Agenda

- About SDLC
- Addressing Delivery Challenges
- Agile and DevOps – How is it connected?
- What CI and CD Pipeline
- About DevOps and DevOps Tools

Let's talk projects:

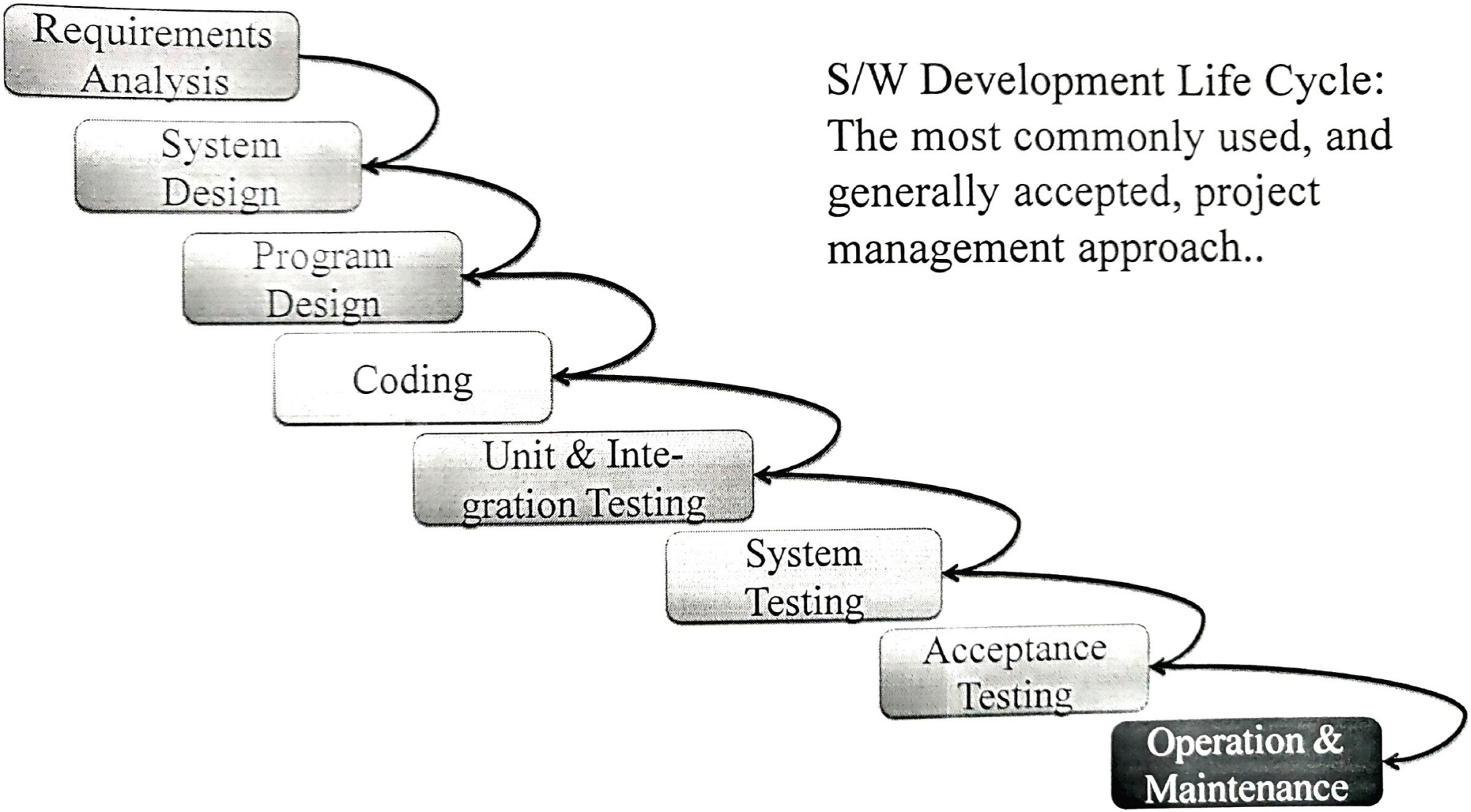
--Almost all work is done as a "project"

--All projects have a plan, execute, inspect, accept model

--Project Management has become a practice and there are many methods and tools for project management.

--All are about "delivered on time and on budget".

