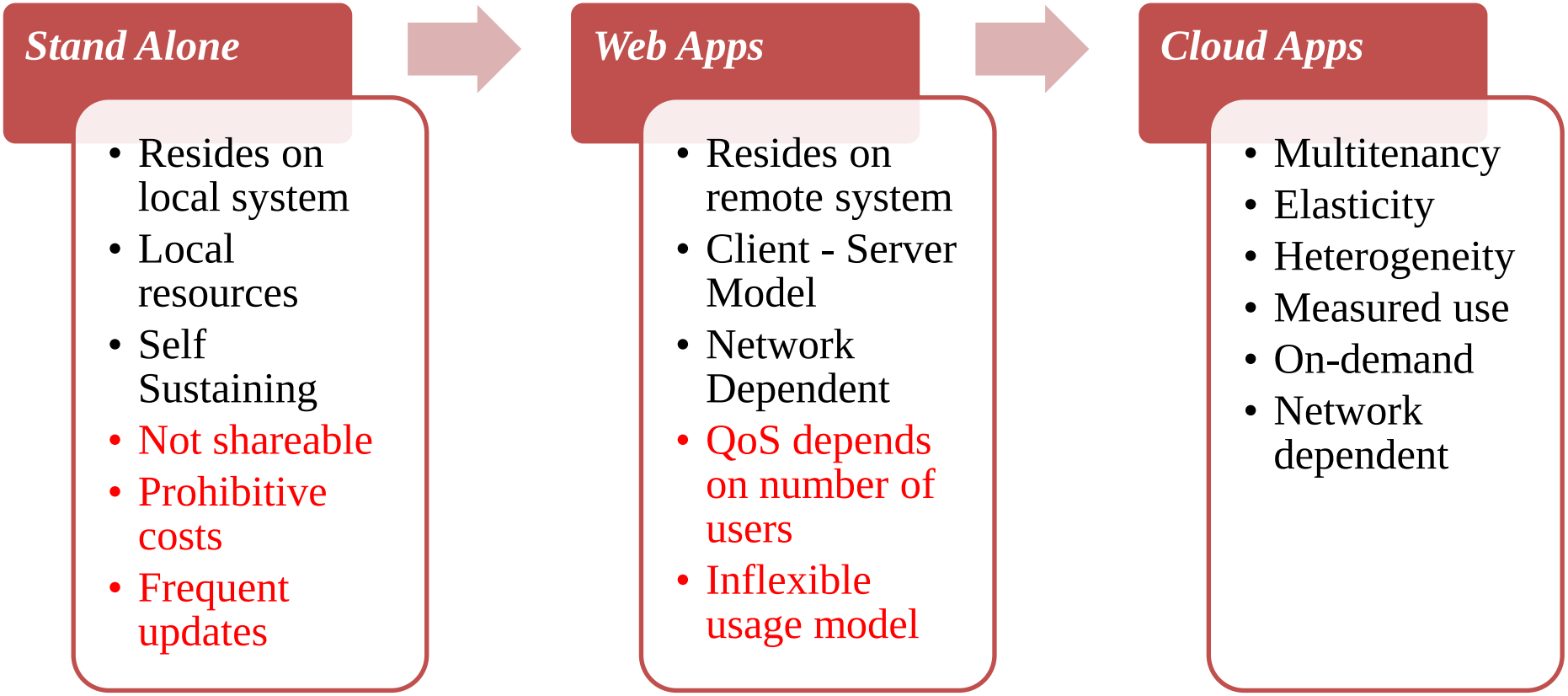
- Computing is considered as a utility, similar to how electricity is provided to homes
- Every connection is metered
- Users can utilize computing power according to their needs
- Pay-as-you-use model

A Brief Overview of Cloud, Edge and Fog Computing

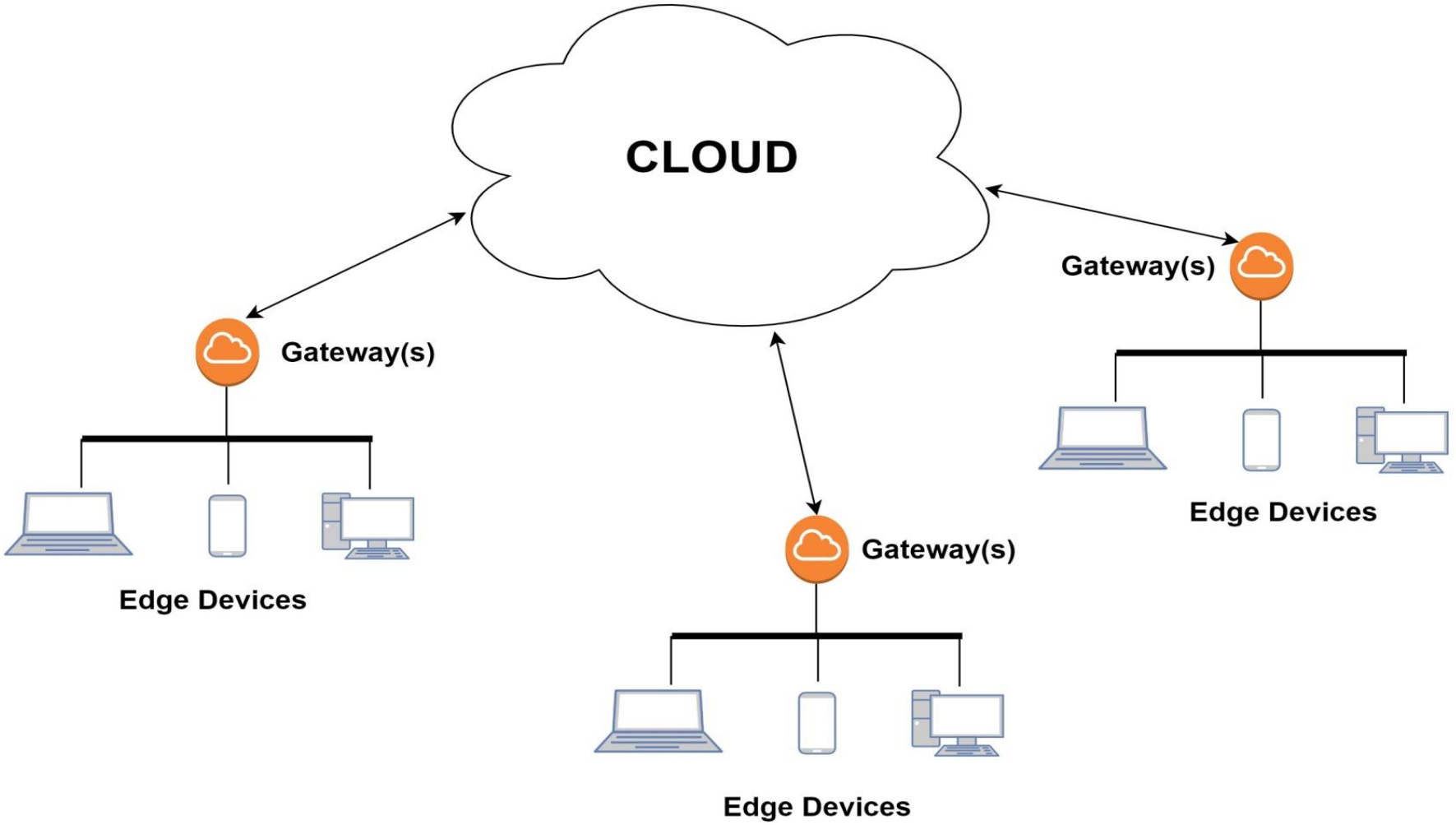
Cloud Computing

- Cloud Computing ≈ Grid Computing + Utility Computing
- Outsource utilities (storage, computation etc.) to a third party, called Cloud Service Provider
- Managed by the service provider
- Illusion of unlimited resources
- Pay-as-you-go model
- Accessible through a network

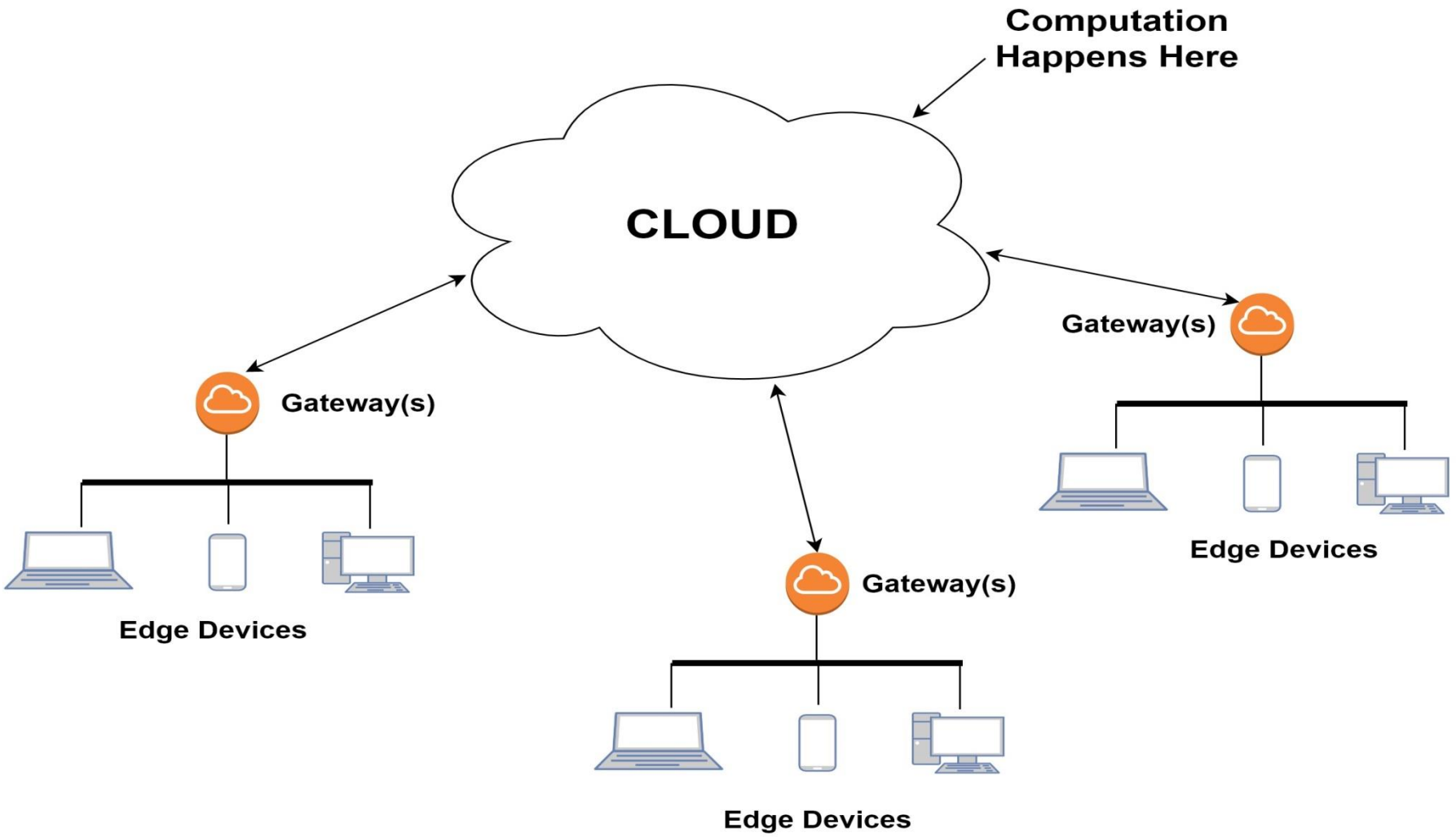
Evolution of Applications



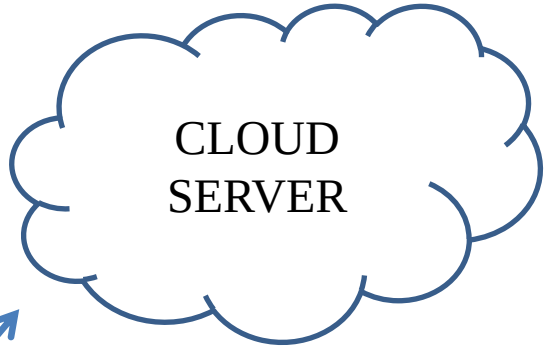
A Simple Architecture



Cloud Computing



Is Cloud Computing the Best Choice Always?

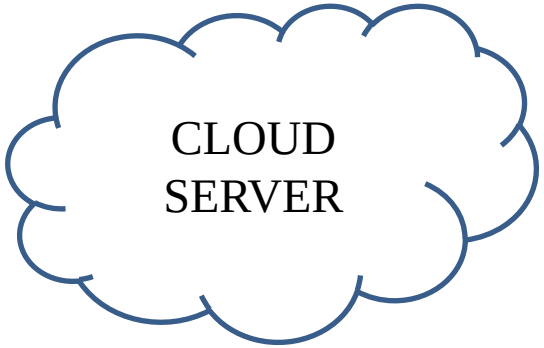


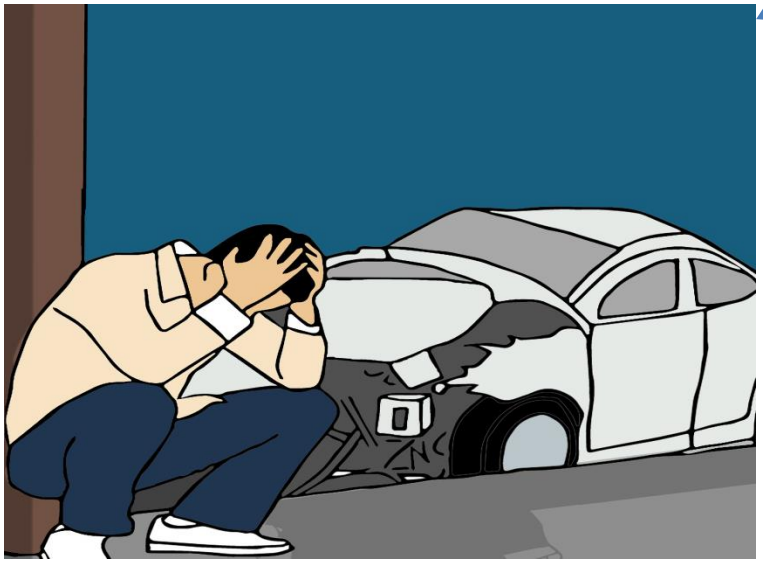
The car in front of me is braking! What should I do?

Consider the scenario of driverless cars

Is Cloud Computing the Best Choice Always?

BRAKE!!!
Sorry for the delay ☹



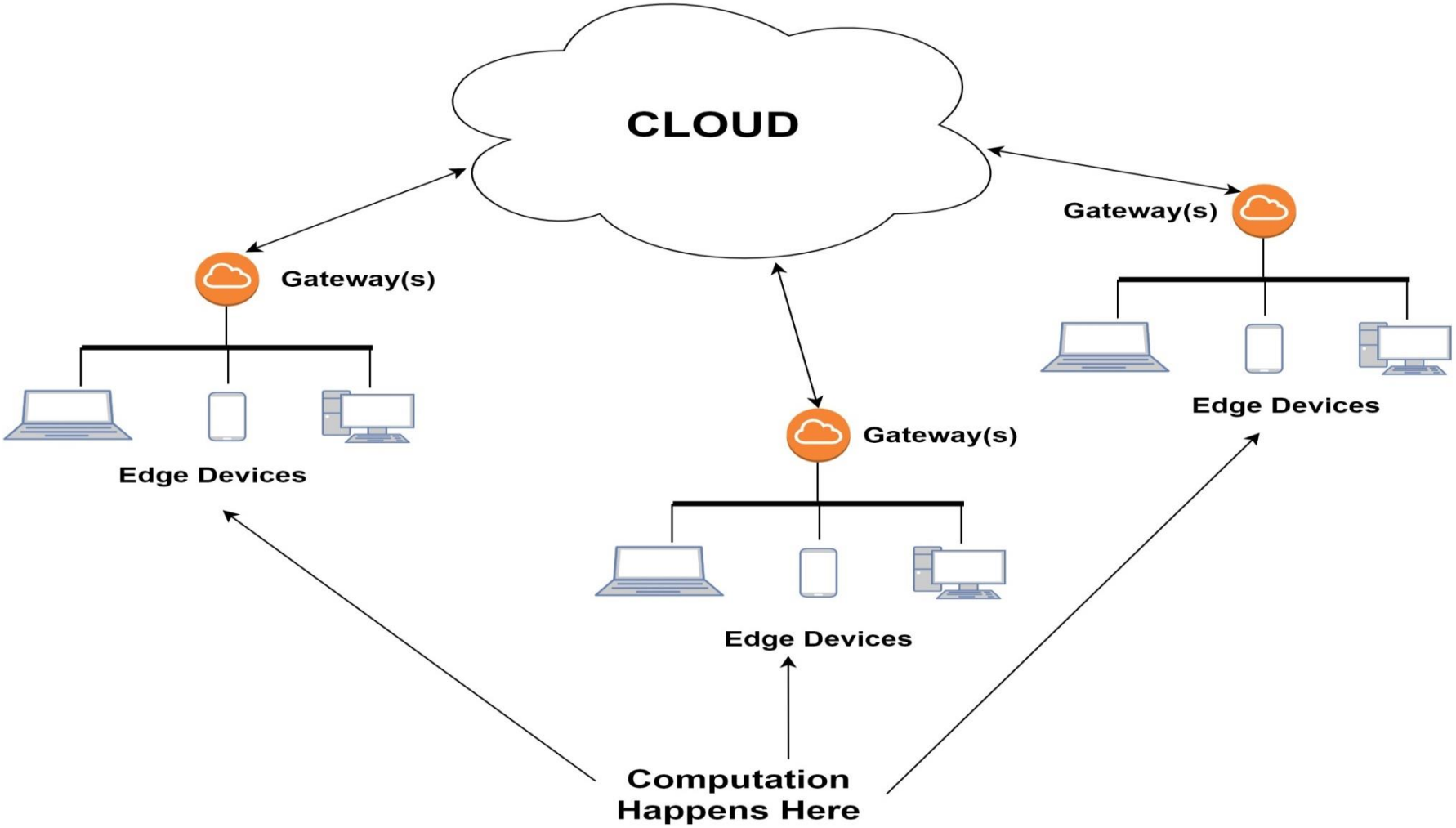


Consider the scenario of driverless cars

Problem: Self Driving Cars

- In real time systems, would there be enough time for data to be processed by a cloud server?
- In such cases, wouldn’t it be better if data was processed closer to the source?

Edge Computing



Features of Edge Computing

- **Advantages**
  - Low Latency
  - Faster Decisions
  - Privacy
- **Disadvantages**
  - Edge devices will have lower computational power
  - Lack of a global (or network level) view

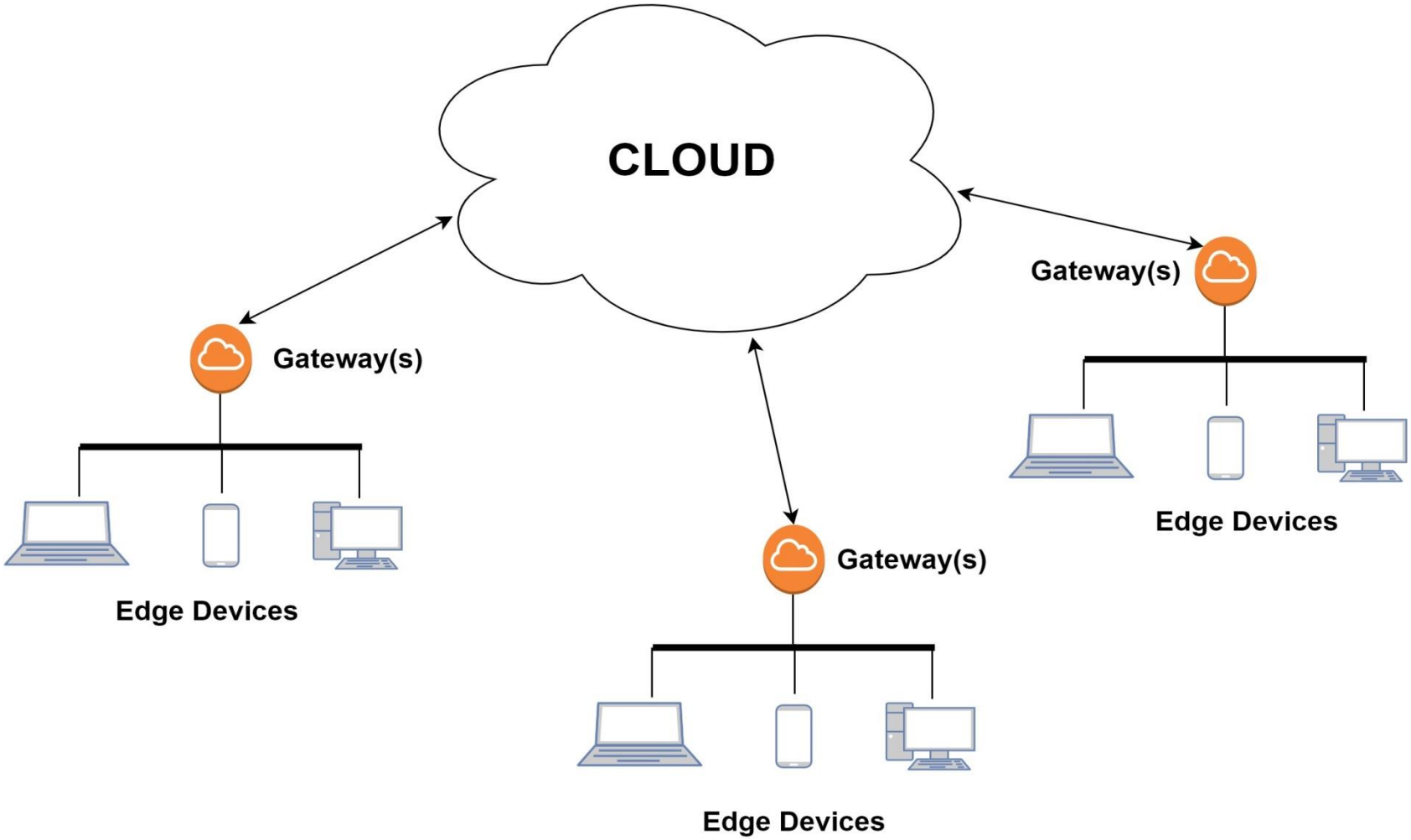
Automated Museum Tour Guides

More and more museums are now relying on automated robots to replace museum tour guides. For example, the Smithsonian Museum launched a robot named “Pepper” in 2018

An Issue with Edge Computing

- Consider the case of the automated museum tour guides:
  - All information that the robot needs is localized – need not use the “cloud” realistically
  - Interaction with tourists – need for local processing power
  - What if some exhibits are closed? Or the order changes?
  - There is a need for a broader, museum-level view for efficient functioning

Fog Computing



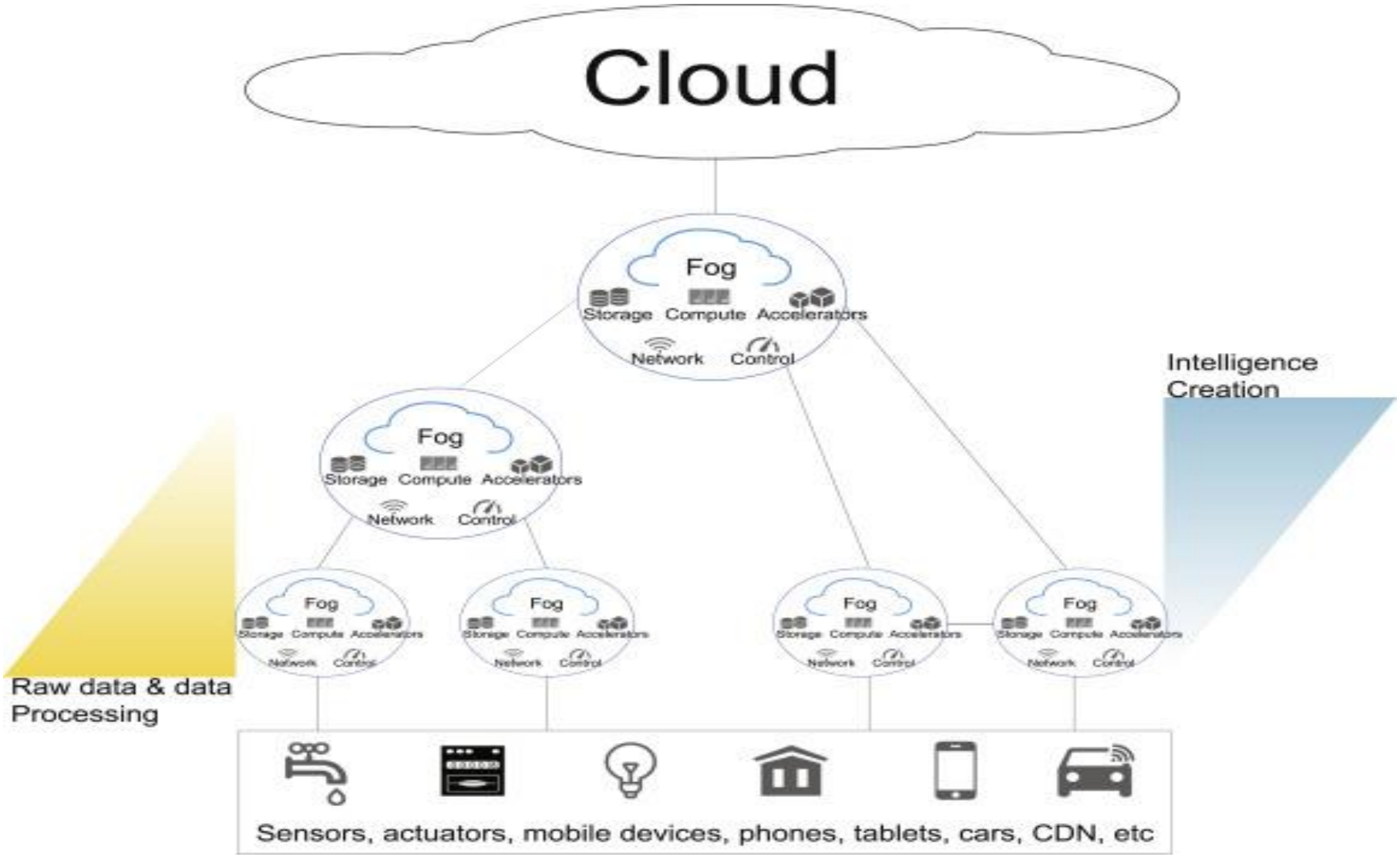
Fog Computing

- Move computation to a location between the nodes and the cloud
- Mobile users have predictable service demands subject to their locations
- Fog Servers can periodically connect with cloud servers and cache location specific information
- Information from edge devices can be processed by fog servers to provide an interactive experience

Features of Fog Computing

- Contextual location awareness + low latency
- Geographical distribution
- Heterogeneity
- Interoperability and federation
- Real-time interactions
- Scalability and agility of federated, fog-node clusters

Hierarchy of Edge, Fog and Cloud



A Note on Mist Computing

- **Mist computing** is a lightweight and rudimentary form of fog computing that resides directly within the network fabric at the edge of the network fabric
- Uses microcomputers and microcontrollers to feed into fog computing nodes and cloud computing services.

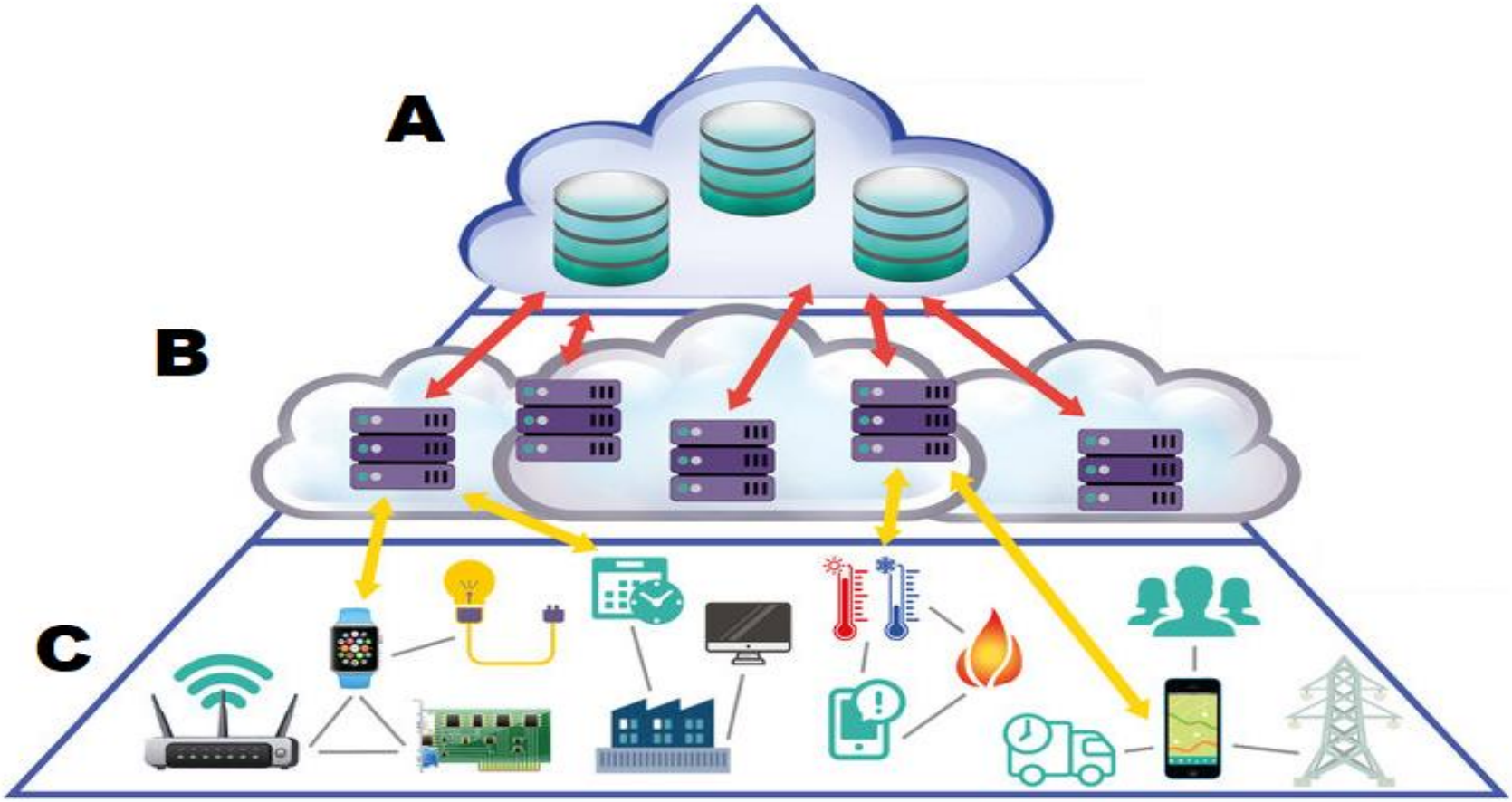
[Widely varying definitions of fog, mist and edge computing can be seen in literature. The definitions used here are adopted from NIST specifications]

A Note on Dew Computing

- What happens to cloud computing when you lose network access?
- Dew Computing is a framework wherein a version of the cloud architecture is replicated on a local system – immune to loss of connectivity
- Applicable to systems with direct human interaction, such as laptops, desktops and mobile devices
- Eg: Dropbox

Basics of Cloud Computing

Review



1. Mist Computing 2. Fog Computing 3. Cloud Computing
4. Edge Computing 5. Dew Computing

A Simple Example

- ABC Corporation is an e-commerce company
- They have traditionally hosted their website on an on-premise server
- Recently, they have been getting a lot of business
- Their IT head, Bob, observes that their server sometimes operates very near its capacity
- What options does Bob have?

Two Options

Horizontal Scaling

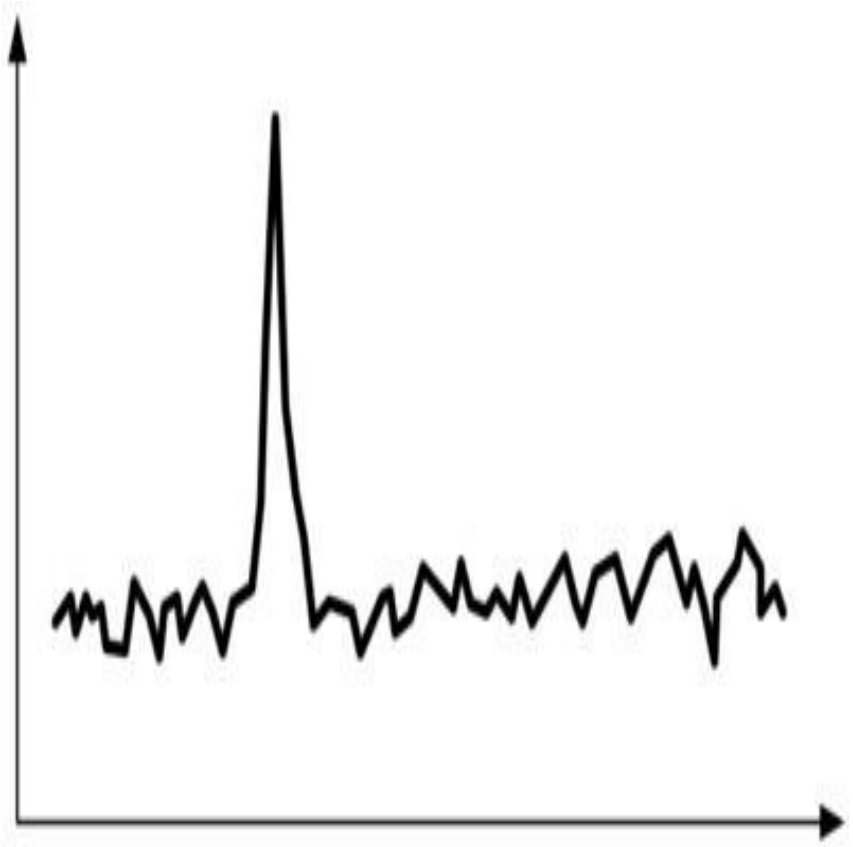
- Add more systems to handle the extra load
- Usually less expensive in the long run
- No downtime
- Potentially unlimited scaling

Vertical Scaling

- Upgrade the existing system
- Usually expensive in the long run
- Possible downtime
- There is a limit to which scaling can be done

But what if…

- But what if the website only shows a spike in usage during holiday season?
- If you upgrade, your new hardware remains unused most of the year
- If you don’t upgrade, the company loses out on important revenue



Secret Option #3

- Bob decides to move their business to the cloud
  - Automatic scaling when usage increases
  - When usage decreases, the server resources are scaled back
  - No upfront capital cost
  - Pay only for the resources that are used

A Formal Definition of Cloud Computing

“ 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”

A Formal Definition of Cloud Computing

“ 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”

A Formal Definition of Cloud Computing

“ 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”

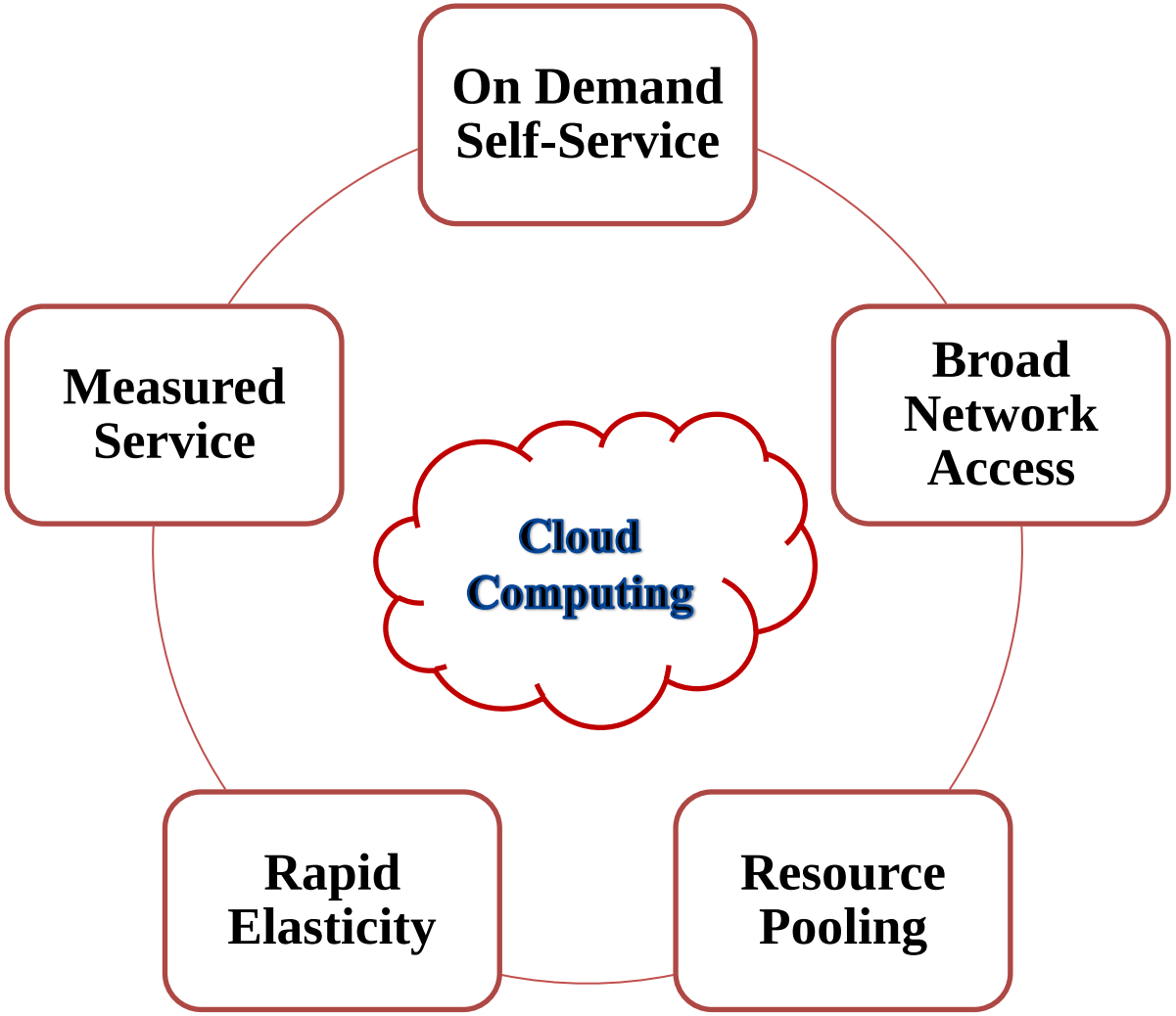
A Formal Definition of Cloud Computing

“ 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”

A Formal Definition of Cloud Computing

“ 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”

Features of Cloud Computing



Features of Cloud Computing

1. **On Demand Self – Service**
   - Resources can be provisioned and released as an when needed
   - No need for human interaction
2. **Broad Network Access**
   - Resources can be accessed over the network
   - Any device – smartphone, tablet, laptop, thin client, PC etc.

Features of Cloud Computing

3. **Resource Pooling**
   - Service provider has a pool of resources that can be allocated and deallocated to customers
4. **Rapid Elasticity**
   - Resources can be allocated on the fly
   - Customers get an illusion of unlimited resources
5. **Measured Service**
   - Pay per use

SLA – The Warranty for Cloud Services

- SLA (Service Level Agreement) is a bond for performance negotiated between the cloud services provider and the client
  - Availability and Performance
  - Security / privacy of the data
  - Disaster Recovery expectations
  - Location of the data
  - Access and portability to the data
  - Process to identify problems and resolution expectations
  - Dispute mediation process (e.g. escalation process, consequences)
  - Exit Strategy with expectations on the provider to ensure smooth transition

Cloud Architecture

User / Client Layer

Network Layer

Cloud Management Layer

Hardware Resource Layer

Anatomy of the Cloud

Application

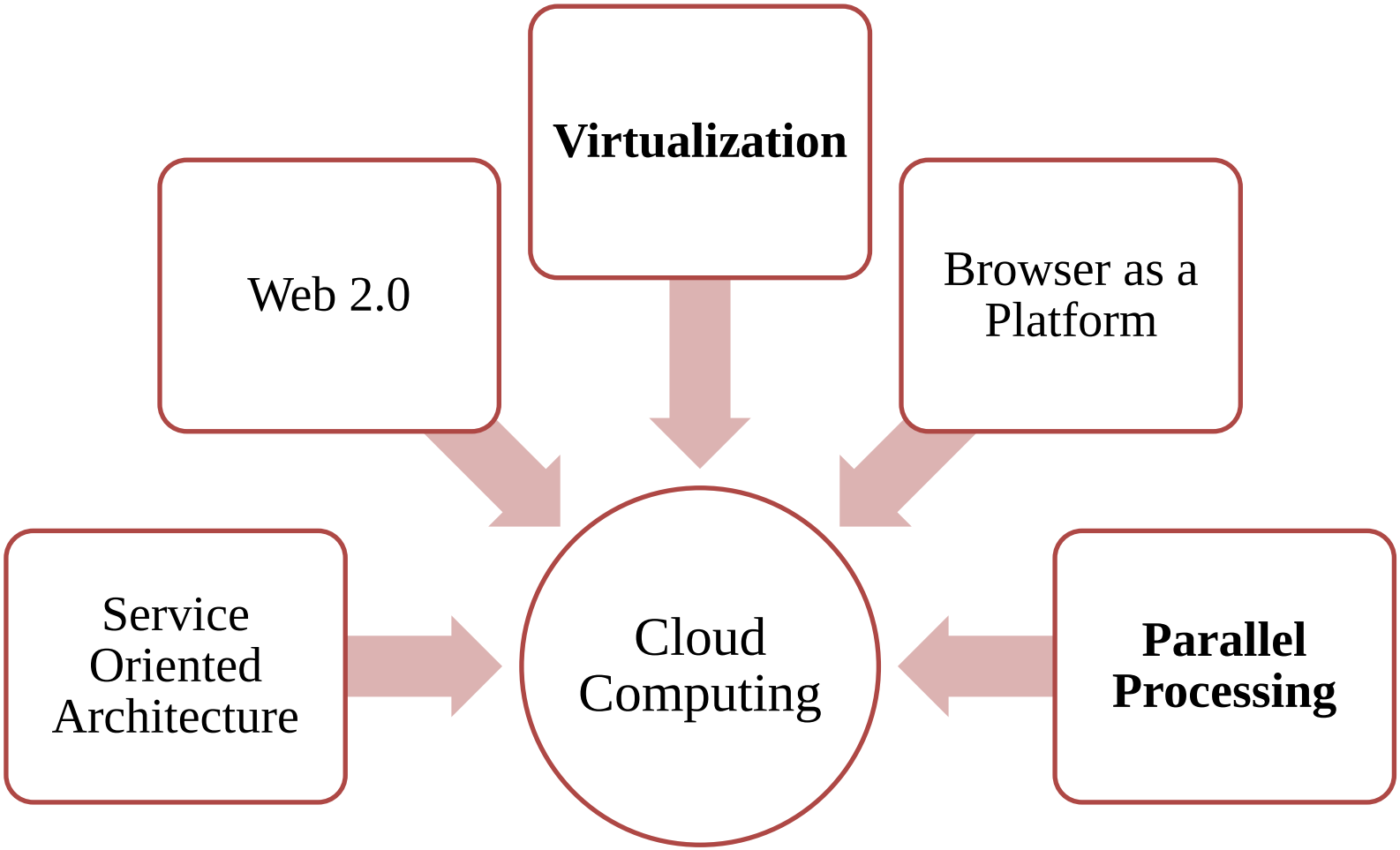
Platform

Virtualized Infrastructure

Virtualization

Server / Storage / Datacentres

Enabling Technologies for Cloud Computing



Web 2.0

Dr. Amit Praseed

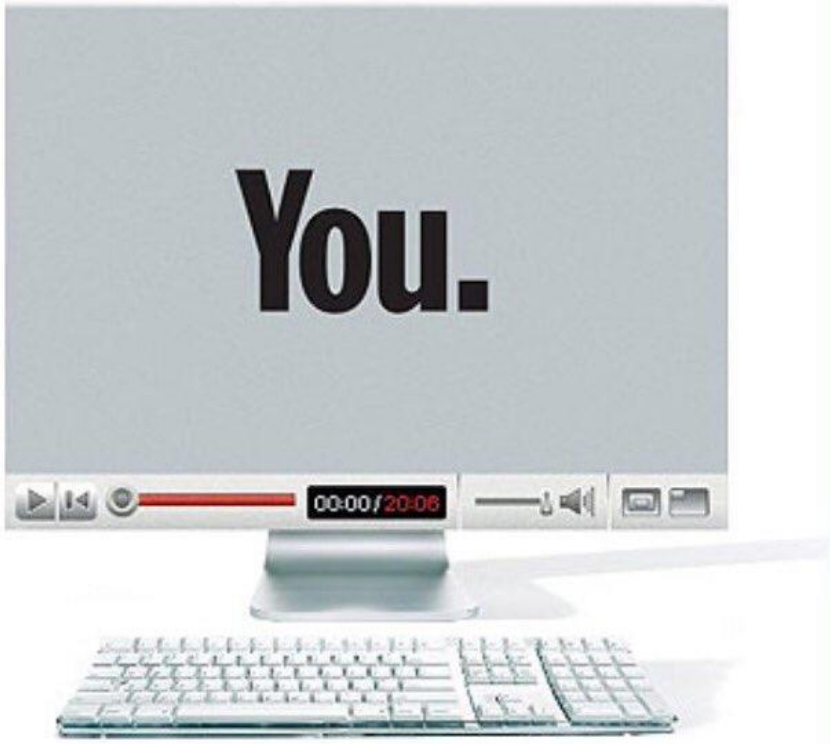
Who was Time Person of the Year 2006?