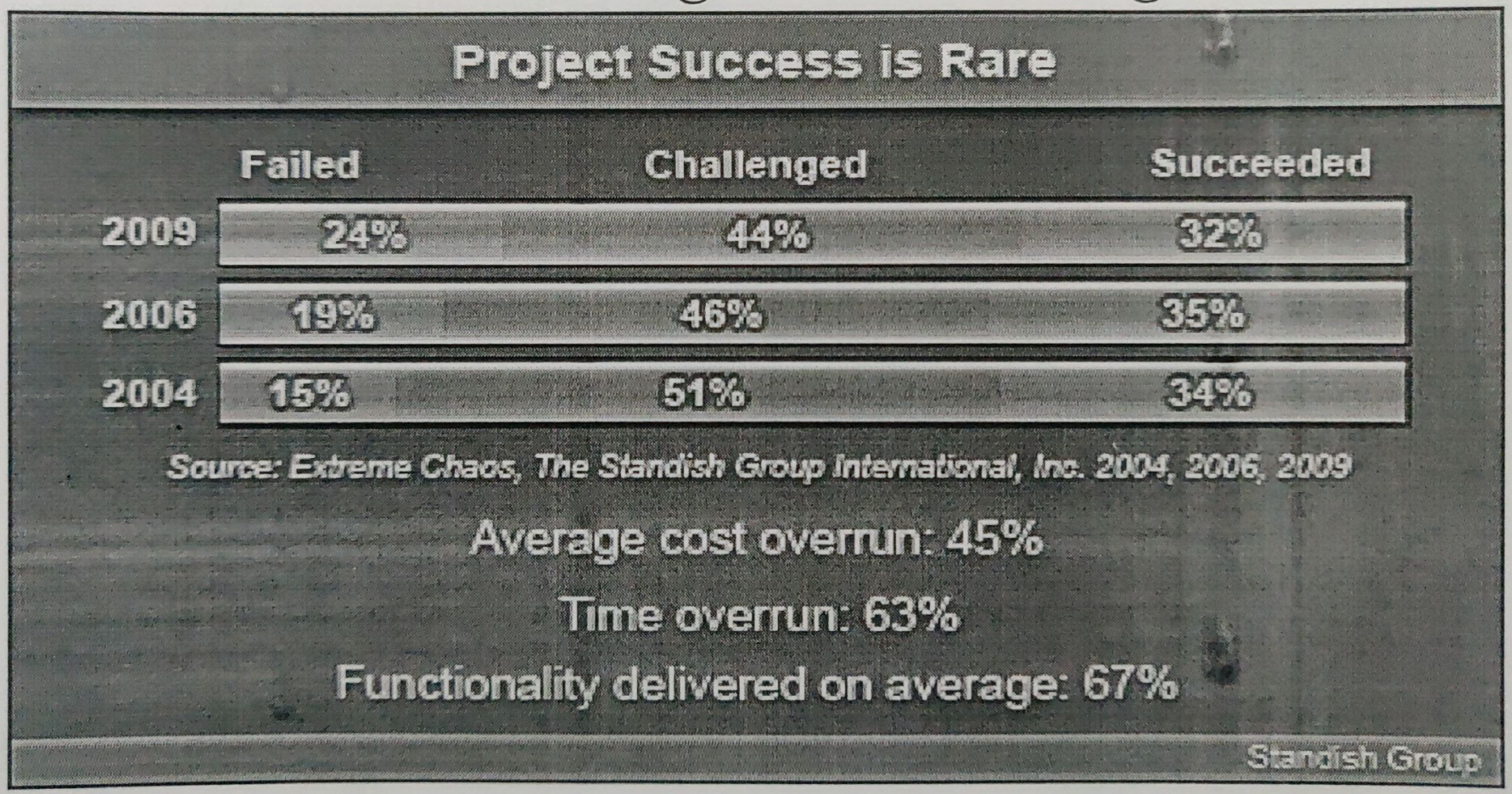
SDLC – What is it?



S/W Development Life Cycle: The most commonly used, and generally accepted, project management approach..

After 30 years of SDLC…

What we're doing is not working!

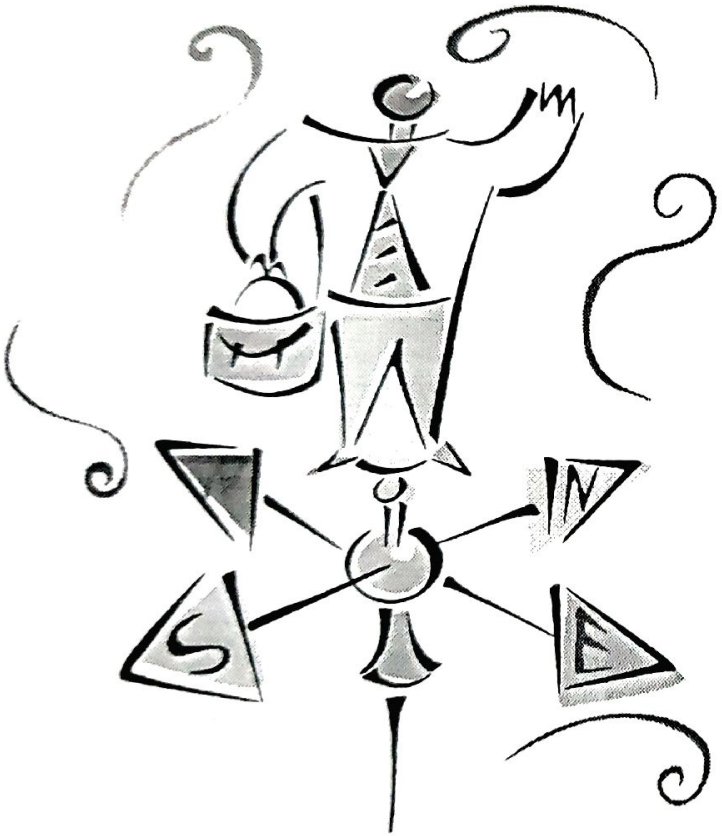


So, what happened?

- Did the customer get what was wanted? Were the actual needs met?

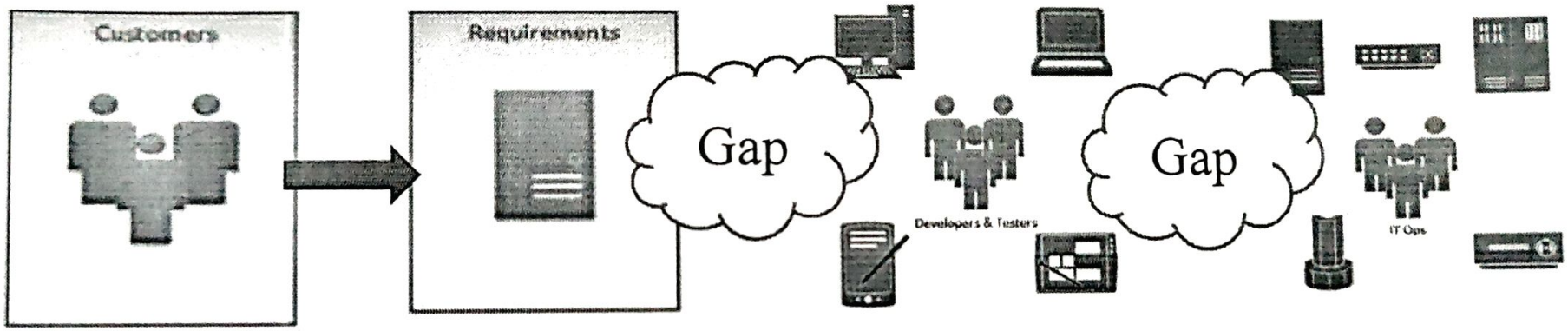
- Why?