DECEMBER 25, 2006 / JANUARY 1, 2007

www.time.com

TIME

PERSON OF THE YEAR

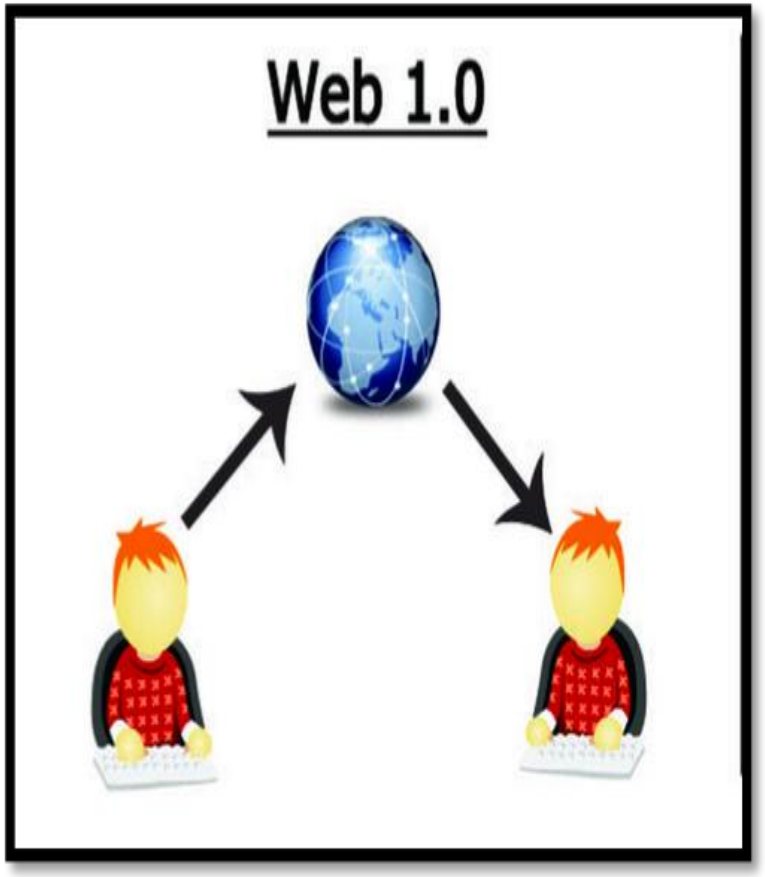


Yes, you.
You control the Information Age.
Welcome to your world.

What do you do on the Web?

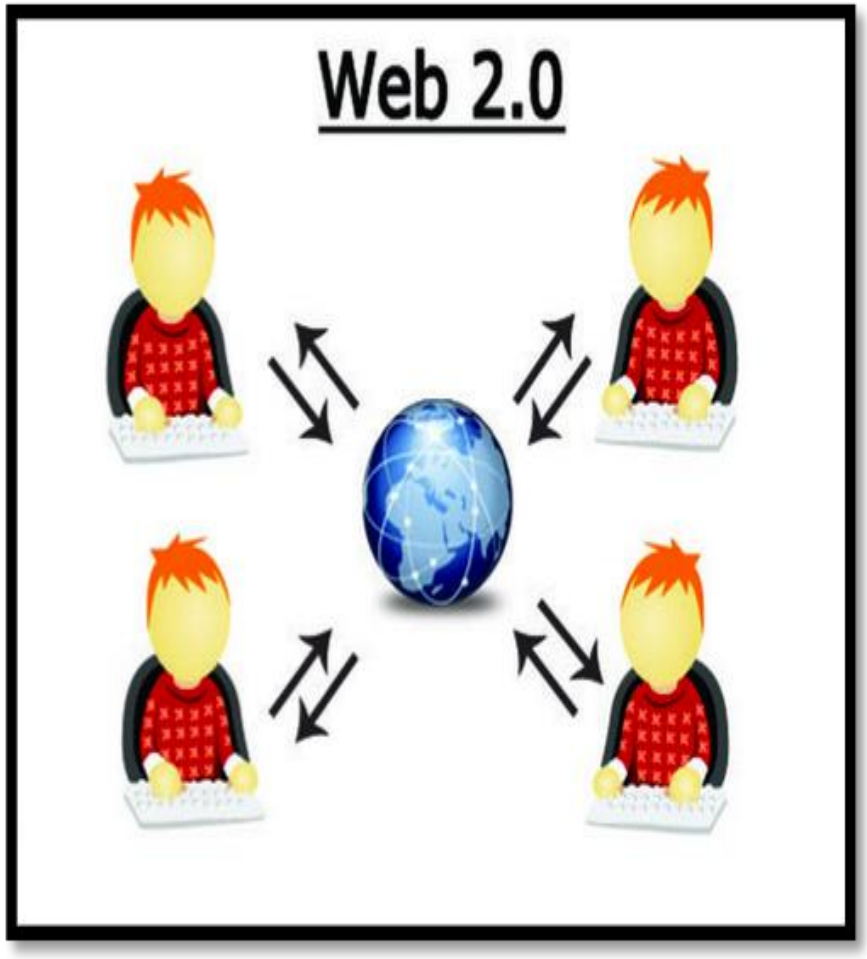
It wasn’t always like this…

- “Read only” web
- Few users in general
- Static websites
- Few publishers of content
- Majority accessed the content
- Slow network speeds and lower Internet penetration



Enter Web 2.0

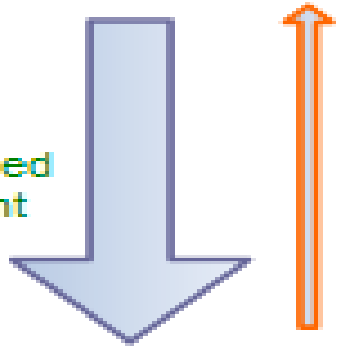
- “Read - write” web
- Huge increase in user base due to Internet penetration and higher network speeds
- Any user can publish content
- Focus is on user participation

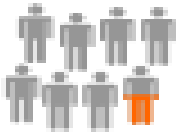


Web 1.0

"the mostly read-only Web"

250,000 sites



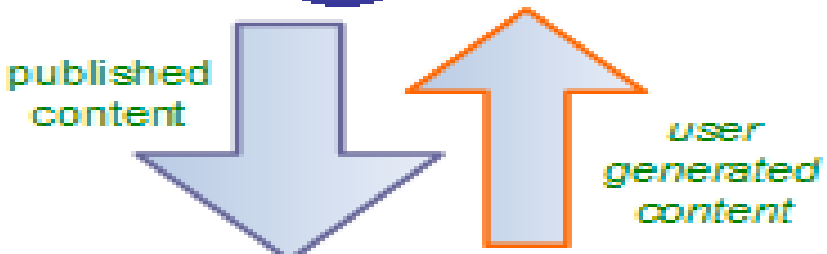


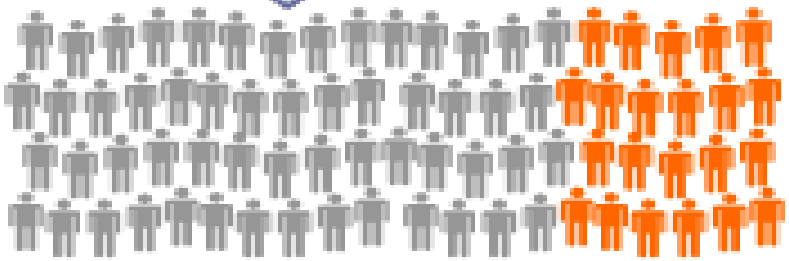
45 million global users

Web 2.0

"the wildly read-write Web"

80,000,000 sites





1 billion+ global users

Harnessing Collective Intelligence

- Hyperlinking is the foundation of the web.
  - Google's breakthrough in search was PageRank
  - eBay's product is the collective activity of all its users
  - Amazon has made a science of user engagement.
    - They have an order of magnitude more user reviews, invitations to participate in varied ways on virtually every page
    - They use user activity to produce better search results.

Harnessing Collective Intelligence

- Wikipedia
  - "with enough eyeballs, all bugs are shallow"
- Sites like del.icio.us and Flickr
  - "folksonomy" : A style of collaborative categorization of sites using freely chosen keywords, often referred to as tags.
- Collaborative spam filtering products
  - aggregate the individual decisions of email users about what is and is not spam
  - outperforming systems that rely on analysis of the messages themselves.
- Blogs, Comments and RSS

Data is Paramount

- The race is on to own certain classes of core data
  - location, identity, calendaring of public events, product identifiers and namespaces
- Mashups : Collecting and aggregating data from multiple sources

End of the Software Release Cycle

- Operations must become a core competency
- Users must be treated as co-developers
- Use of lightweight programming models
- Software above the level of a single device
- Rich User Experience
  - AJAX, Javascript

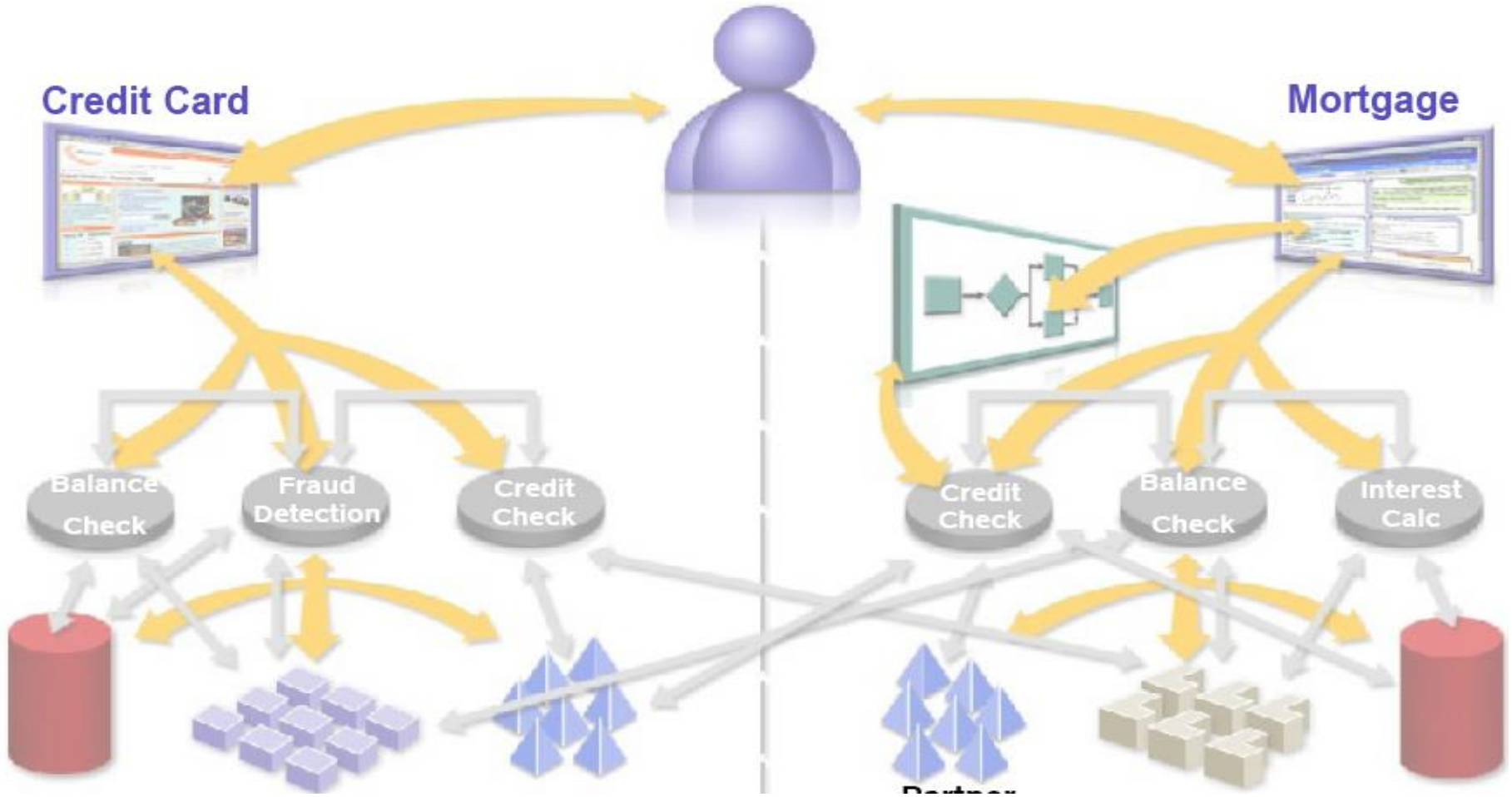
Web 3.0?

- Centralized services → Decentralized services
- Machine readable information
- “Semantic Web”

Service Oriented Architecture

Dr. Amit Praseed

A Simple Bank Scenario



Customer Data

Backend System

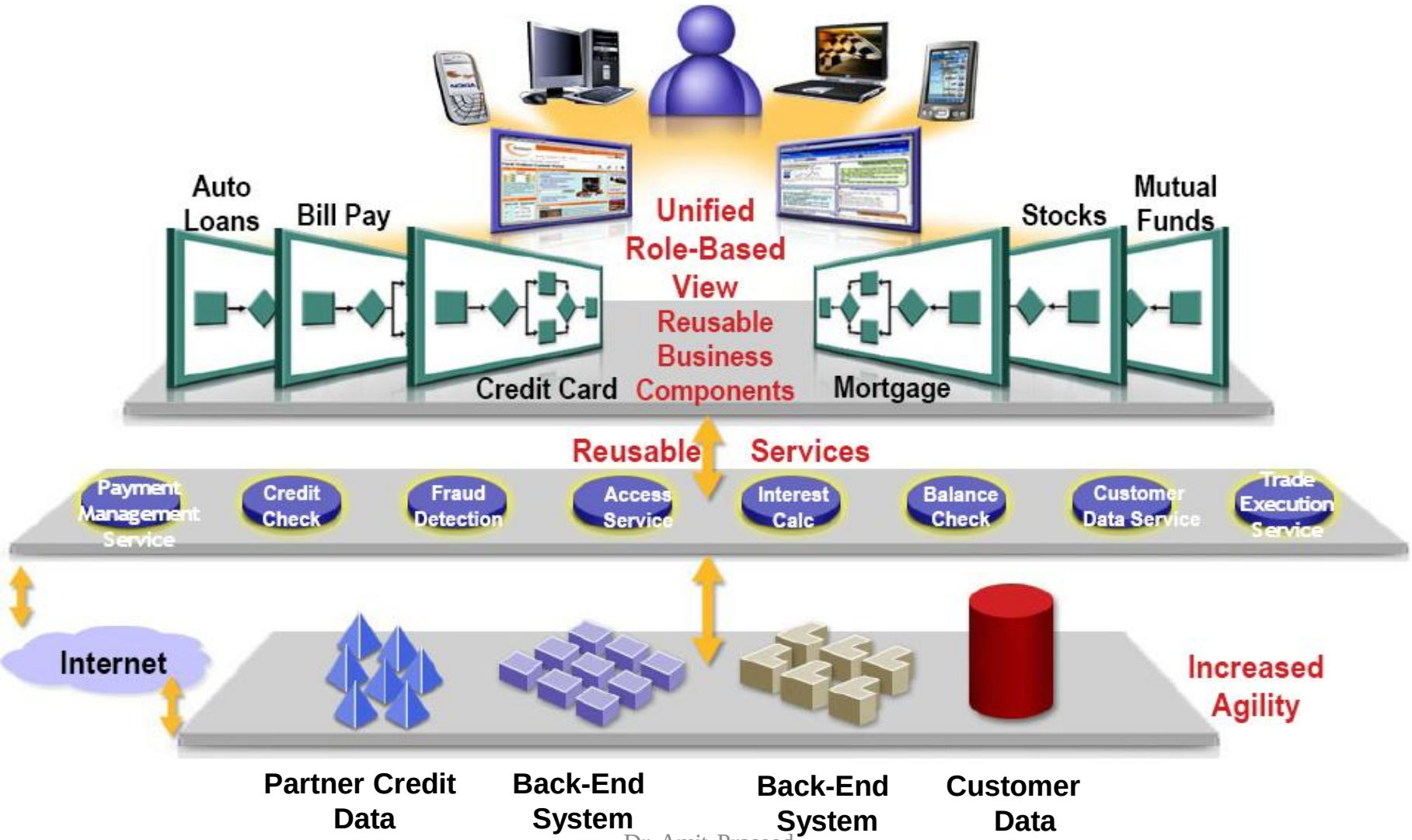
Partner Credit Data

Partner Credit Data

Backend System

Customer Data

A Better Architecture

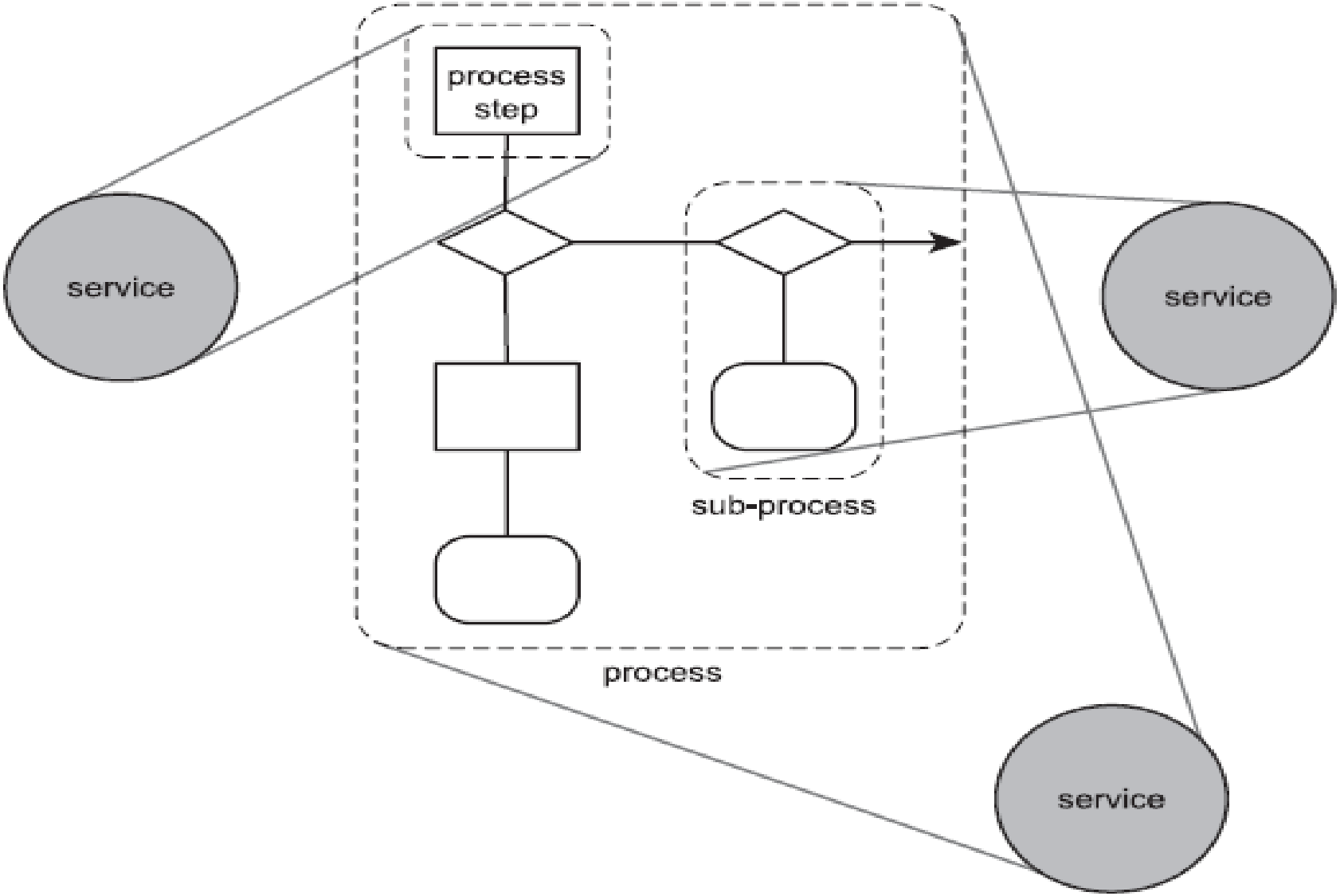


Service Oriented Architecture (SOA)

- Service-Oriented Architecture (SOA) is an architectural style.
- Applications built using an SOA style deliver functionality as **services that can be used or reused** when building applications or integrating within the enterprise or trading partners.

What constitutes a Service?

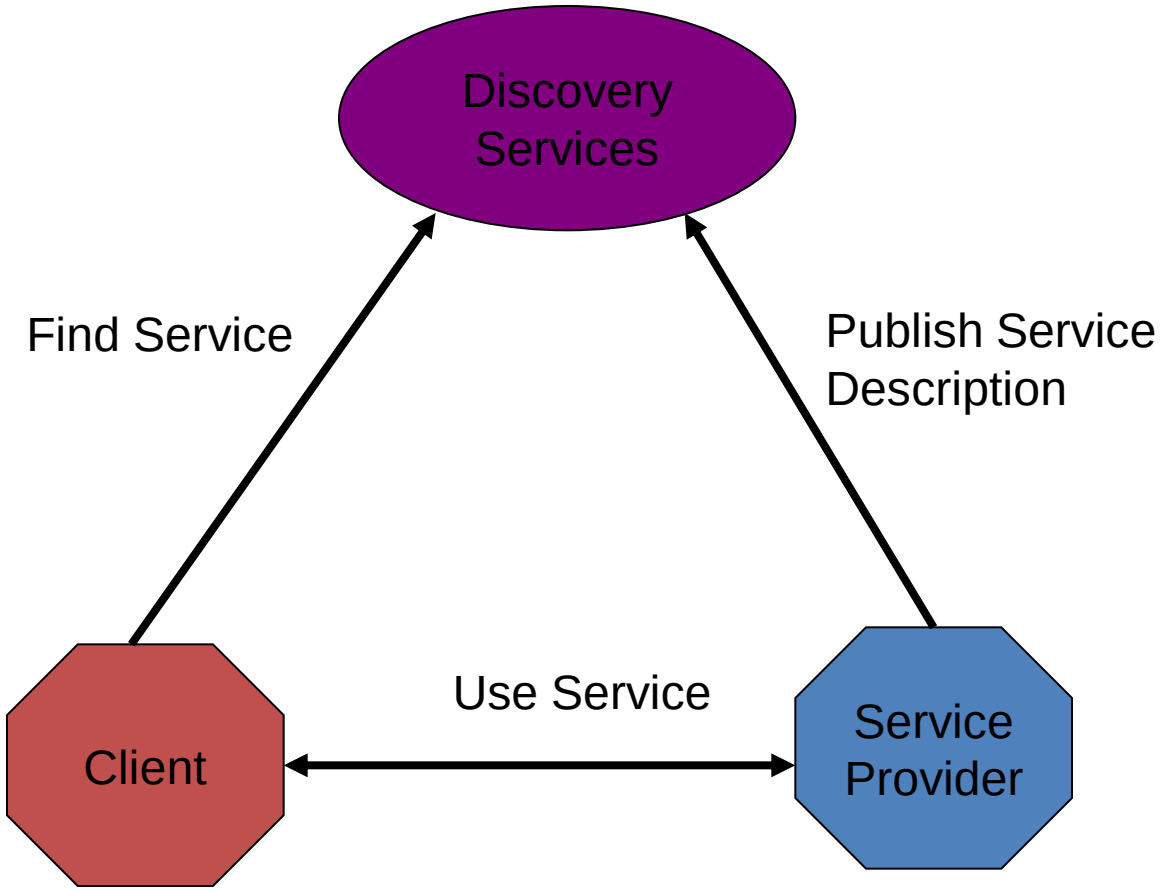
- A logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit, provide weather data, consolidate drilling reports).
- Self-contained.
- *May be* composed of other services
- Is a “black box” to consumers of the service.


process
step
service
service
sub-process
process
service

SOA Features

- *Loose coupling*
- *Service contract*
- *Autonomy*
- *Abstraction*
- *Reusability*
- *Statelessness*
- *Discoverability*

SOA Framework



Web Services

- A web service is “a software system designed to support interoperable machine-to-machine interaction over a network”
- It differs from a web application in one key point
  - Web applications enable communication between human clients and machines
  - Web services enable communication between clients

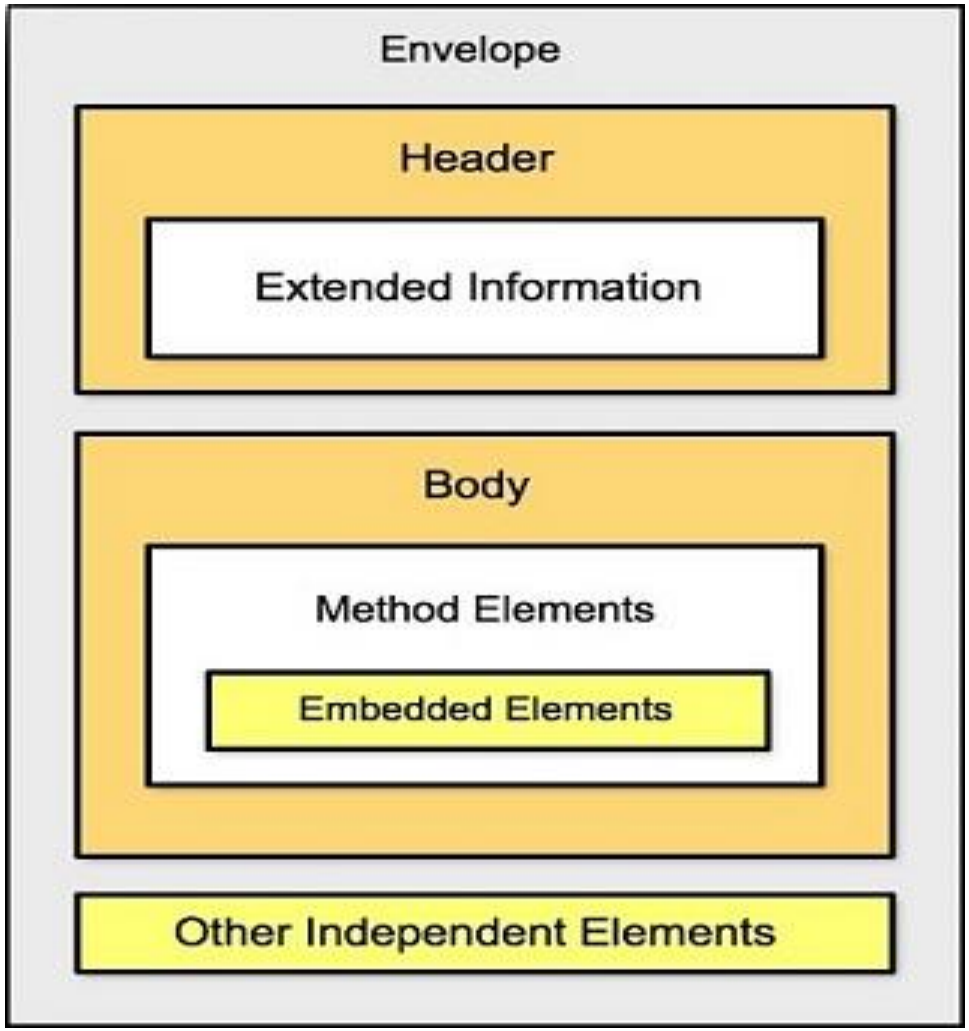
Composition of Web Services

- SOAP – Simple Object Access Protocol
  - XML based protocol used for communication
- WSDL - Web Services Description Language
  - describes the interface, a set of operations supported by a web service in a standard format.
- UDDI - Universal Description, Discovery, and Integration
  - provides a global registry for advertising and discovery of web services

SOAP Message Example

```
<?xml version="1.0"?>
<SOAP:Envelope
    xmlns:xsi="http://www.w3.org/1999/XMLSchema/instance"
    xmlns:xsd="http://www.w3.org/1999/XMLSchema/instance"
    xmlns:SOAP="urn:schemas-xmlsoap-org:soap.v1">
  <SOAP:Body>
      <calculateArea>
        <origin>
           <x xsd:type="float">10</x>
           <y xsd:type="float">20</y>
        </origin>
        <corner>
           <x xsd:type="float">100</x>
           <y xsd:type="float">200</y>
        </corner>
       </calculateArea>
   </SOAP:Body>
</SOAP:Envelope>
```

SOAP Message Structure



SOAP Protocol Extensions

- WS-* is used to refer to extensions to the basic Web services framework established by first-generation standards
  - **WS-Security**: Handles encryption and digital signatures
  - **WS-Policy**: Expands on WS-Security, enabling you to more specifically detail how and by whom a service can be used.
  - **WS-I**: Provides a set of standards and practices to prevent interoperability issues, as well as standardized tests to check for problems.
  - **WS-BPEL**: Provides a way to specify interactions between multiple web services, such as branching and concurrent processing

Issues with SOAP

- SOAP was meant to facilitate easier communication between machines using a common platform of XML
  - Difficult to debug
  - Difficult to build
  - Difficult to use
- Salesforce had built an “Internet as a Service” package for selling APIs, but their use of complex XML proved to be a major deterrent.

REST

- A simpler way to implement SOA is using REpresentational State Transfer (REST)
  - Resource Identification through URI
  - Uniform, Constrained Interface
  - Self-Descriptive Message
  - Stateless Interactions
- The use of REST made API calls effortless, leading to companies like EBay, Amazon and Flickr cashing in

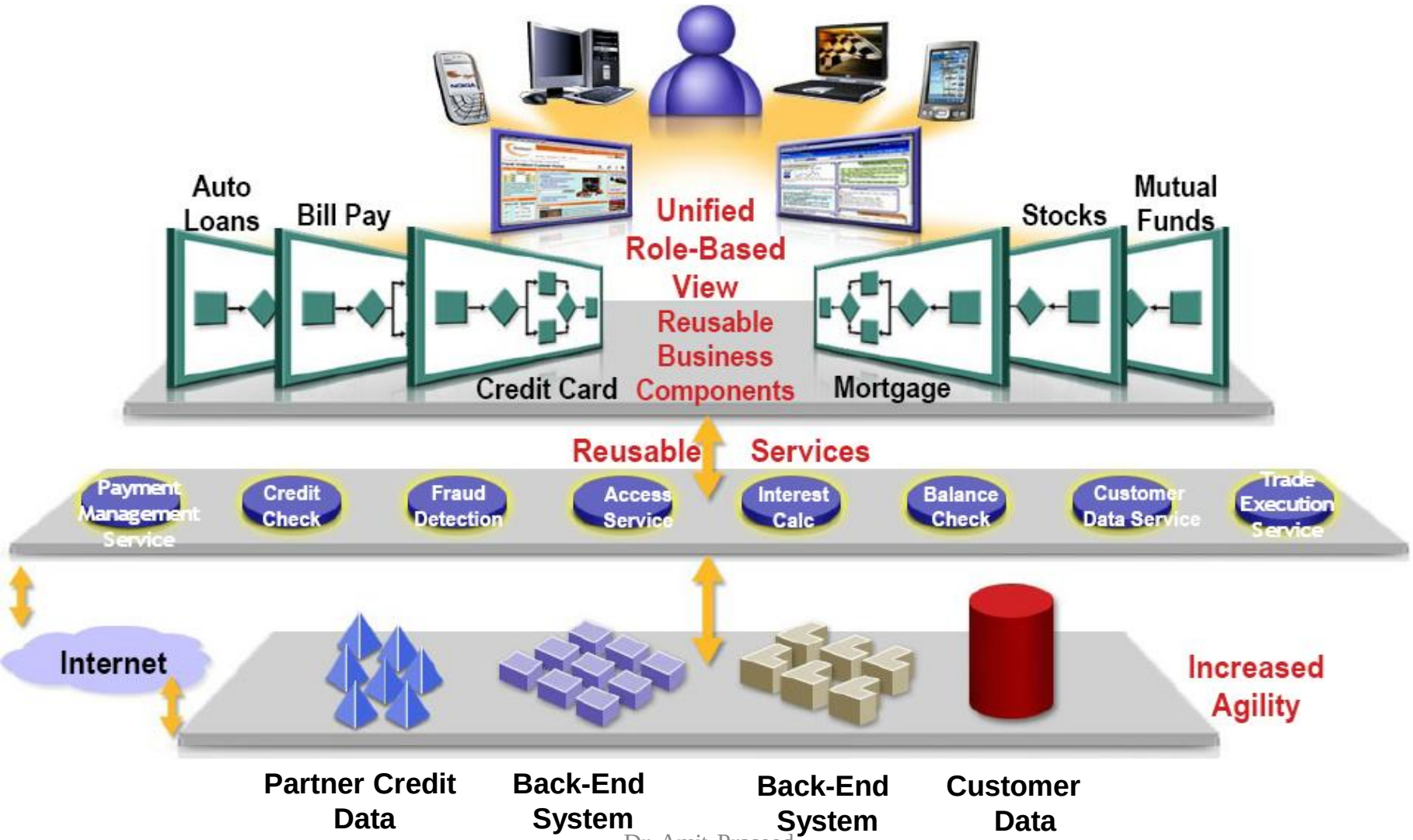
A Comparison

Feature	SOAP	REST
Nature	Protocol	Architectural Style
State	Stateful/Stateless	Stateless
Format	XML	XML, JSON, plaintext etc.
Transfer Protocol	HTTP, HTTPS, TCP, FTP, SMTP etc.	HTTP, HTTPS
Security	WS-Security, ACID, HTTPS, SSL	HTTPS, SSL
Speed	Slow	Fast
Learning Curve	Difficult	Easy
Community	Small	Large

So, is SOAP dead?

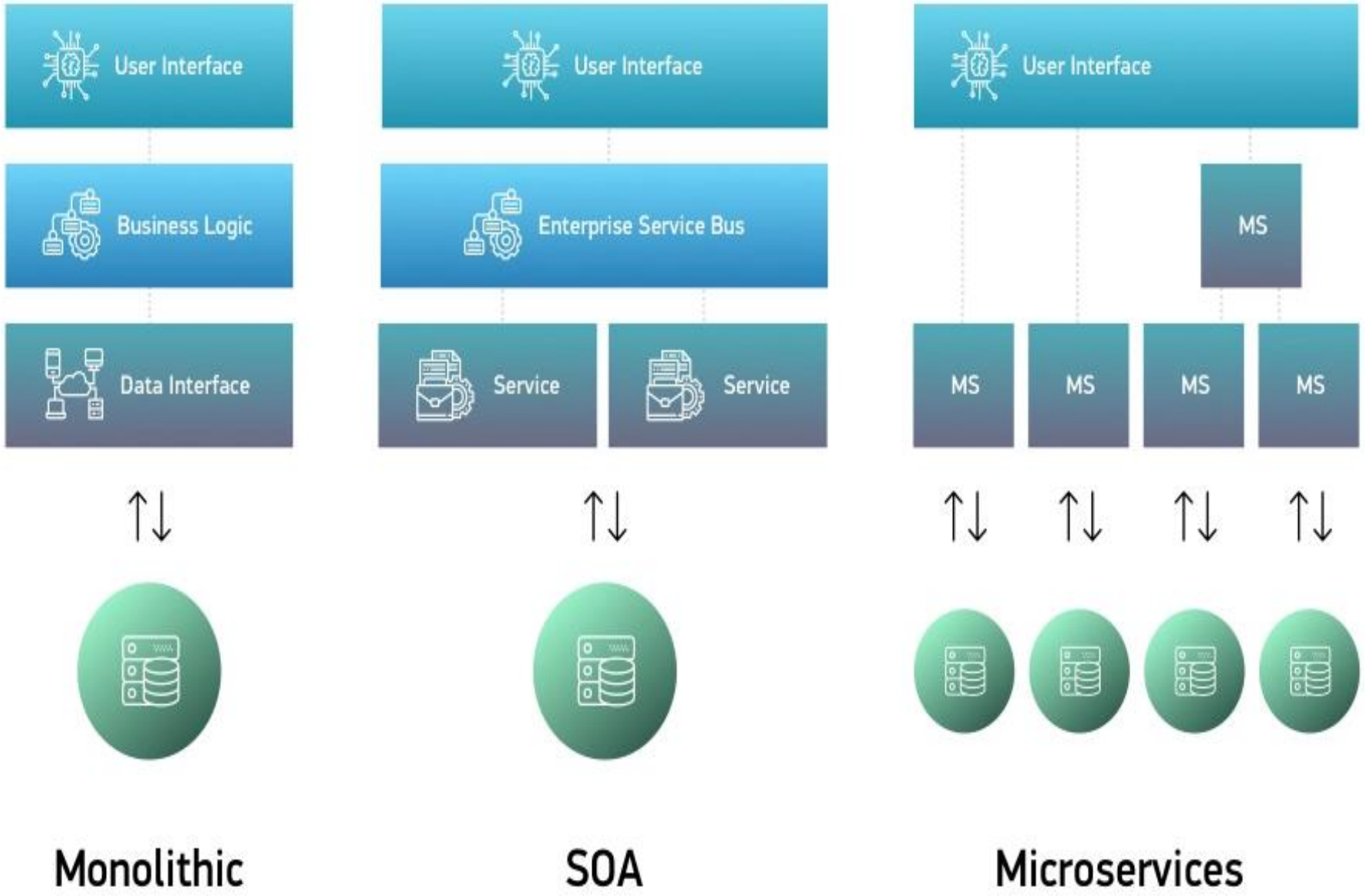
- REST is a simple and efficient way to build and use a Service Oriented Architecture
- Currently, nearly 83% of APIs use REST
- However, SOAP does have its benefits
  - Inbuilt successful/retry mechanism
  - Inbuilt security features
  - Extensible
  - Customizable

A Look Back at the SOA Architecture



Microservices : The Next Evolution

- Microservice Architecture involves the splitting of large software applications into decoupled modules, running unique processes and communicating through APIs
- Every microservice is an independent software component by itself, so they do not have to be modified frequently
- Almost all popular organizations like EBay, Amazon and Netflix employ microservices


User Interface
Business Logic
Data Interface
User Interface
Enterprise Service Bus
Service
Service
User Interface
MS
MS
MS
MS
MS
Monolithic
SOA
Microservices

Service Registration and Discovery

- Registration
  - Self Registration
  - Third Party Registration
- Discovery
  - Client Side Discovery
  - Server Side Discovery

Cloud Service Models

Dr. Amit Praseed

Can you name some cloud services which you use?

Cloud Services

- There are three types of services offered by cloud service providers
  - Software as a Service (SaaS)
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)

Software as a Service

- Provides access to applications running on a cloud infrastructure, including network, servers, OS, storage or individual applications
- Accessible from client devices via a browser or program interface
- Customer does not manage or control underlying infrastructure or applications – limited customization
- Eg: Google Documents, Sharepoint etc.

SaaS Features

- One to Many
- Web Access
- Centralized Management
- Multi-device support
- Scalability
- Availability
- API integration

When to use SaaS?

- You should use SaaS for:
  - On demand software
  - Software for startups
  - Software compatibility
  - Varying loads
- SaaS might not be a good option if:
  - Unreliable network connection
  - Confidential data

SaaS - Pros and Cons

Pros

- No installation
- Cost savings
- Less maintenance
- Ease of Access
- Dynamic scalability
- Disaster Recovery
- Multitenancy

Cons

- Security
- Connectivity
- Loss of Control

Platform as a Service

- Provides ability to create and/or deploy applications on cloud infrastructure
  - Languages, libraries, services and tools are provided by the vendor
- Customer does not manage or control underlying infrastructure but can customize the applications
- Eg: Google App Engine, Microsoft Azure etc.

Features of PaaS

- All-in-one (dev+test+deploy+host+maintain)
- Web Access
- Offline access
- Built in Scalability
- Collaborative
- Diverse tools available

When to use PaaS?

- PaaS is a good option if:
  - Collaborative development
  - Automated testing and deployment
  - Less time to market
- PaaS might not be a good option if:
  - Frequent migration / Vendor Lock-In
  - Infrastructure customization
  - Platform flexibility
  - On-prem integration

PaaS – Pros and Cons

Pros

- Quick development and deployment
- Reduces ownership cost
- Agile software development
- Collaboration
- Ease of use
- Less maintenance
- Scalability

Cons

- Vendor Lock-In
- Security
- Less flexibility
- Depends on network connection

Infrastructure as a Service

- Provides ability to provision processing, storage, networks, and other resources
  - Deploy and run arbitrary software, OS and applications
- Customer does not manage or control underlying infrastructure but can control the OS, storage and deployed applications
- Eg: Amazon Web Services etc.

Features of IaaS

- Web Access to resources
- Centralized management
- Elasticity and dynamic scaling
- Shared infrastructure
- Preconfigured VMs
- Metered Services

IaaS – Pros and Cons

Pros

- Pay as you use
- Reduced operational cost
- Elastic resources
- Better resource utilization
- Green IT

Cons

- Security
- Interoperability / Vendor Lock-In
- Network performance

Comparison of Cloud Services

Traditional Setup	Infrastructure as a Service (IaaS)	Platform as a Service (PaaS)	Software as a Service (SaaS)
Data	Data	Data	Data
Applications	Applications	Applications	Applications
Runtime Environment	Runtime Environment	Runtime Environment	Runtime Environment
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Network	Network	Network	Network

Under user's control Under common control Under provider's control

Future of Cloud - XaaS

- Network as a Service (NaaS)
- Desktop as a Service (DEaaS)
- Storage as a Service (STaaS)
- DB as a Service (DBaaS)
- Data as a Service (DaaS)
- Security as a Service (SECaaS)
- Identity as a Service (IDaaS)

Cloud Deployment Models

Dr. Amit Praseed

How to host the Cloud?

- Remember Bob from ABC Corporation?
- He has decided to outsource their business to the cloud!!
- But he faces some difficult decisions!
  - Should he approach a third party?
  - Should he ask his IT team to build a cloud infrastructure on their company premises?

- At its core, the cloud is simply a datacentre, with software capable of managing virtualized resources according to user demand
- While choosing a cloud solution, consider
  - Where will the datacentre reside?
  - Who owns the datacentre?
  - Who can use resources within the datacentre?
  - Who operates and manages the datacentre?

Public vs Private Cloud

Public Cloud

- Provisioned for open use by general public
- Owned, managed and operated by business, academic or government organizations
- Exists on the premises of the cloud provider
- Eg: Gmail, Microsoft Azure, Dropbox etc.

Private Cloud

- Exclusively used by an organization
- Usually managed, operated and owned by the organization
- Usually resides on the organization premises
- Open source tools like OpenStack and Eucalyptus can be used to build private clouds

Here are the facts!

- ABC Corp. website has a significant user base right now, and is expected to grow over time
- The data maintained by the company is related to product details, and no financial details are maintained (assume)
- Which cloud model – private or public – would you recommend?

Here are the facts!

- What would be the case where there is also sensitive data that has to be maintained?
  - Credit card details
  - Company expansion plans
  - Other customer and employee data

Public Cloud Features

- Scalable
- Affordable
- Always available
- Stringent SLAs
- Less secure

When should one opt for Public Clouds?

- Larger user base
- Varying resource usage
- Lack of infrastructure
- Financial constraints

Private Cloud Features

- Secure
- More control available
- Weak SLAs

When should one opt for Private Clouds?

- Sufficient Funds
- Security and autonomy is paramount
- Few users
- Sufficient resources are available

Hybrid Cloud

- Organization manages both private and public clouds
- Services can be used from either cloud, depending on organizational policies
- Sometimes, a private cloud could be used be used till its capacity is met, after which the workload spills over to the public cloud – **Cloud Bursting**

Community Cloud

- Infrastructure is shared between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.)
- Managed internally or by a third-party and hosted internally or externally
- Eg: IBM SoftLayer cloud for federal agencies

Features of Community Cloud

- Collaborative effort
- No party has full control
- Partially secure
- Cost effective

When should one opt for Community Clouds?

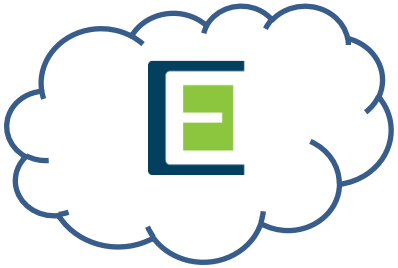
- Need for collaboration
- Financial constraints
- Less maintenance costs

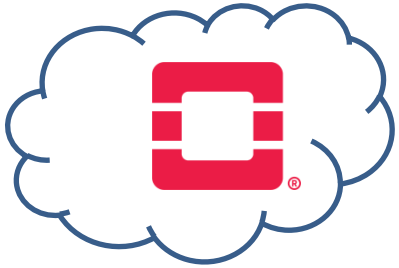
Cloud Deployment Models - Summary

- Depending on who owns and operates the cloud, and who can access the cloud, there are 4 deployment models
  - Public Cloud
  - Private Cloud
  - Community Cloud
  - Hybrid Cloud

Can a cloud run out of resources?

Can we connect the Clouds?





Intercloud

- Intercloud – “Cloud of Clouds”
- Mesh of clouds unified based on open standard protocols to provide interoperability
- Interconnects multiple cloud providers’ infrastructures.
  - Focus is on direct interoperability between public cloud service providers

Need for Intercloud

- Scalability + wider resource availability
- Interoperability + avoiding vendor lock-in
- Availability and disaster recovery
- Geographic distribution and low latency access
- Legal and regulatory issues

Types of Interclouds

- **Federation Clouds**: A set of cloud providers willingly interconnect their cloud infrastructures in order to share resources
  - Voluntary contribution
  - Suitable for collaboration of governmental clouds or private cloud portfolios
  - Types of federation clouds are Peer to Peer and Centralized clouds.
- **Multi-Cloud**: A client or service uses multiple independent clouds
  - No voluntary interconnection and sharing
  - Managing resource provisioning and scheduling is the responsibility of client or their representatives.
  - Used to utilize resources from both governmental clouds and private cloud portfolios.

Types of Federated Clouds

- **Peer to peer Inter-Cloud federation**
  - Clouds collaborate directly with each other but may use distributed entities for directories or brokering.
  - Eg: RESERVOIR (Resources and Services Virtualization without Barriers Project), Open Cirrus etc
- **Centralized Inter-Cloud federation**
  - Clouds use a central entity to perform or facilitate resource sharing.
  - The central entity acts as a storehouse where the available cloud resources are registered.
  - Eg: Contrail, Dynamic Cloud Collaboration (DCC) and Federated Cloud Management.

Types of Multi Clouds

- **Multicloud Service**
  - Clients access multiple clouds through a service.
  - A service is hosted by the cloud client either externally or in-house.
  - Eg: OPTIMIS, mOSAIC, STRATOS and Commercial Cloud Management Systems
- **Multicloud Libraries**
  - Clients develop their own brokers by using a unified cloud API as a library
  - Facilitate the usage of clouds in a uniform
  - Eg: Java library JClouds, Python library Apache LibClouds, Ruby library, Apache DeltaCloud, PHP library SimpleCloud, Apache Nuvem

Virtualization in Cloud Computing

Dr. Amit Praseed

How do Cloud Services Work?