It's All About…



Change!

Addressing Delivery Challenges



Customers Requirements Dev & Testers IT Ops

Agile - The Key Components

- ***User Stories*** – Simple statements of requirements written from the "customer's" point of view.

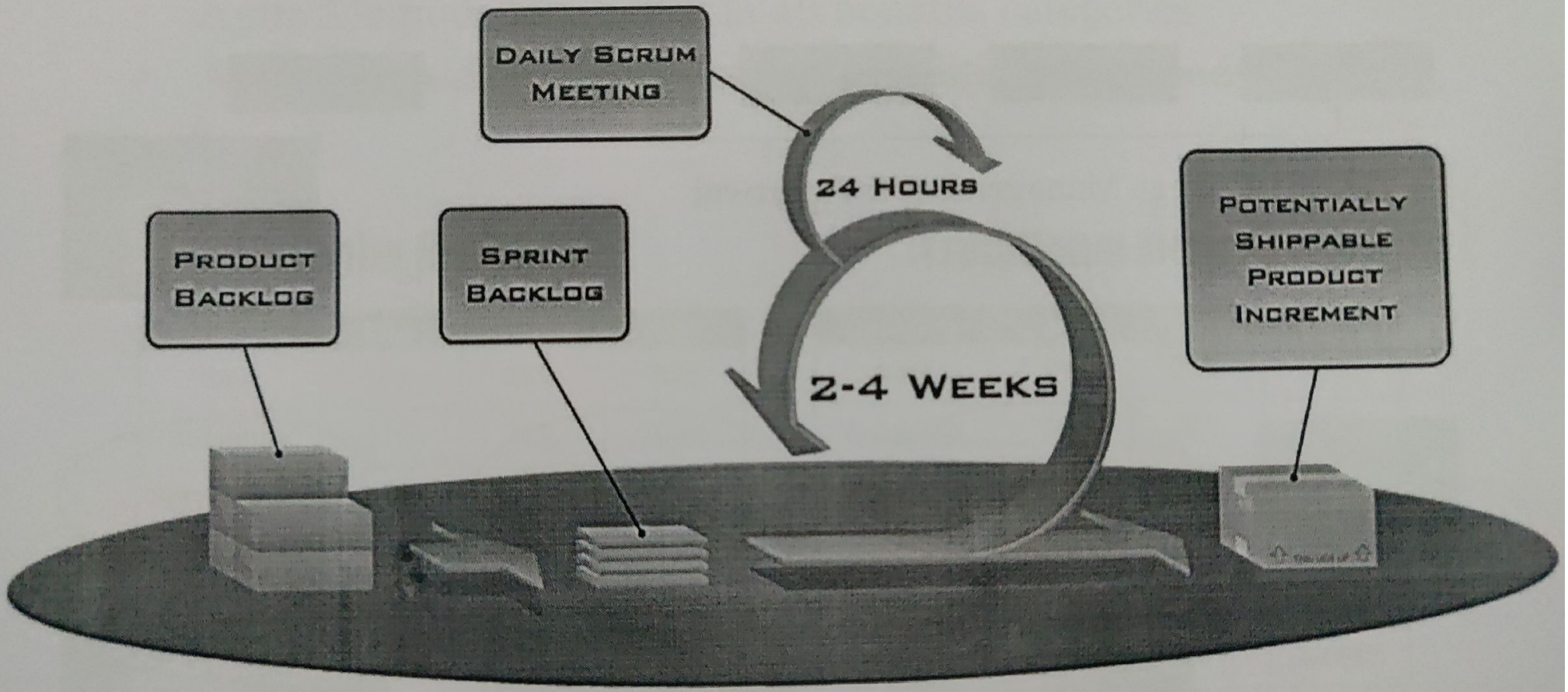
- ***Product Backlog*** – Collection of user stories that need to be addressed to consider the effort (Product) complete.

- ***Sprint (aka Iteration)*** – A fixed length work. An Agile project is a sequence of sprints.

- ***Sprint Planning Session*** – A team meeting in which the product owner reviews and explains each backlog items and it's priority.

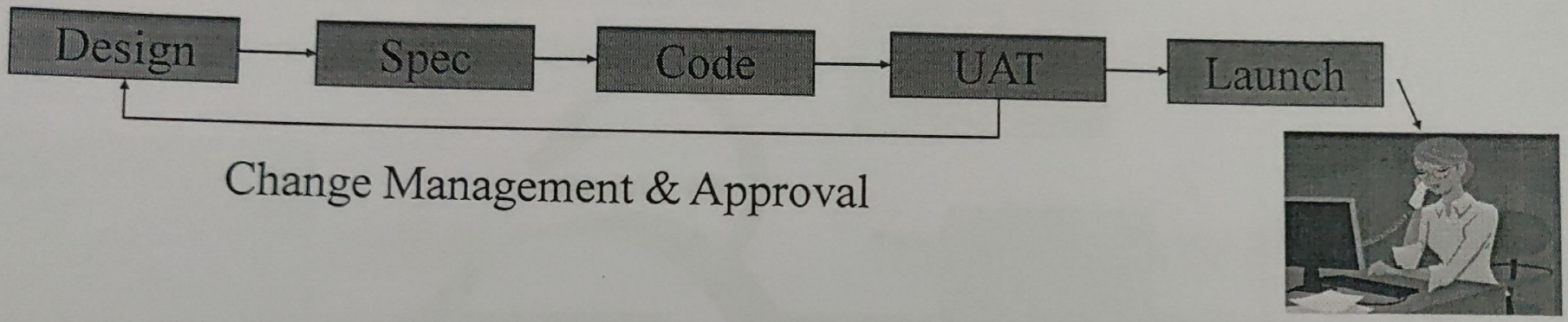
- ***Sprint Review Session*** – At the closure of each sprint, work completed is presented and reviewed, lessons learned, the overall sprint is evaluated

The Agile Model

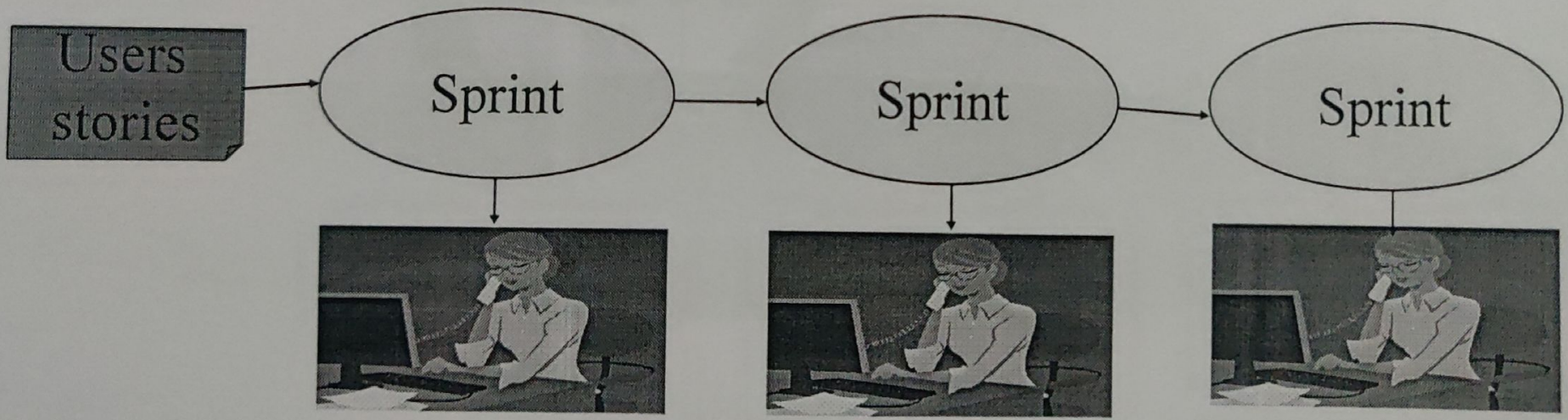


Faster – better - cheaper

Waterfall Approach



Agile Approach



Why Agile?

- Agile Methods fare better in all benefits categories
- Agile Methods 459% better than Traditional Methods