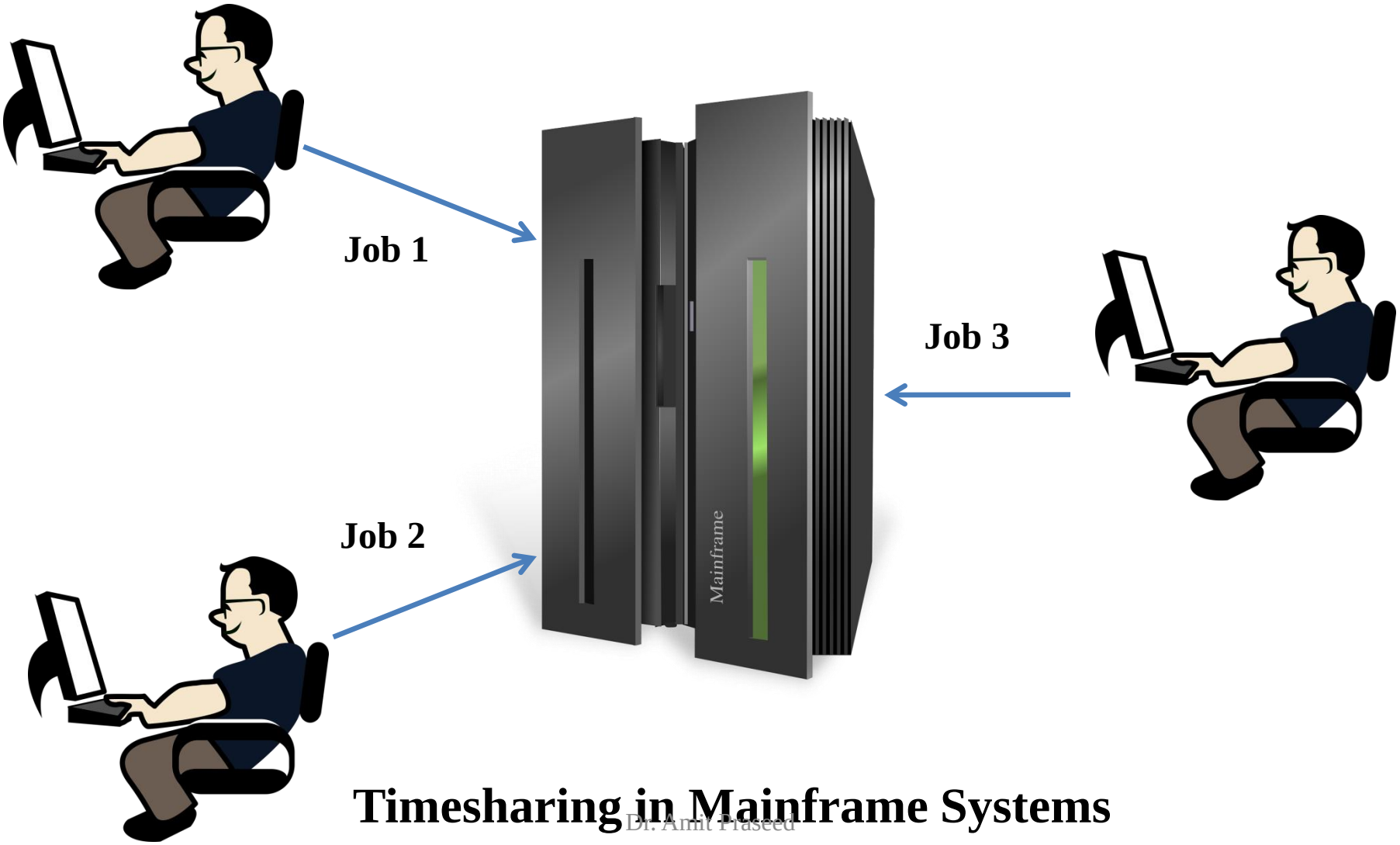
- Assume a cloud service provider has a datacentre with 4 CPUs and 8 GB RAM
  - Alice wants a system with 1 CPU and 2 GB RAM
  - Bob wants a system with 2 CPUs and 2 GB RAM
  - Carol wants a system with 1 CPU and 4 GB RAM
- In a traditional IT setup, this would be impossible!
- Solution: Create **virtual machines** with the required specifications and provide to the customers
- This uses a disruptive technology known as **virtualization**

- Put in simple terms, virtualization means

  creating an illusion of something which is not actually present

- Virtualization is used very commonly nowadays
  - **Virtual memory** gives us the illusion of a significantly larger memory than we physically have
  - **Virtual Reality** games allow users to perceive a world that doesn't physically exist

A Brief History of Virtualization

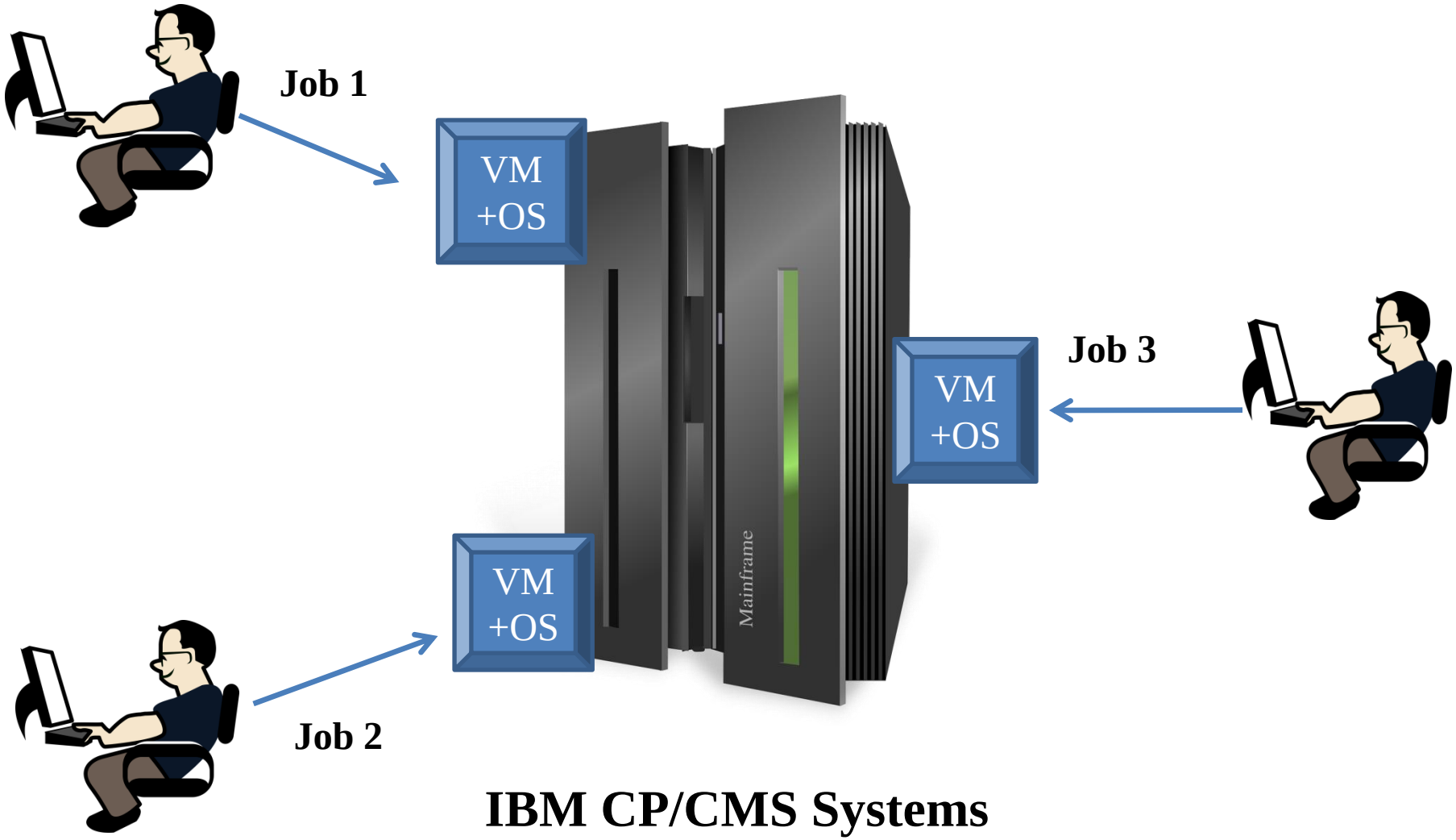


Timesharing in Mainframe Systems

A Brief History of Virtualization

- **Timesharing in Mainframes**
  - Support multiple users through terminals
  - When users block for I/O, system executes jobs from other users
  - System still executes only one job at a time
  - Creates an illusion of multiple jobs being processed at the same time
  - Later, a time quantum was introduced to increase server utilization

A Brief History of Virtualization

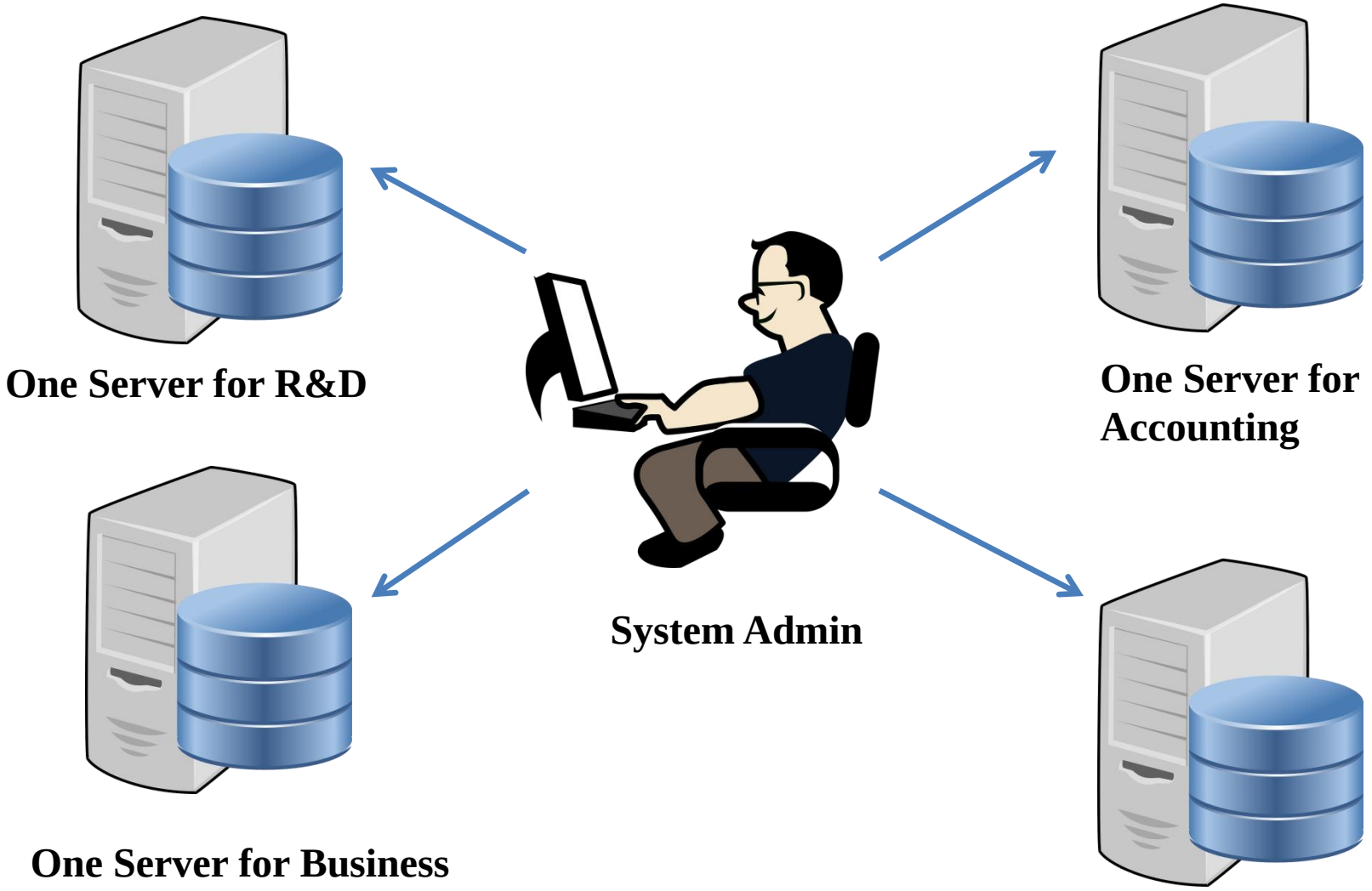


IBM CP/CMS Systems

A Brief History of Virtualization

- **IBM CP/CMS Systems**
  - First virtualized operating system
  - Every user gets a separate “virtual machine” for operating
  - Every user interacts with their own version of OS
  - No concept of time sharing – multiple tasks can be run simultaneously
  - No conflicts between users, so more reliable
  - The rise of personal computers led to a small decline in the importance of virtualization for a period of time

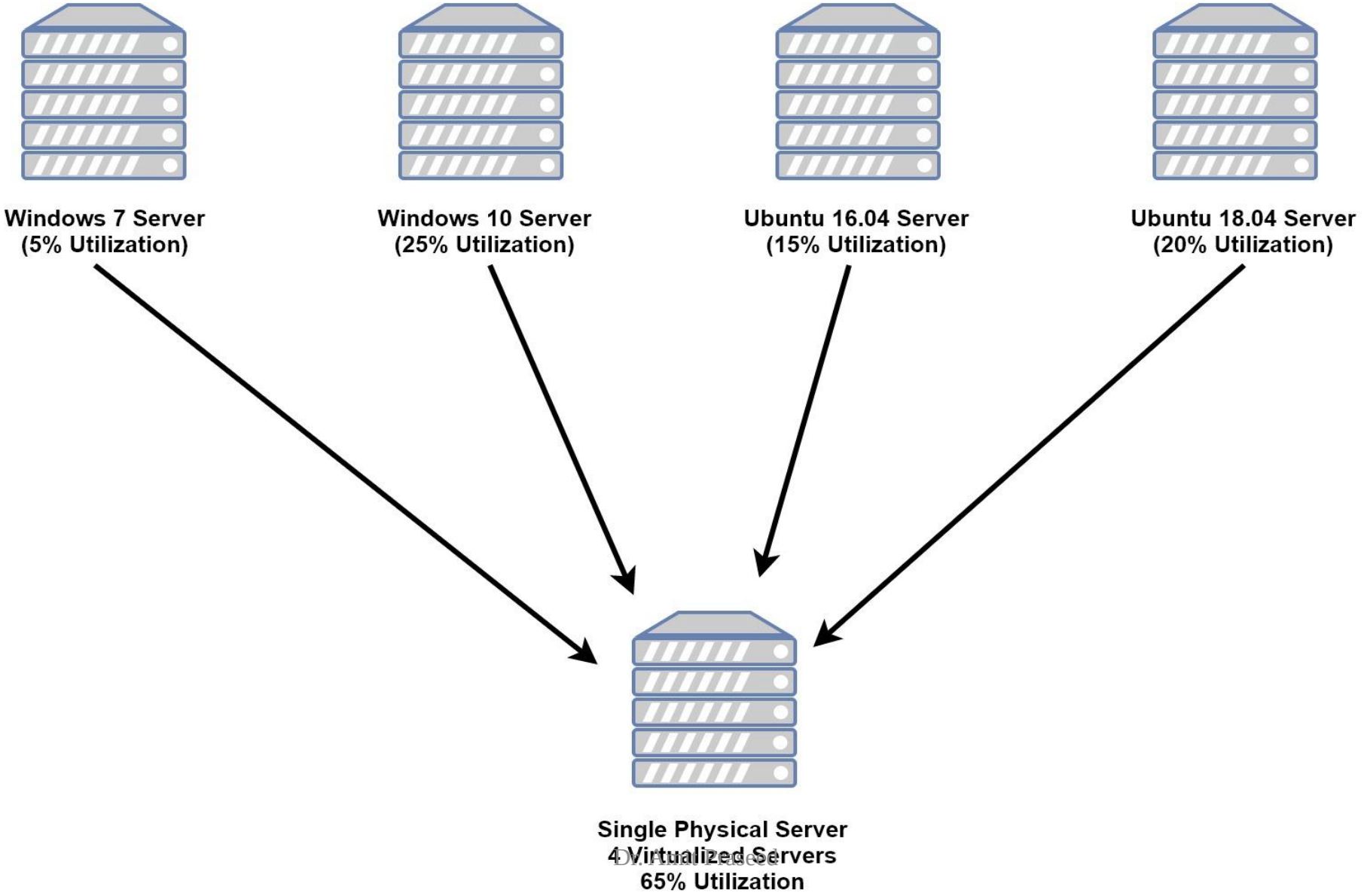
Need for Virtualization in Data Centres



Need for Virtualization in Data Centres

- System administrators allocated one machine per application
  - Increased stability – what if one application interfered with the other?
  - Increased security – hiding “sensitive” data
- Issues
  - Increased capital cost
  - Low server utilization

Virtualization



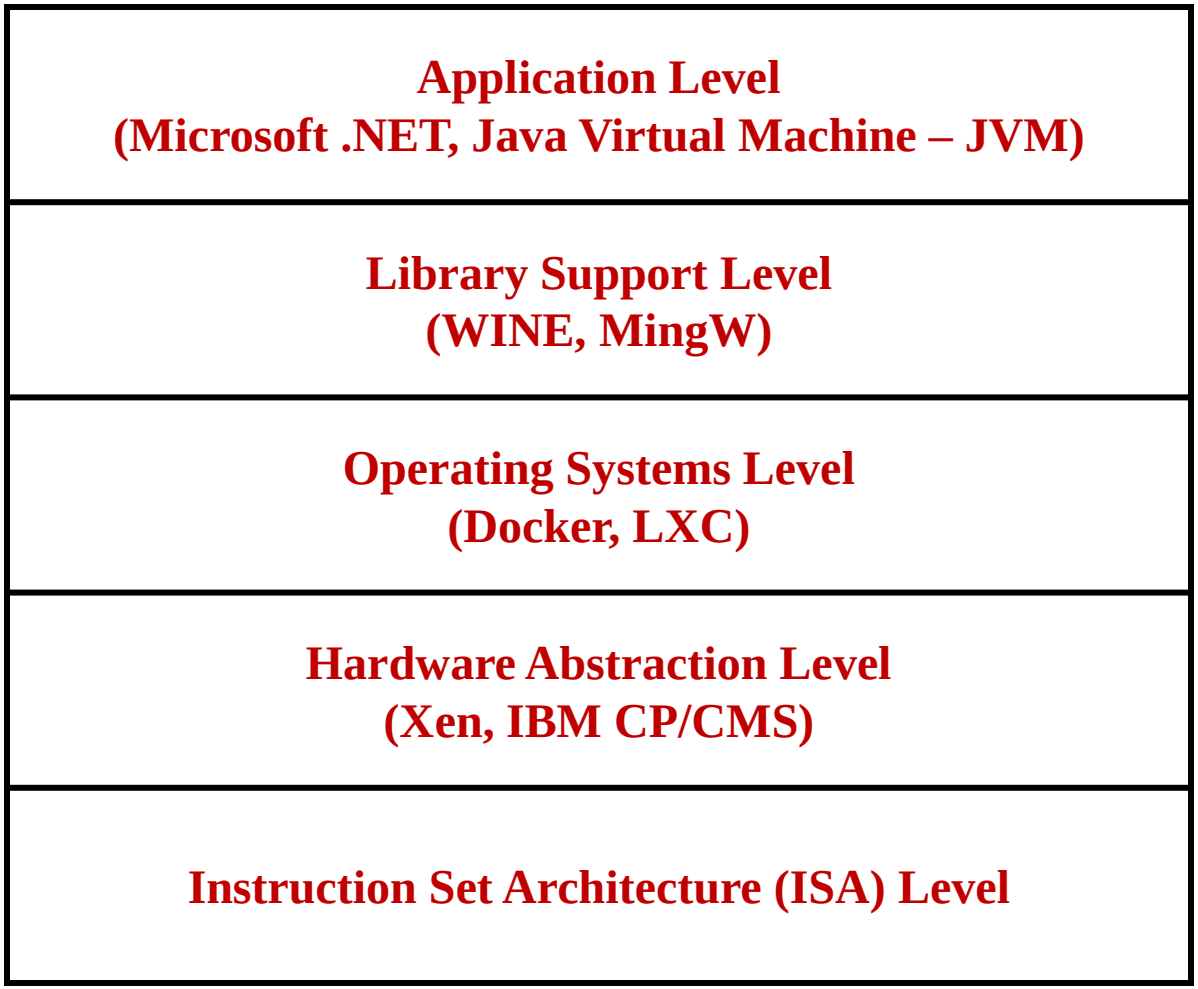
Virtualization

- Process of creating a function of a resource simulated or emulated in software identical to that of the corresponding physical resource
- **Two key points**:
  - It is a software simulation of a physical resource
  - Users must be able to use the virtualized resource exactly as they would use a physical resource

What can be virtualized?

- Desktop
- Application
- Server
- Storage
- Network

Levels of Virtualization



Merits of Different Types of Virtualization

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	X	XXXX	XXX	XXX
Hardware-level virtualization	XXXX	XXX	XXXX	XXX
OS-level virtualization	XXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXX	XXXX

Conditions for Effective Virtualization

- **Efficiency** : All innocuous instructions are executed by the hardware directly, with no intervention at all on the part of the control program
- **Resource Control** : It must be impossible for that arbitrary program to affect the system resources, i.e. memory, available to it; the allocator of the control program is to be invoked upon any attempt.
- **Equivalence** : Any program *K* executing with a control program resident performs in a manner indistinguishable from the case when the control program did not exist and *K* had whatever freedom of access to privileged instructions that the programmer had intended

[Popek and Goldberg 1974: "Formal Requirements for Virtualizable Third Generation Architectures" Communications of the ACM]

A Simple Solution - Emulation

- A machine could simply be emulated in software
  - File to represent disk
  - Case-statement to implement individual opcodes
  - Registers could be implemented as variables
- This violates the efficiency criteria!!!
  - Software does not allow effective implementations of hardware mechanisms such as interrupts
  - Each guest instruction is executed by several host instructions.
- **Instead, can we directly use the host hardware?**
  - VM executes as a process on host
  - Host processor executes its instructions.

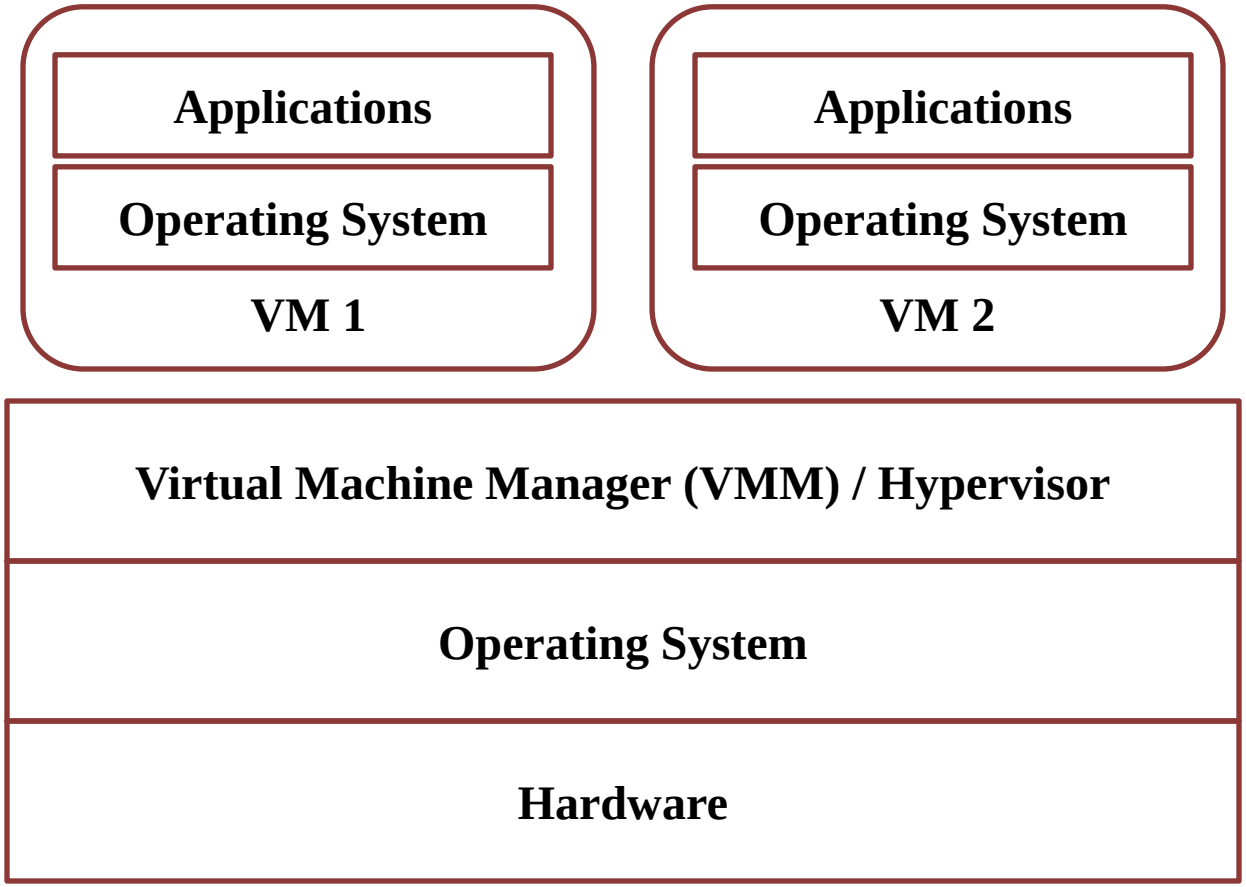
Revisiting the Rules

- **Rules for Efficient Virtualization**
  - **Efficiency** : All innocuous instructions are executed by the hardware directly, with no intervention at all on the part of the **control program**
  - **Resource Control** : It must be impossible for that arbitrary program to affect the system resources, i.e. memory, available to it; the allocator of the **control program** is to be invoked upon any attempt.
  - **Equivalence** : Any program *K* executing with a **control program** resident performs in a manner indistinguishable from the case when the **control program** did not exist and *K* had whatever freedom of access to privileged instructions that the programmer had intended
- The "CONTROL PROGRAM" is usually a layer of software that mediates between the VMs and the underlying hardware
- Called a Virtual Machine Manager (VMM) or hypervisor

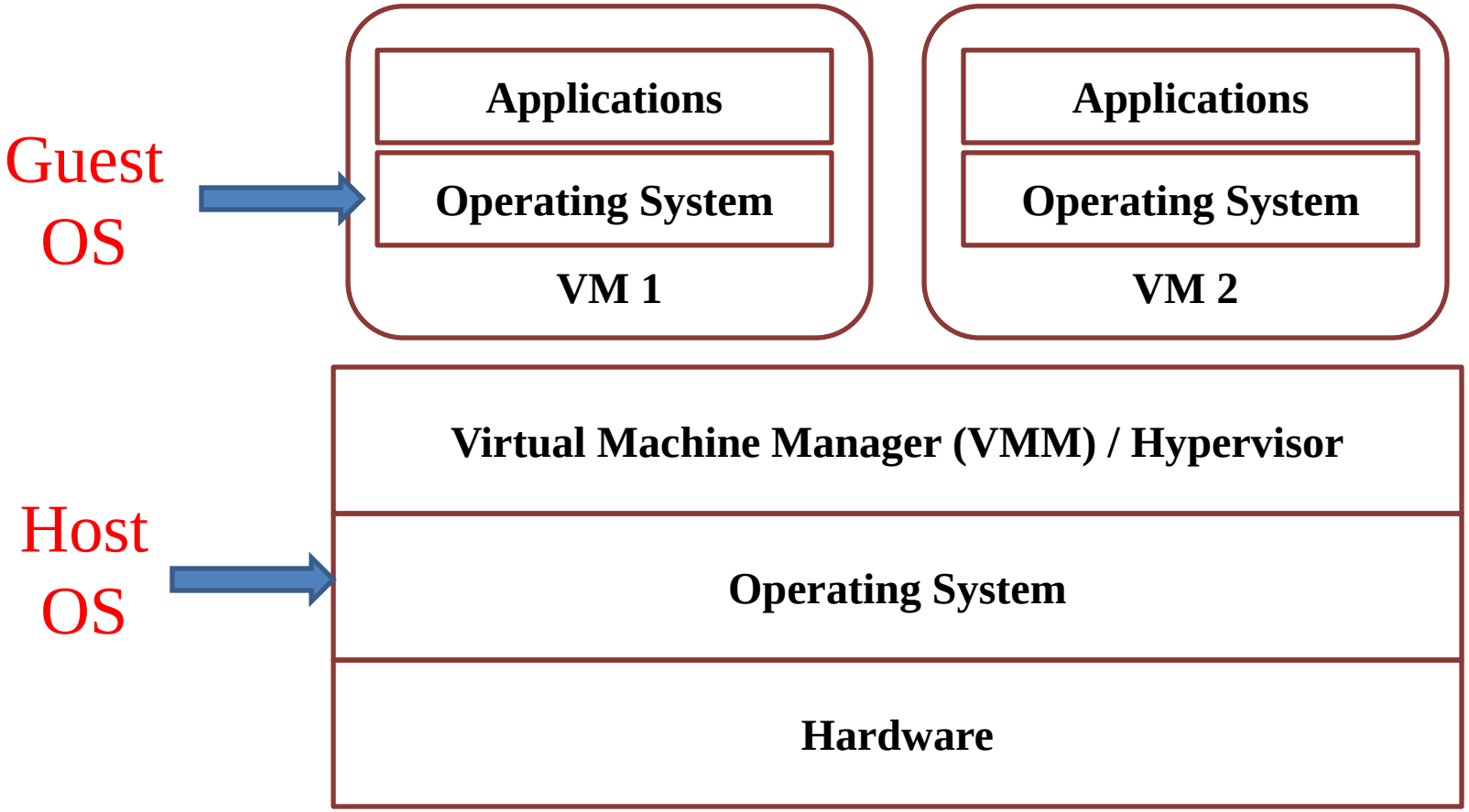
Role of Hypervisor

- Provide an environment for programs which is essentially identical to the original machine
- Ensure that programs run in this environment should show, at worst, only minor decreases in speed
- Ensure complete control of the system resources.
  - Allocation
  - Separation
  - Preemption

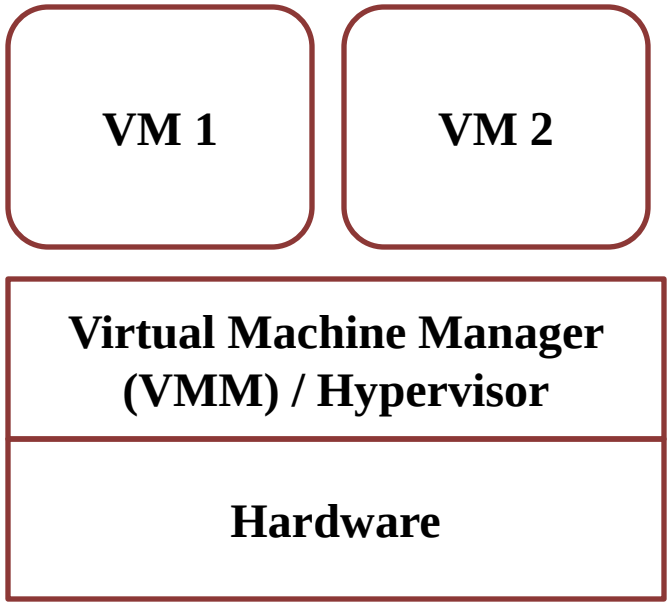
A General Architecture



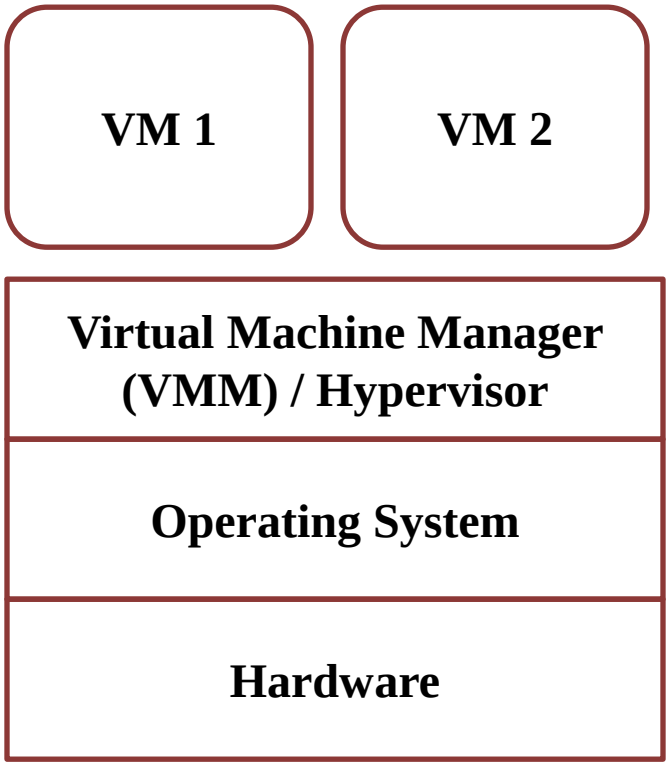
A General Architecture



Types of Hypervisors



Type 1 (Bare-Metal) Hypervisor



Type 2 (Hosted) Hypervisor

Comparison of Hypervisors

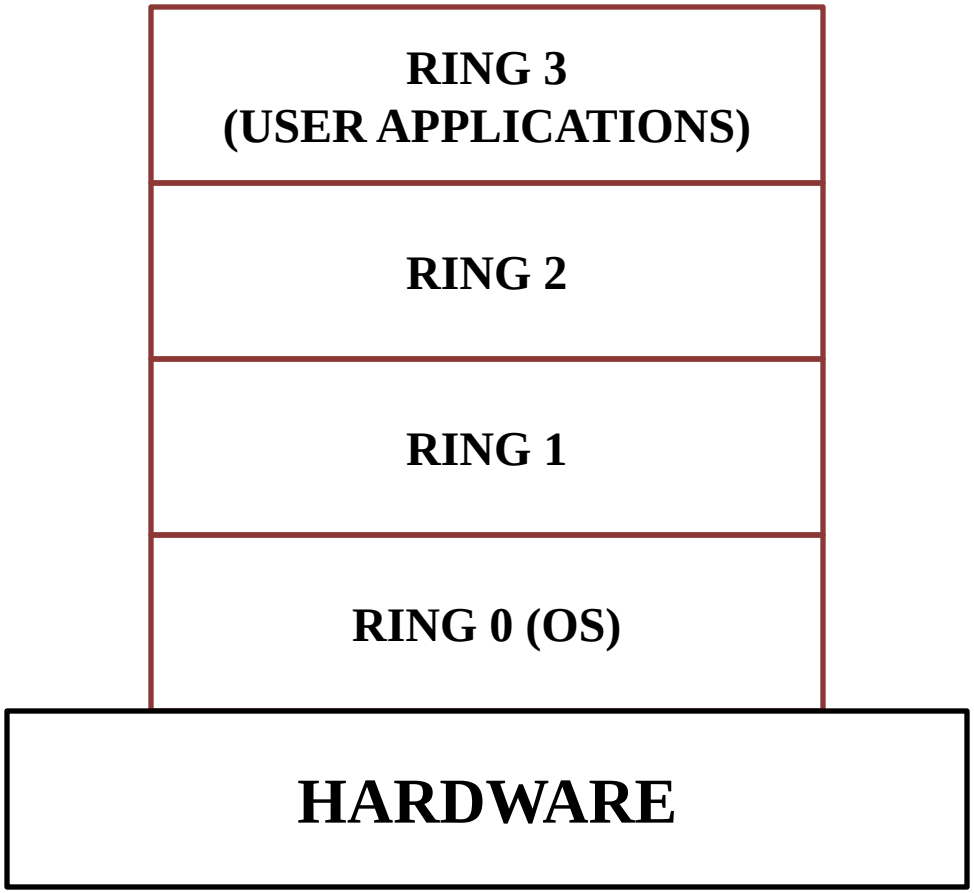
Type - 1 Hypervisor

- Resides directly on the hardware (“bare metal”)
- Communicates directly with the hardware resources
- More efficient
- More secure
- Eg: Citrix/Xen Server, VMware ESXi and Microsoft Hyper-V

Type – 2 Hypervisor

- Resides on top of the operating system (“hosted”)
- Communicates with hardware through the OS
- Less efficient
- Less secure
- Eg: Oracle Virtual Box, VMware Workstation etc.

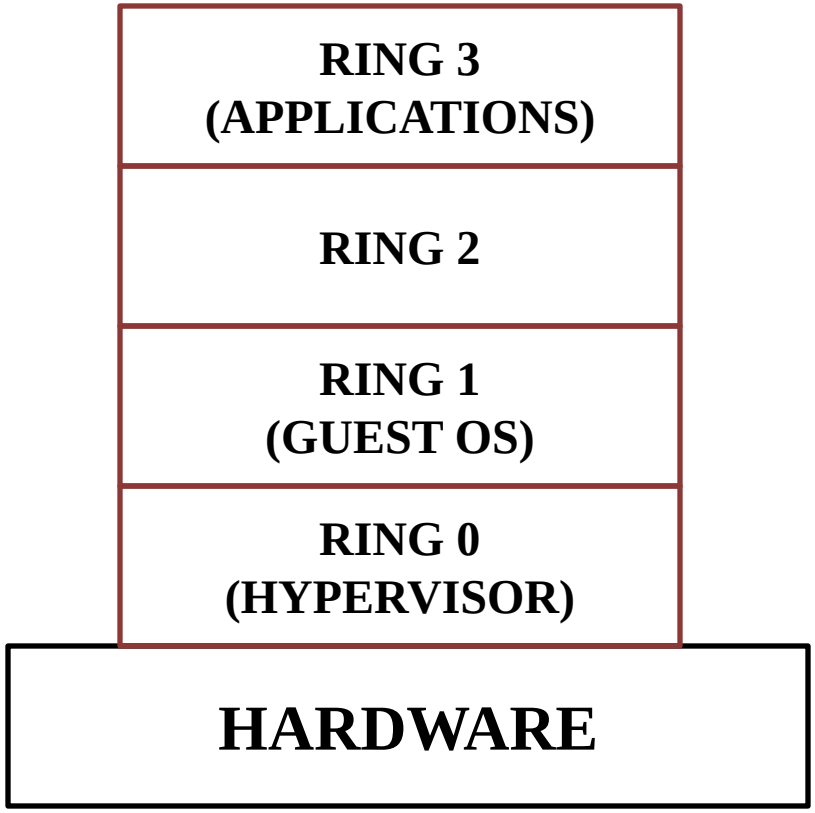
“Protection Ring” Concept



The Difficulty with Virtualization

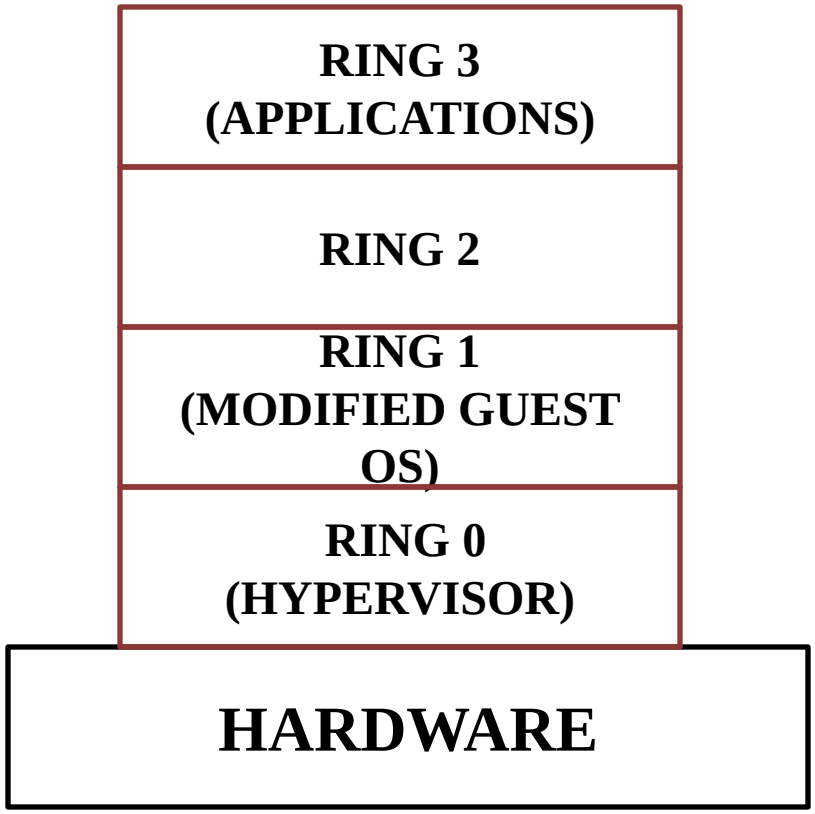
- All modern operating systems are built to run at Ring 0
  - They are designed to issue privileged instructions designed to modify memory and hardware directly
- For virtualization, guest OS resides on top of a hypervisor
  - Guest OS can only operate at a Ring > 0
  - This causes problems when the guest OS issues privileged instructions
  - The hypervisor must intercept and translate privileged instructions before passing it over to the hardware

Full Virtualization



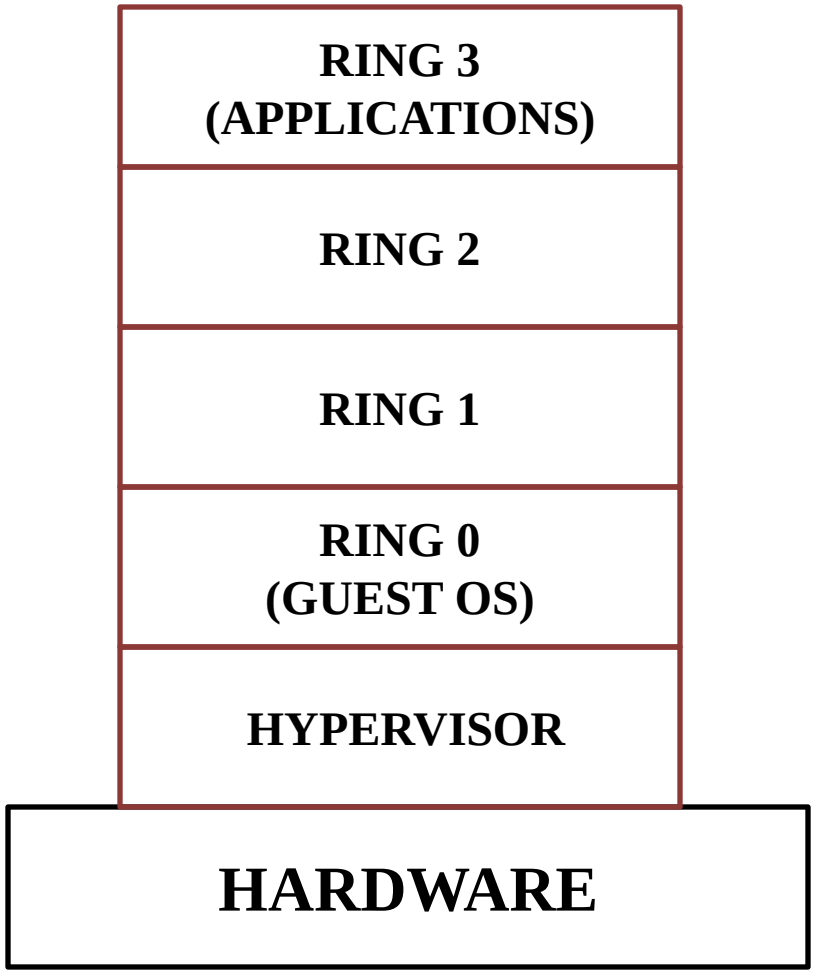
- Hypervisor operates at Ring 0
- Hypervisor scans the request stream
  - Captures and translates privileged instructions
  - Guest OS thinks it is directly working with hardware
- Performance is impacted due to binary translation

Para Virtualization



- Hypervisor resides in a privileged layer beneath the guest OS
- Guest OS is modified, so it doesn't execute privileged instructions
  - It executes hypercalls to the hypervisor
- Better performance
  - Limited use due to the need to modify OS

Hardware Assisted Virtualization



- Hardware allows hypervisor to reside in a privileged ring
- Privileged and sensitive calls are set to automatically trap to the hypervisor
- Can use unmodified OS + better performance
- Requires hardware support

Types of Virtualization

Type of Virtualization	Requires Hardware Support?	Requires Guest OS Modification?
Full Virtualization	No	No
Para virtualization	No	Yes
Hardware Assisted Virtualization	Yes	No

Cloud Service Models

Dr. Amit Praseed

Can you name some cloud services which you use?

Cloud Services

- There are three types of services offered by cloud service providers
  - Software as a Service (SaaS)
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)

Software as a Service

- Provides access to applications running on a cloud infrastructure, including network, servers, OS, storage or individual applications
- Accessible from client devices via a browser or program interface
- Customer does not manage or control underlying infrastructure or applications – limited customization
- Eg: Google Documents, Sharepoint etc.

SaaS Features

- One to Many
- Web Access
- Centralized Management
- Multi-device support
- Scalability
- Availability
- API integration

When to use SaaS?

- You should use SaaS for:
  - On demand software
  - Software for startups
  - Software compatibility
  - Varying loads
- SaaS might not be a good option if:
  - Unreliable network connection
  - Confidential data

SaaS - Pros and Cons

Pros

- No installation
- Cost savings
- Less maintenance
- Ease of Access
- Dynamic scalability
- Disaster Recovery
- Multitenancy

Cons

- Security
- Connectivity
- Loss of Control

Platform as a Service

- Provides ability to create and/or deploy applications on cloud infrastructure
  - Languages, libraries, services and tools are provided by the vendor
- Customer does not manage or control underlying infrastructure but can customize the applications
- Eg: Google App Engine, Microsoft Azure etc.

Features of PaaS

- All-in-one (dev+test+deploy+host+maintain)
- Web Access
- Offline access
- Built in Scalability
- Collaborative
- Diverse tools available

When to use PaaS?