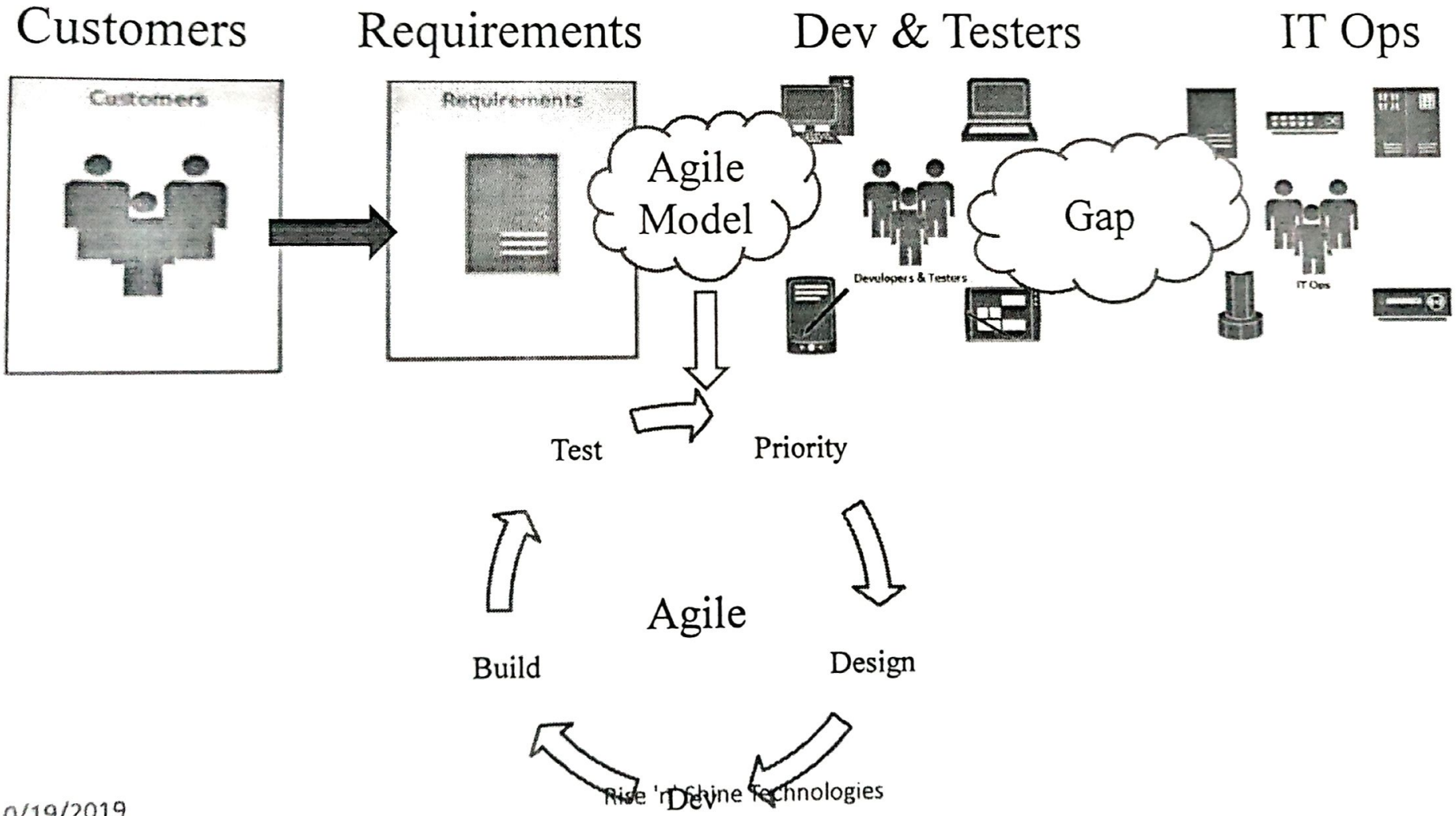
Agile Methods

Category	Low	Median	High
Cost	10%	26%	70%
Schedule	11%	71%	700%
Productivity	14%	122%	712%
Quality	10%	70%	1,000%
Satisfaction	70%	70%	70%
ROI	**240%**	**2,633%**	**8,852%**

Traditional Methods

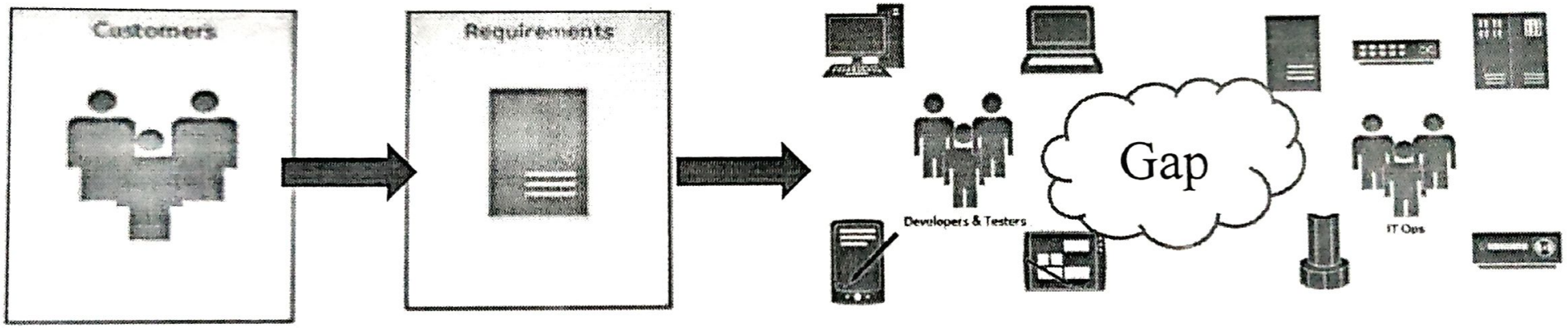
Category	Low	Median	High
Cost	3%	20%	87%
Schedule	2%	37%	90%
Productivity	9%	62%	255%
Quality	7%	50%	132%
Satisfaction	-4%	14%	55%
ROI	**200%**	**470%**	**2,770%**

Agile Model


Customers
Requirements
Dev & Testers
IT Ops
Customers
Requirements
Agile Model
Gap
Devulopers & Testers
IT Ops
Test
Priority
Agile
Build
Design
Dev

Why DevOps?

Why DevOps? Delivery Challenges



Customers Requirements Dev & Testers IT Ops

Why Gaps? Developers View

- Objectives/Activities:
  - Implementing Feature/Enhancements
  - Integration of the Code with Other Developers
  - Performing Build Activities (Converting code to package formats)
  - Deploying that package to Dev Environment
  - Evaluating all the Unit Test Cases
- Dev systems may not be same as production system
- Developers will have faster turn around time w.r.t features/Enhancements
- Not much concerned about the infrastructural as well as deployment impact because of the code changes

Why Gaps? Operations View

- Objectives:
  - Setting up High Availability Infrastructures
  - Load Balancing
  - Fail Over
  - Scaling Features
  - Deploying Changes to the Remaining environments other than Dev env
- Rewarded mainly for uptime
- Lesser turn around time w.r.t feature deployment and testing due to large number of dev builds coming their way