- PaaS is a good option if:
  - Collaborative development
  - Automated testing and deployment
  - Less time to market
- PaaS might not be a good option if:
  - Frequent migration / Vendor Lock-In
  - Infrastructure customization
  - Platform flexibility
  - On-prem integration

PaaS – Pros and Cons

Pros

- Quick development and deployment
- Reduces ownership cost
- Agile software development
- Collaboration
- Ease of use
- Less maintenance
- Scalability

Cons

- Vendor Lock-In
- Security
- Less flexibility
- Depends on network connection

Infrastructure as a Service

- Provides ability to provision processing, storage, networks, and other resources
  - Deploy and run arbitrary software, OS and applications
- Customer does not manage or control underlying infrastructure but can control the OS, storage and deployed applications
- Eg: Amazon Web Services etc.

Features of IaaS

- Web Access to resources
- Centralized management
- Elasticity and dynamic scaling
- Shared infrastructure
- Preconfigured VMs
- Metered Services

IaaS – Pros and Cons

Pros

- Pay as you use
- Reduced operational cost
- Elastic resources
- Better resource utilization
- Green IT

Cons

- Security
- Interoperability / Vendor Lock-In
- Network performance

Comparison of Cloud Services

Traditional Setup	Infrastructure as a Service (IaaS)	Platform as a Service (PaaS)	Software as a Service (SaaS)
Data	Data	Data	Data
Applications	Applications	Applications	Applications
Runtime Environment	Runtime Environment	Runtime Environment	Runtime Environment
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Network	Network	Network	Network

Under user's control Under common control Under provider's control

Future of Cloud - XaaS

- Network as a Service (NaaS)
- Desktop as a Service (DEaaS)
- Storage as a Service (STaaS)
- DB as a Service (DBaaS)
- Data as a Service (DaaS)
- Security as a Service (SECaaS)
- Identity as a Service (IDaaS)

Virtualization in Cloud Computing

Dr. Amit Praseed

How do Cloud Services Work?

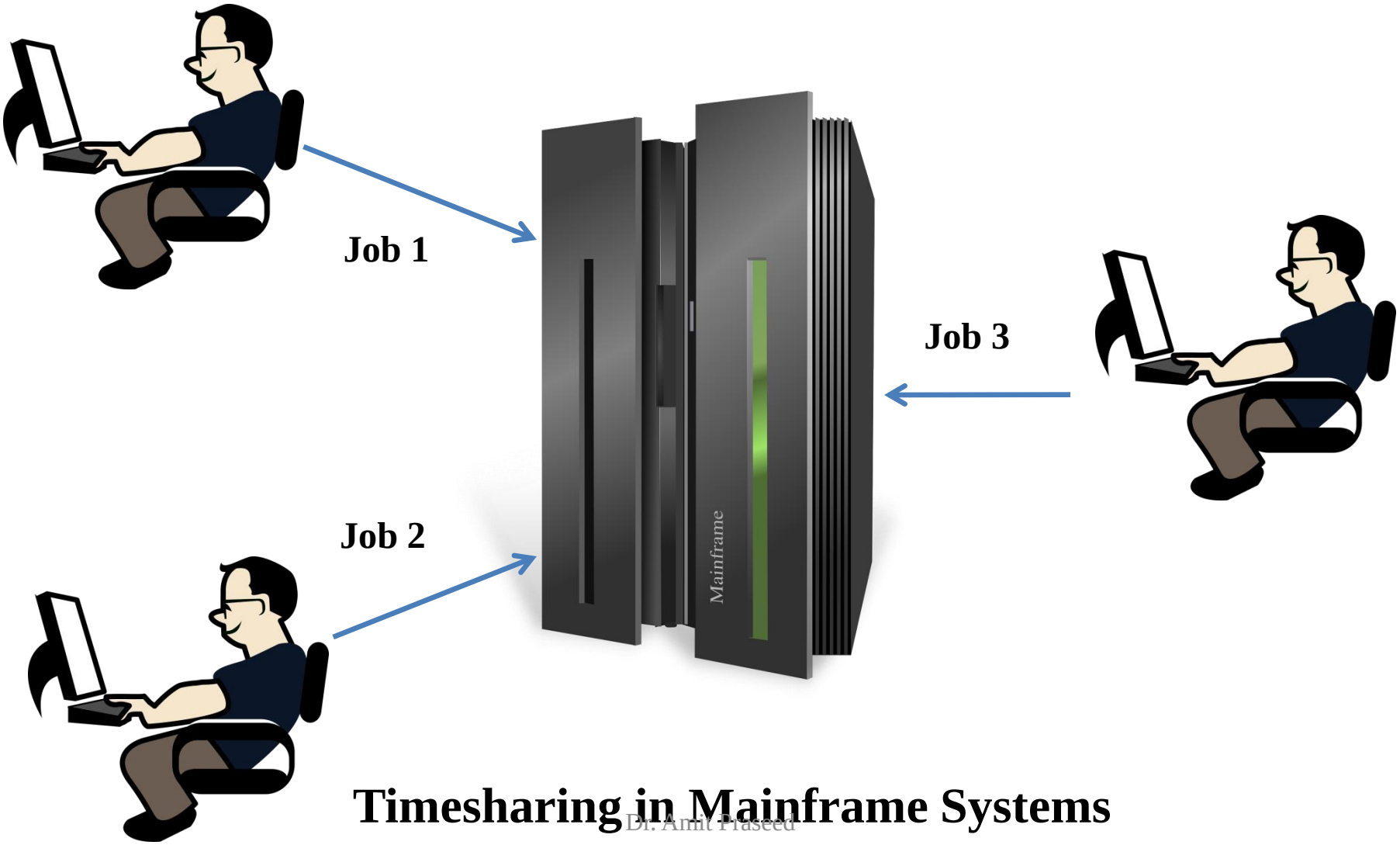
- Assume a cloud service provider has a datacentre with 4 CPUs and 8 GB RAM
  - Alice wants a system with 1 CPU and 2 GB RAM
  - Bob wants a system with 2 CPUs and 2 GB RAM
  - Carol wants a system with 1 CPU and 4 GB RAM
- In a traditional IT setup, this would be impossible!
- Solution: Create **virtual machines** with the required specifications and provide to the customers
- This uses a disruptive technology known as **virtualization**

- Put in simple terms, virtualization means

  creating an illusion of something which is not actually present

- Virtualization is used very commonly nowadays
  - **Virtual memory** gives us the illusion of a significantly larger memory than we physically have
  - **Virtual Reality** games allow users to perceive a world that doesn't physically exist

A Brief History of Virtualization

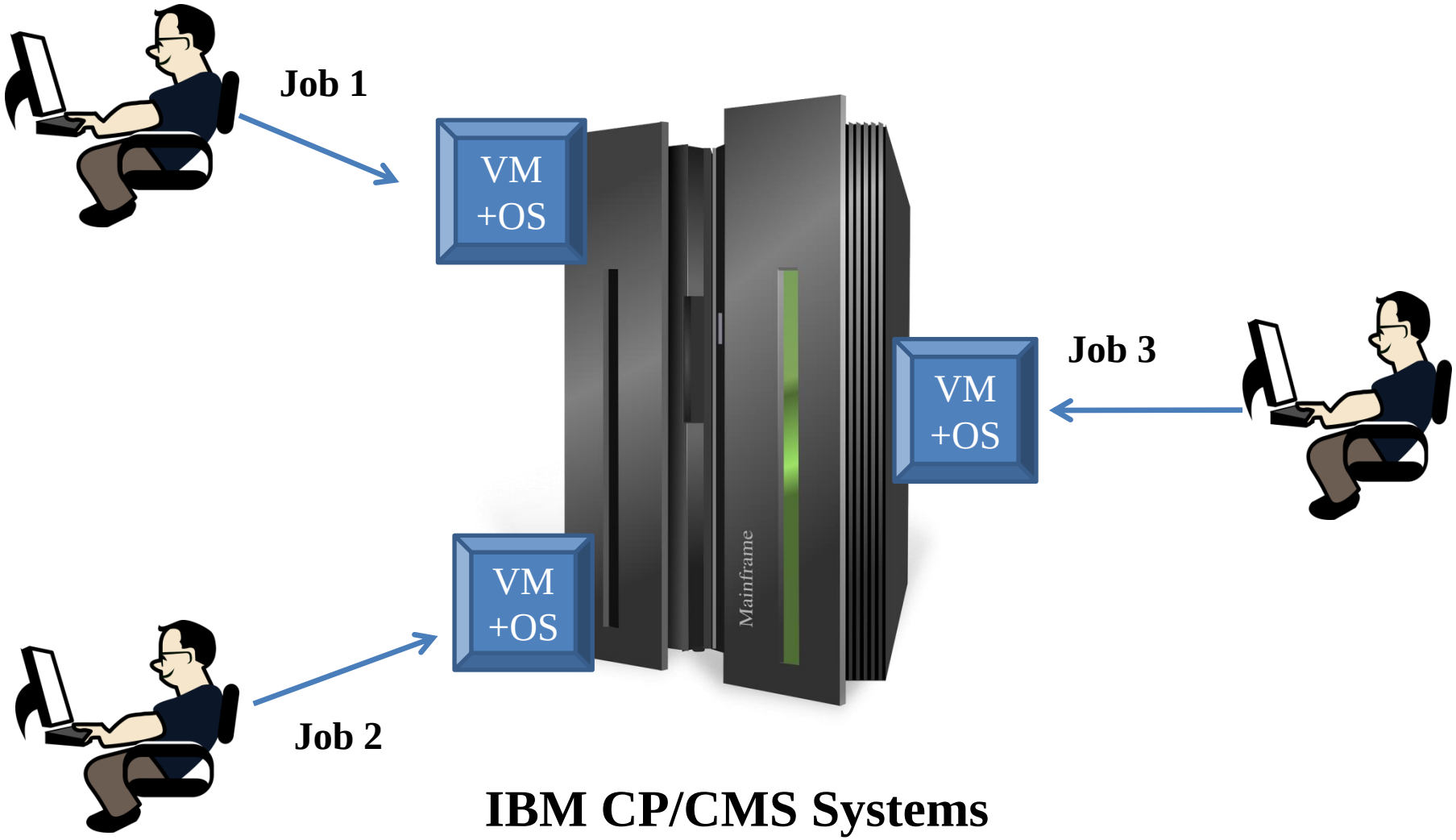


Timesharing in Mainframe Systems

A Brief History of Virtualization

- **Timesharing in Mainframes**
  - Support multiple users through terminals
  - When users block for I/O, system executes jobs from other users
  - System still executes only one job at a time
  - Creates an illusion of multiple jobs being processed at the same time
  - Later, a time quantum was introduced to increase server utilization

A Brief History of Virtualization

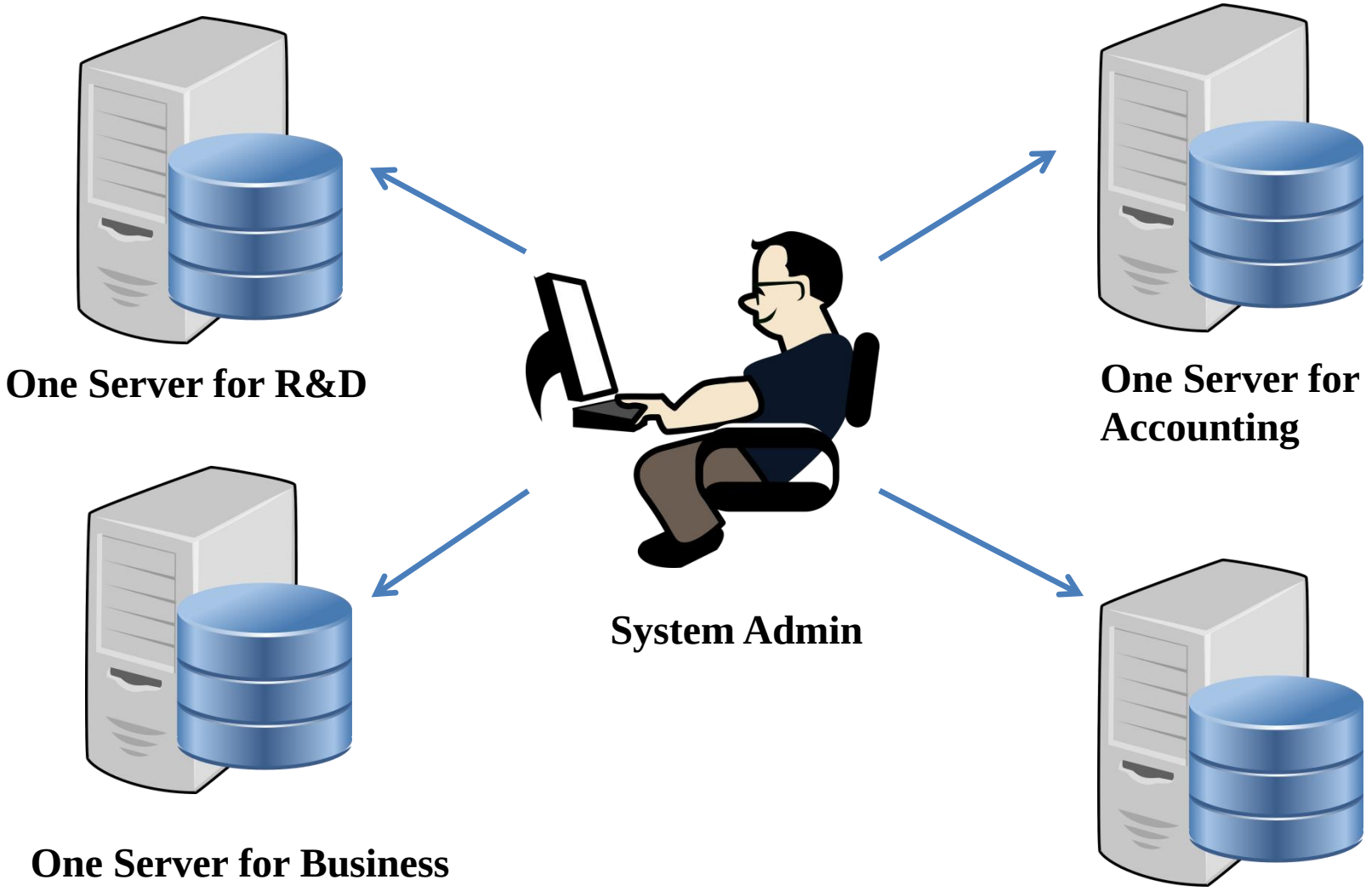


IBM CP/CMS Systems

A Brief History of Virtualization

- **IBM CP/CMS Systems**
  - First virtualized operating system
  - Every user gets a separate “virtual machine” for operating
  - Every user interacts with their own version of OS
  - No concept of time sharing – multiple tasks can be run simultaneously
  - No conflicts between users, so more reliable
  - The rise of personal computers led to a small decline in the importance of virtualization for a period of time

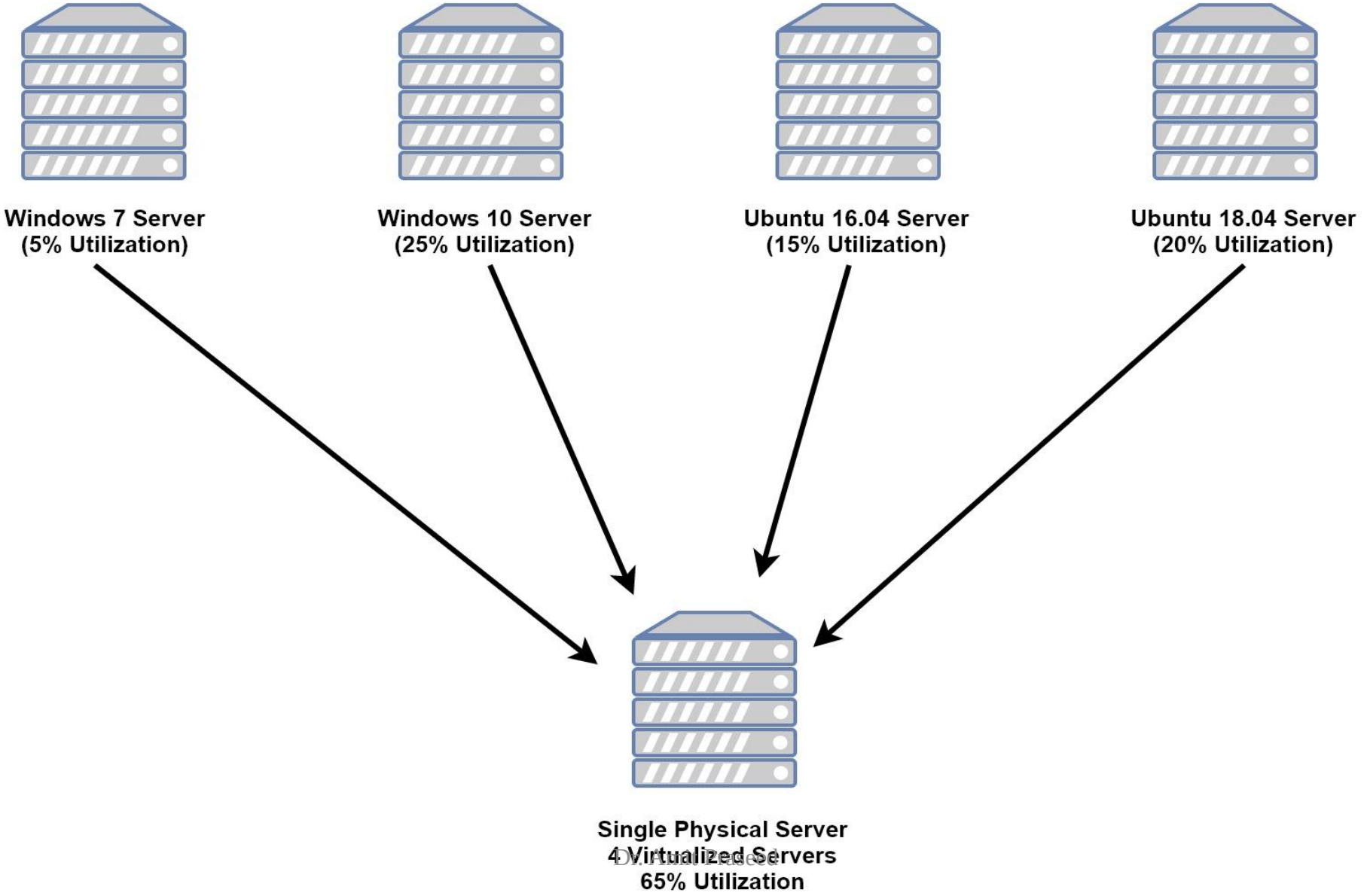
Need for Virtualization in Data Centres



Need for Virtualization in Data Centres

- System administrators allocated one machine per application
  - Increased stability – what if one application interfered with the other?
  - Increased security – hiding “sensitive” data
- Issues
  - Increased capital cost
  - Low server utilization

Virtualization



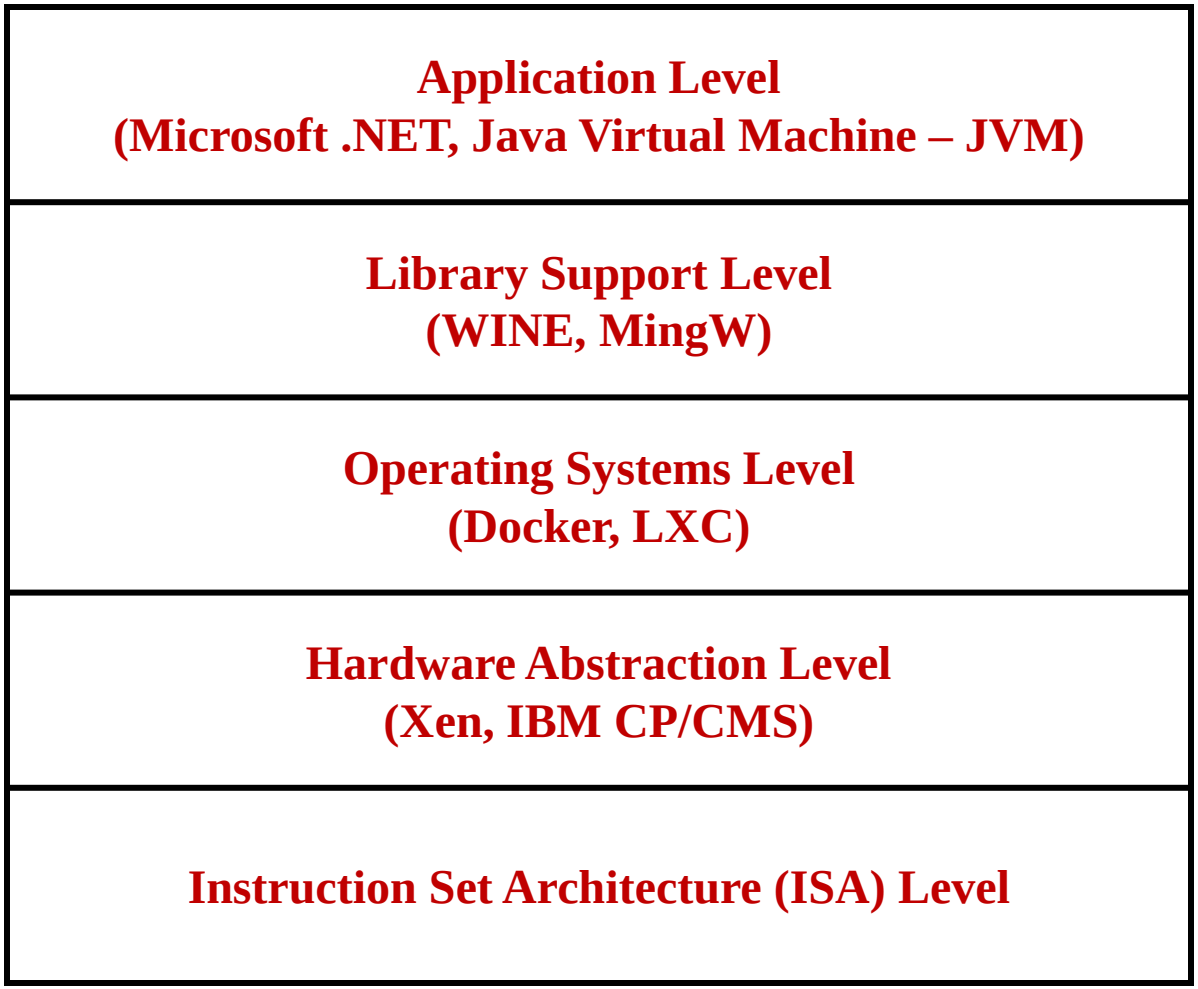
Virtualization

- Process of creating a function of a resource simulated or emulated in software identical to that of the corresponding physical resource
- **Two key points**:
  - It is a software simulation of a physical resource
  - Users must be able to use the virtualized resource exactly as they would use a physical resource

What can be virtualized?

- Desktop
- Application
- Server
- Storage
- Network

Levels of Virtualization



Merits of Different Types of Virtualization

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	X	XXXX	XXX	XXX
Hardware-level virtualization	XXXX	XXX	XXXX	XXX
OS-level virtualization	XXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXX	XXXX

Conditions for Effective Virtualization

- **Efficiency** : All innocuous instructions are executed by the hardware directly, with no intervention at all on the part of the control program
- **Resource Control** : It must be impossible for that arbitrary program to affect the system resources, i.e. memory, available to it; the allocator of the control program is to be invoked upon any attempt.
- **Equivalence** : Any program *K* executing with a control program resident performs in a manner indistinguishable from the case when the control program did not exist and *K* had whatever freedom of access to privileged instructions that the programmer had intended

[Popek and Goldberg 1974: "Formal Requirements for Virtualizable Third Generation Architectures" Communications of the ACM]

A Simple Solution - Emulation

- A machine could simply be emulated in software
  - File to represent disk
  - Case-statement to implement individual opcodes
  - Registers could be implemented as variables
- This violates the efficiency criteria!!!
  - Software does not allow effective implementations of hardware mechanisms such as interrupts
  - Each guest instruction is executed by several host instructions.
- **Instead, can we directly use the host hardware?**
  - VM executes as a process on host
  - Host processor executes its instructions.