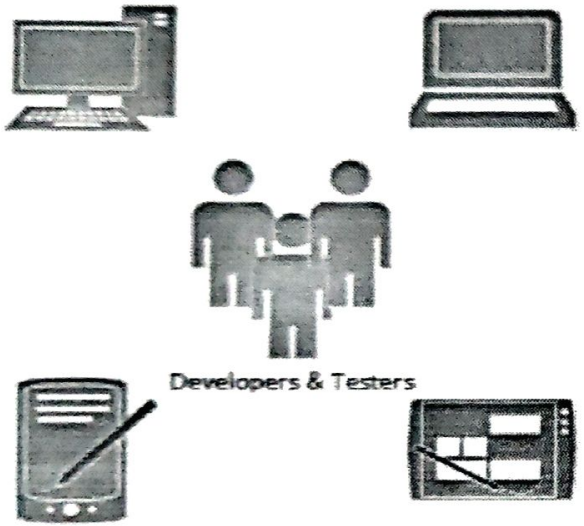
Principles of DevOps

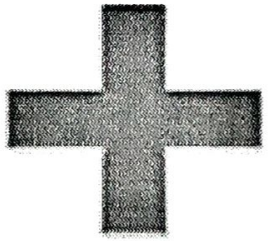
- Develop and test in an environment similar to production
  - To overcome the Infrastructure Issues
- Deploy App builds and Infra changes frequently
  - Continuous Deployment to all the environments
- Validate operation quality continuously
  - Continuous Testing and Continuous Monitoring

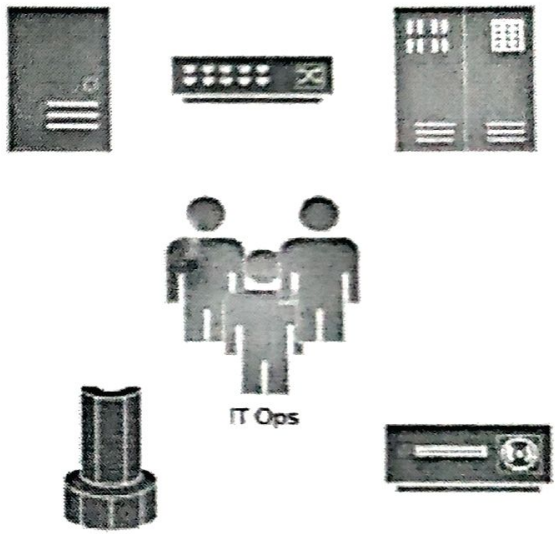
How?

- to setup similar environments?
- to deploy/apply the change to the number of systems and no of env?

What is DevOps?