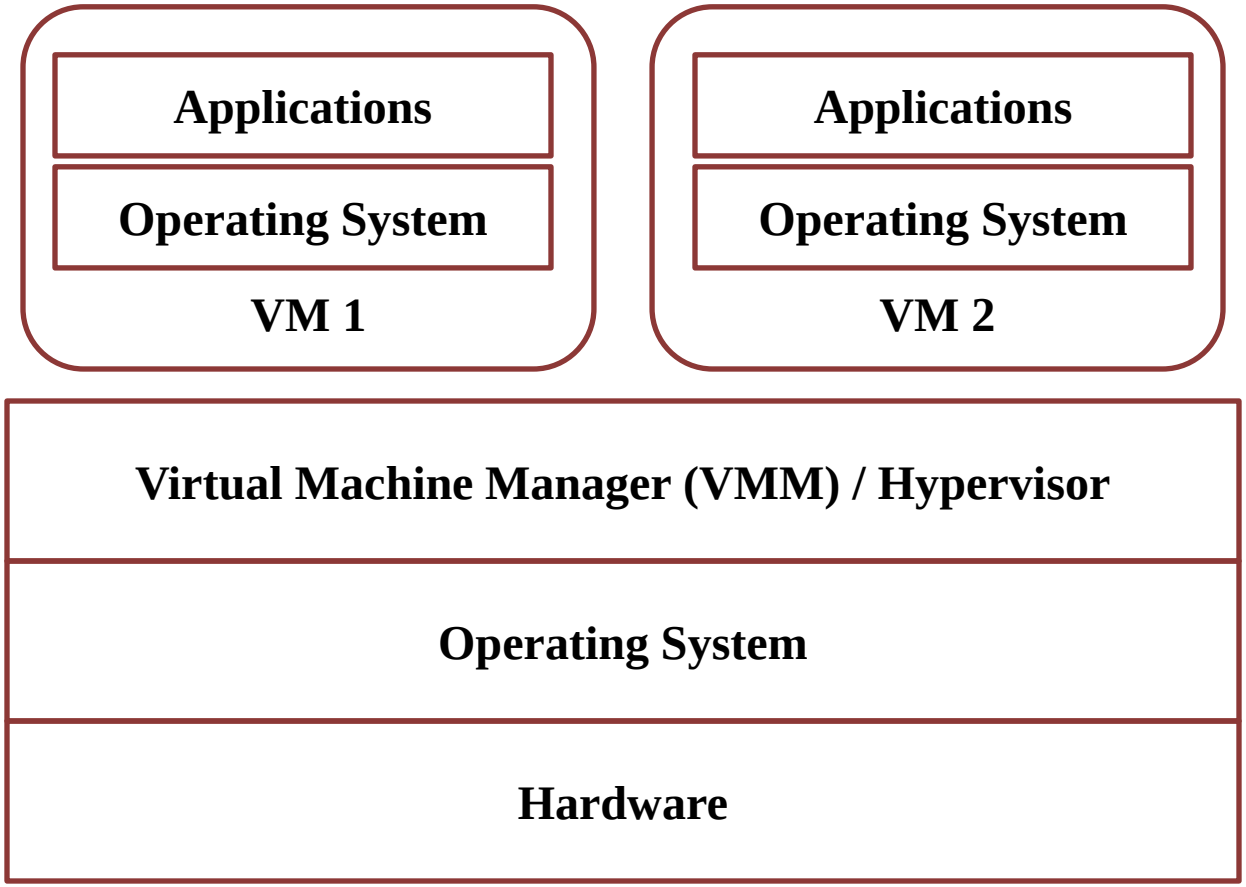
Revisiting the Rules

- **Rules for Efficient Virtualization**
  - **Efficiency** : All innocuous instructions are executed by the hardware directly, with no intervention at all on the part of the **control program**
  - **Resource Control** : It must be impossible for that arbitrary program to affect the system resources, i.e. memory, available to it; the allocator of the **control program** is to be invoked upon any attempt.
  - **Equivalence** : Any program *K* executing with a **control program** resident performs in a manner indistinguishable from the case when the **control program** did not exist and *K* had whatever freedom of access to privileged instructions that the programmer had intended
- The "CONTROL PROGRAM" is usually a layer of software that mediates between the VMs and the underlying hardware
- Called a Virtual Machine Manager (VMM) or hypervisor

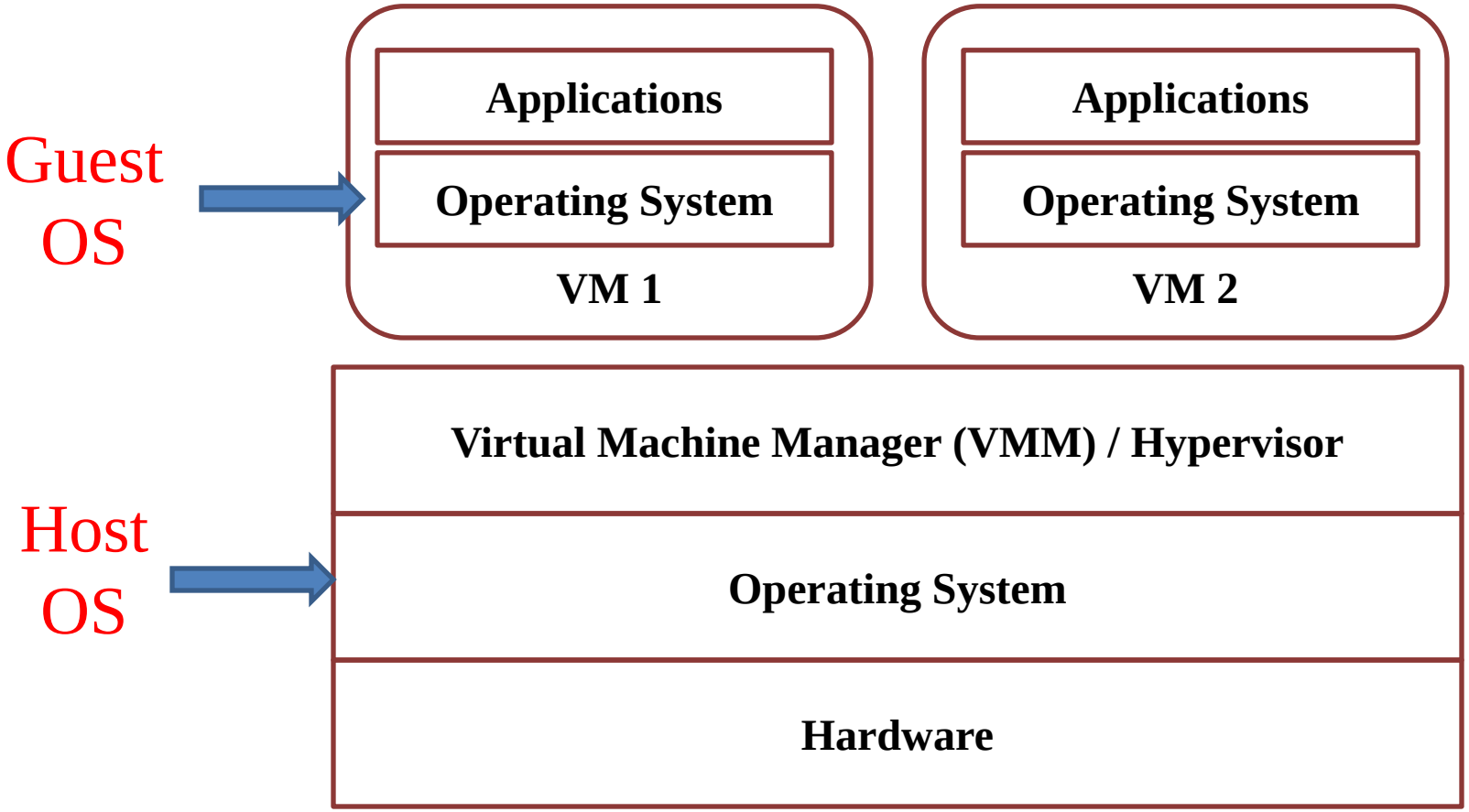
Role of Hypervisor

- Provide an environment for programs which is essentially identical to the original machine
- Ensure that programs run in this environment should show, at worst, only minor decreases in speed
- Ensure complete control of the system resources.
  - Allocation
  - Separation
  - Preemption

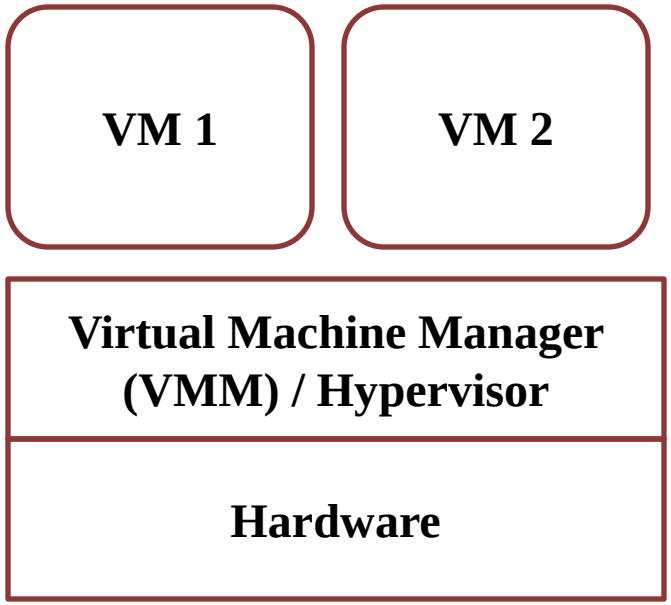
A General Architecture



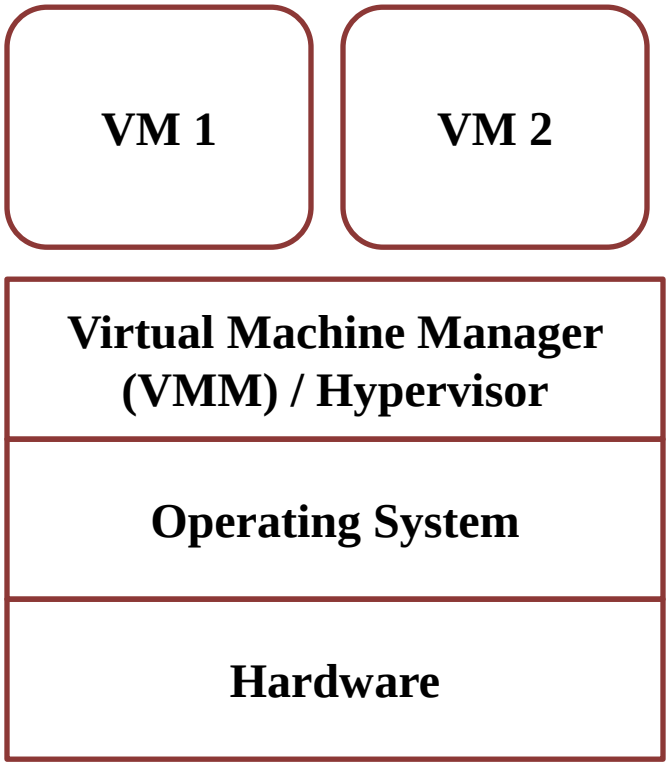
A General Architecture



Types of Hypervisors



Type 1 (Bare-Metal) Hypervisor



Type 2 (Hosted) Hypervisor

Comparison of Hypervisors

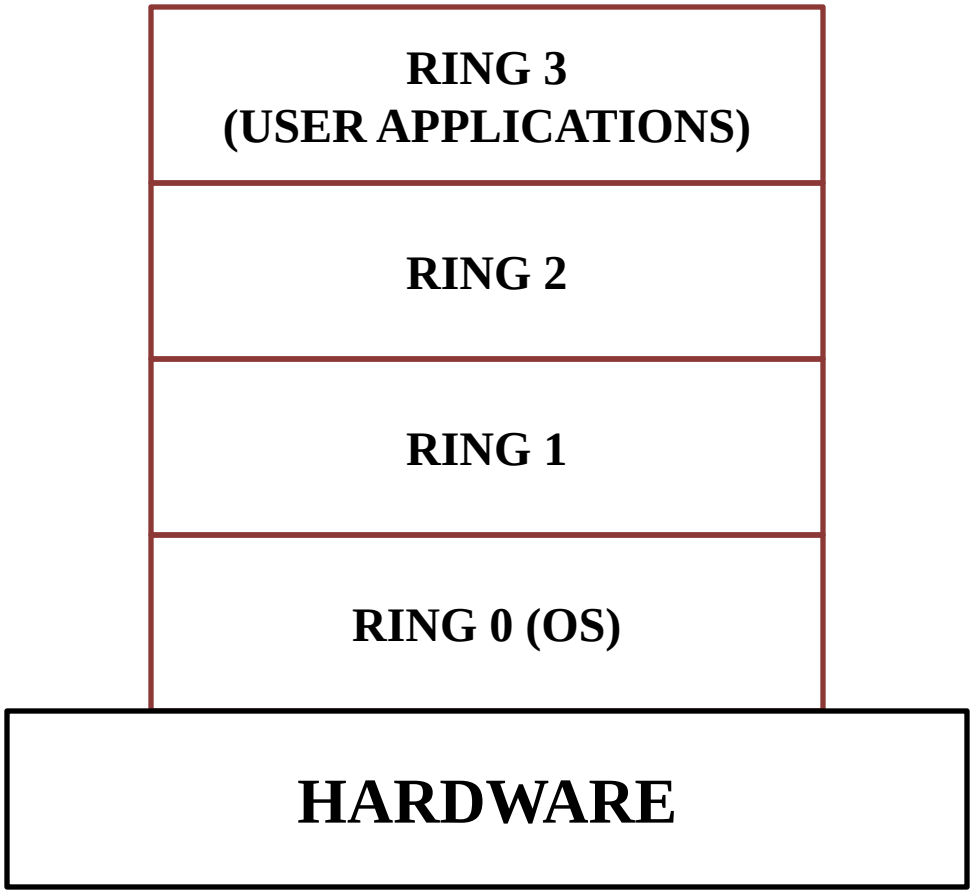
Type - 1 Hypervisor

- Resides directly on the hardware (“bare metal”)
- Communicates directly with the hardware resources
- More efficient
- More secure
- Eg: Citrix/Xen Server, VMware ESXi and Microsoft Hyper-V

Type – 2 Hypervisor

- Resides on top of the operating system (“hosted”)
- Communicates with hardware through the OS
- Less efficient
- Less secure
- Eg: Oracle Virtual Box, VMware Workstation etc.

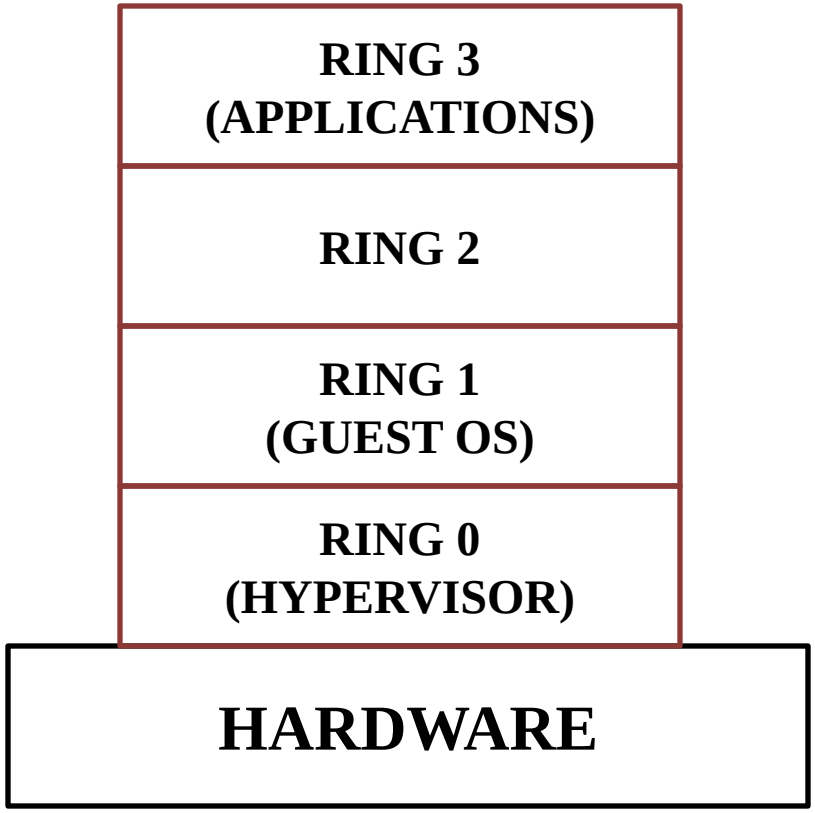
“Protection Ring” Concept



The Difficulty with Virtualization

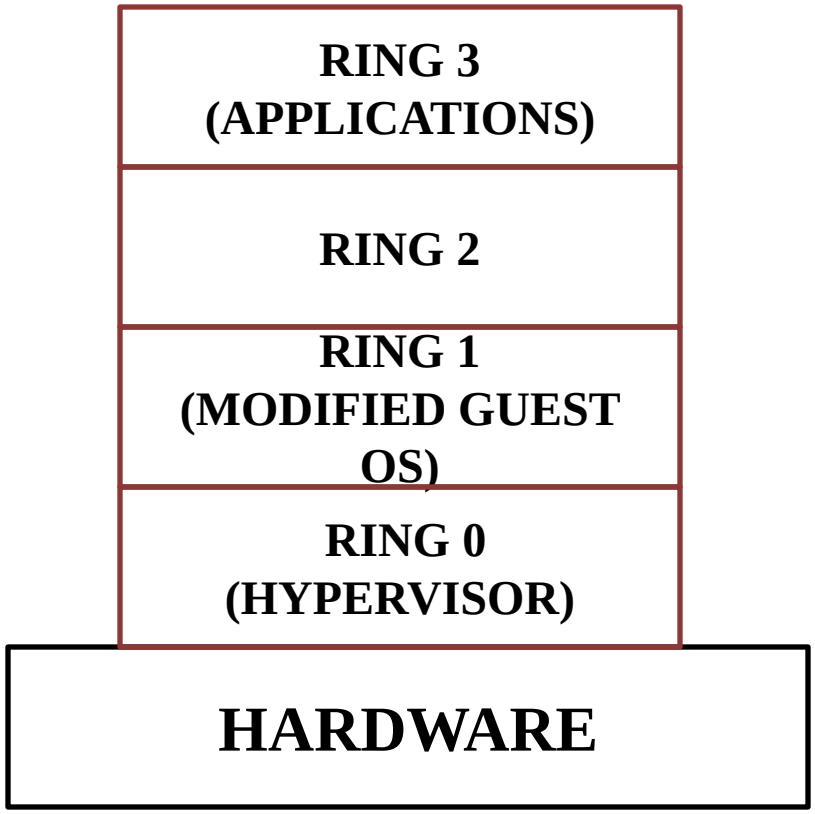
- All modern operating systems are built to run at Ring 0
  - They are designed to issue privileged instructions designed to modify memory and hardware directly
- For virtualization, guest OS resides on top of a hypervisor
  - Guest OS can only operate at a Ring > 0
  - This causes problems when the guest OS issues privileged instructions
  - The hypervisor must intercept and translate privileged instructions before passing it over to the hardware

Full Virtualization



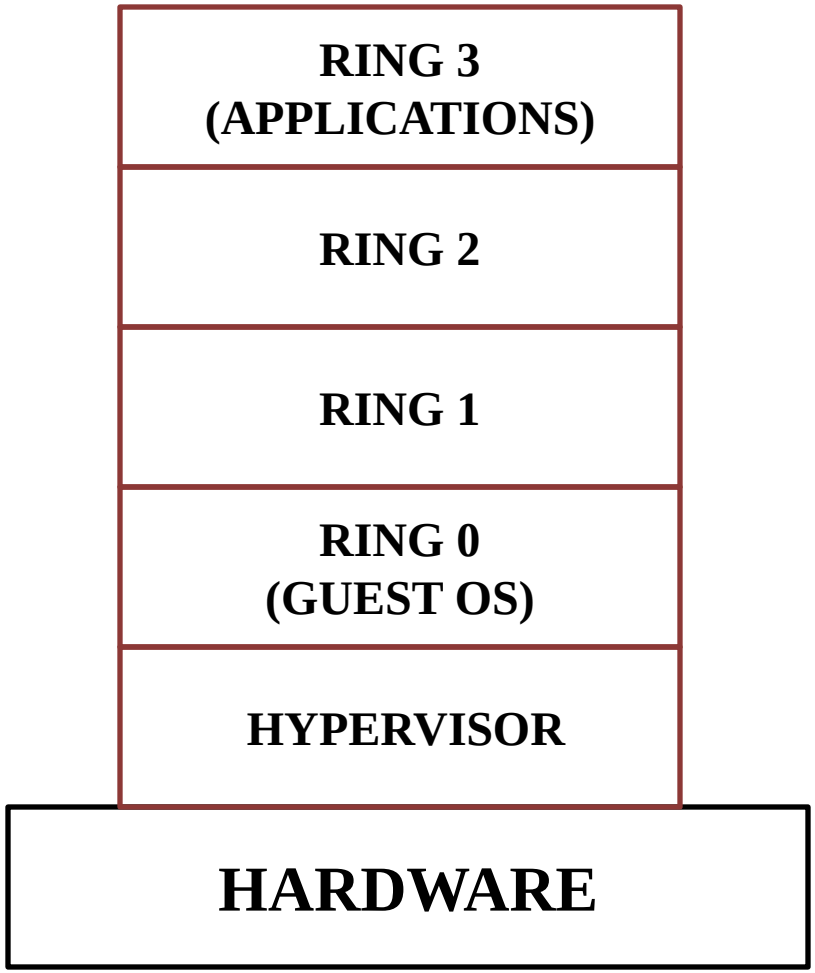
- Hypervisor operates at Ring 0
- Hypervisor scans the request stream
  - Captures and translates privileged instructions
  - Guest OS thinks it is directly working with hardware
- Performance is impacted due to binary translation

Para Virtualization



- Hypervisor resides in a privileged layer beneath the guest OS
- Guest OS is modified, so it doesn't execute privileged instructions
  - It executes hypercalls to the hypervisor
- Better performance
  - Limited use due to the need to modify OS

Hardware Assisted Virtualization



- Hardware allows hypervisor to reside in a privileged ring
- Privileged and sensitive calls are set to automatically trap to the hypervisor
- Can use unmodified OS + better performance
- Requires hardware support

Types of Virtualization

Type of Virtualization	Requires Hardware Support?	Requires Guest OS Modification?
Full Virtualization	No	No
Para virtualization	No	Yes
Hardware Assisted Virtualization	Yes	No