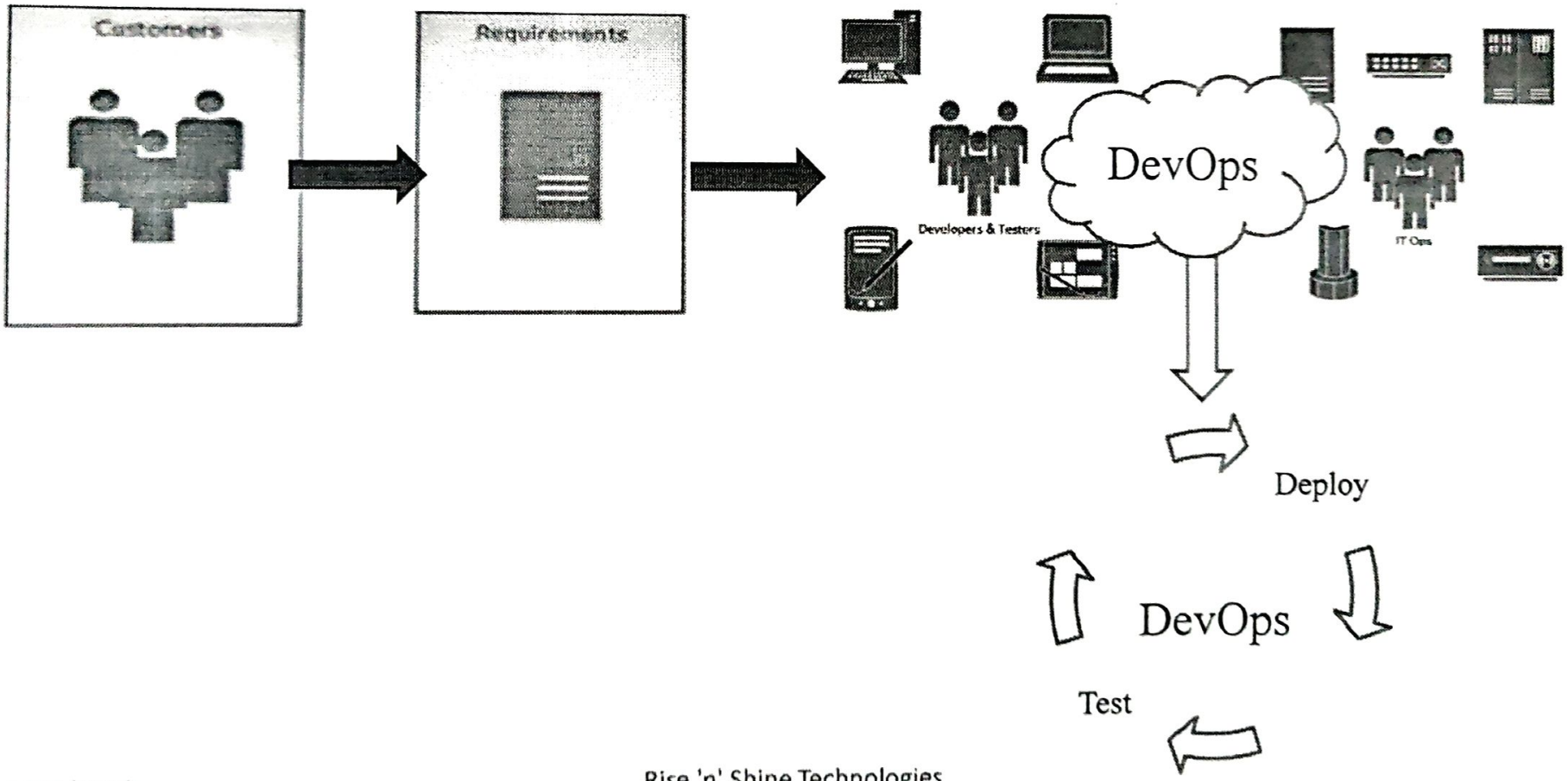


Developers & Testers DevOps IT Operations

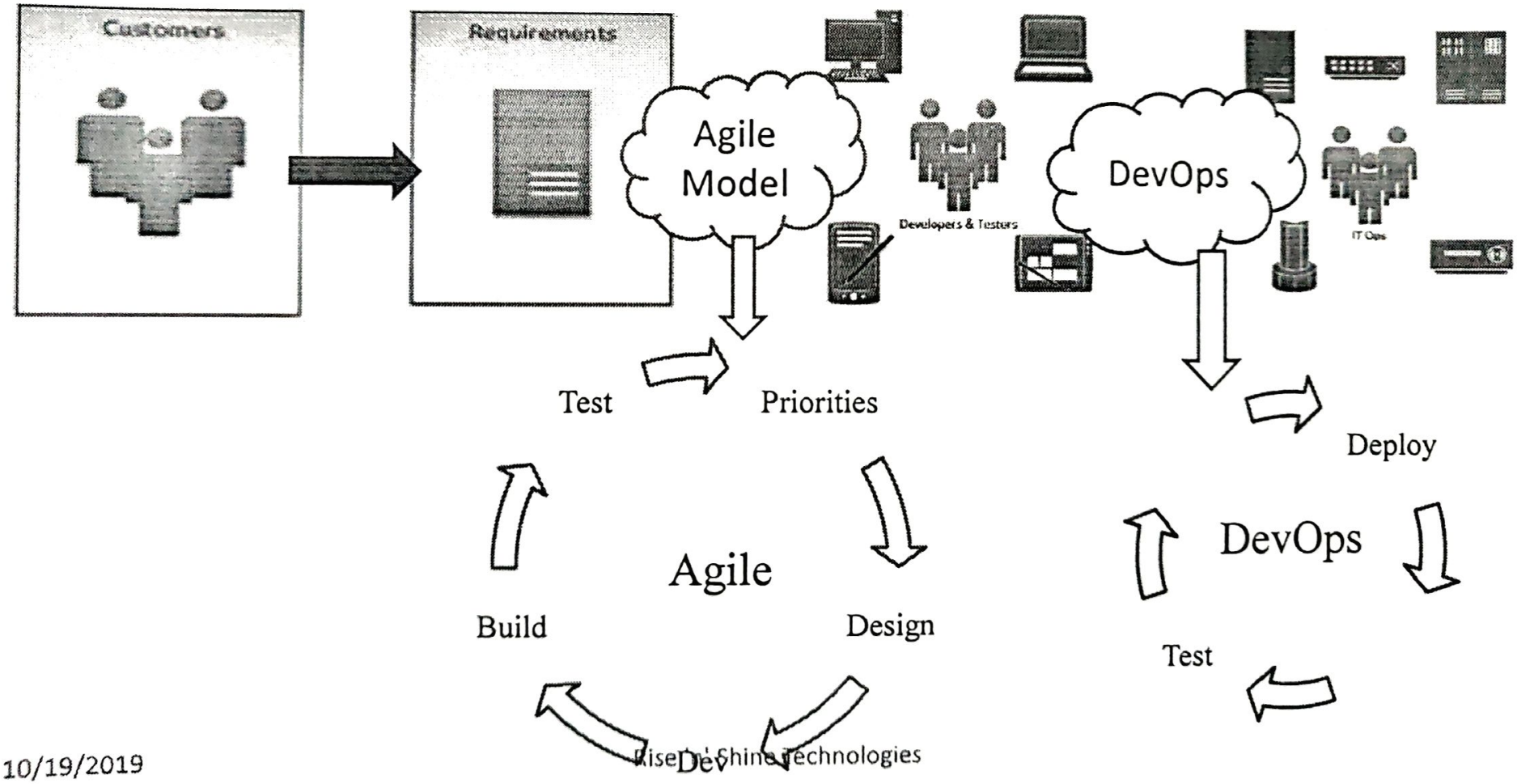
What is DevOps?

- DevOps (a combination of development and operations) is a software development method that stresses **communication, collaboration and integration** between software developers and information technology(IT) professionals.

DevOps

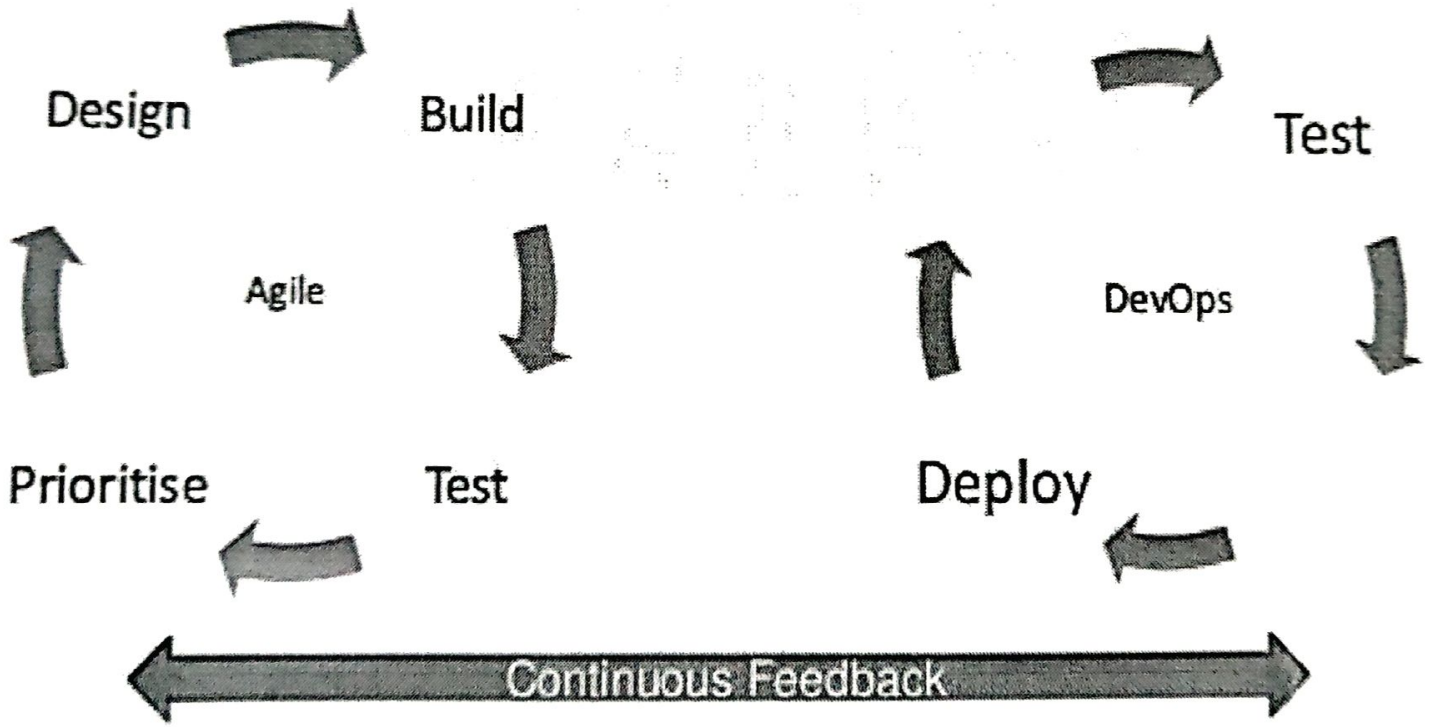

Customers
Requirements
Developers & Testers
DevOps
IT Ops
Deploy
DevOps
Test

Agile & DevOps



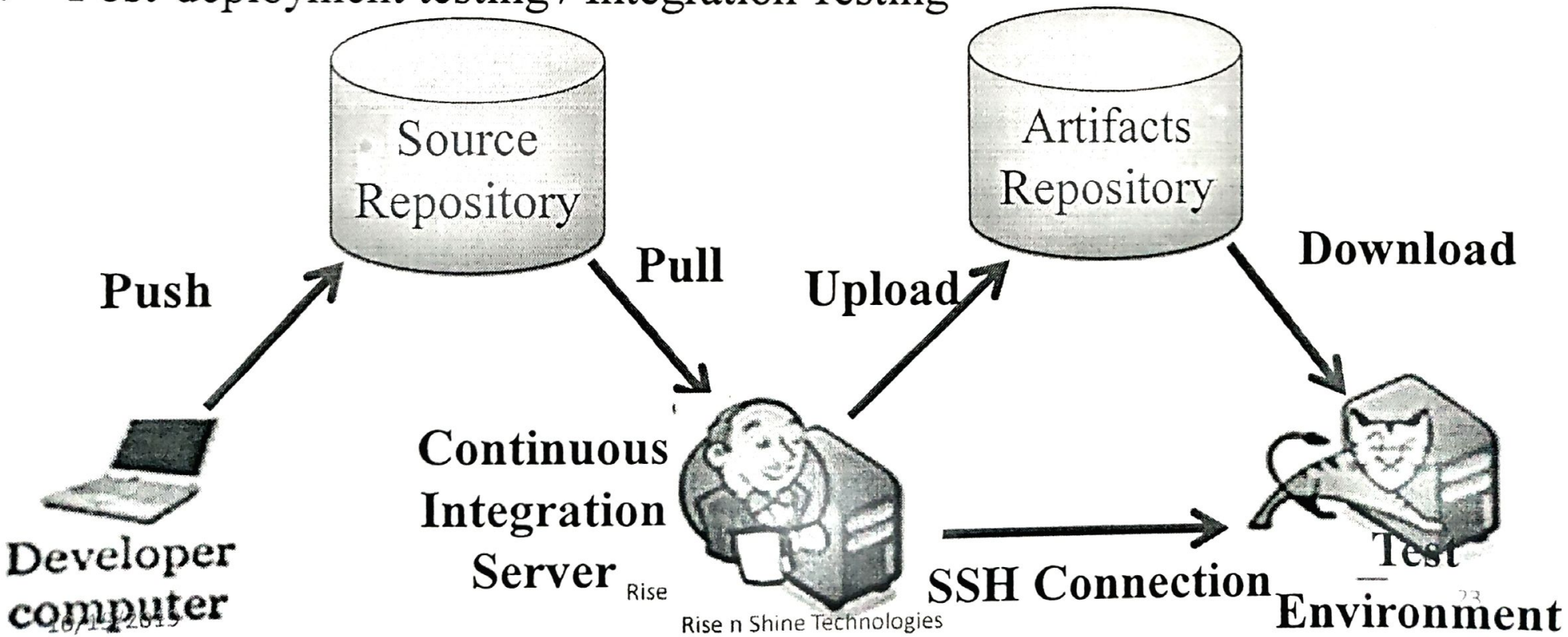
Agile + DevOps

-- Continuous Integration as a Continuous Delivery/Deployment

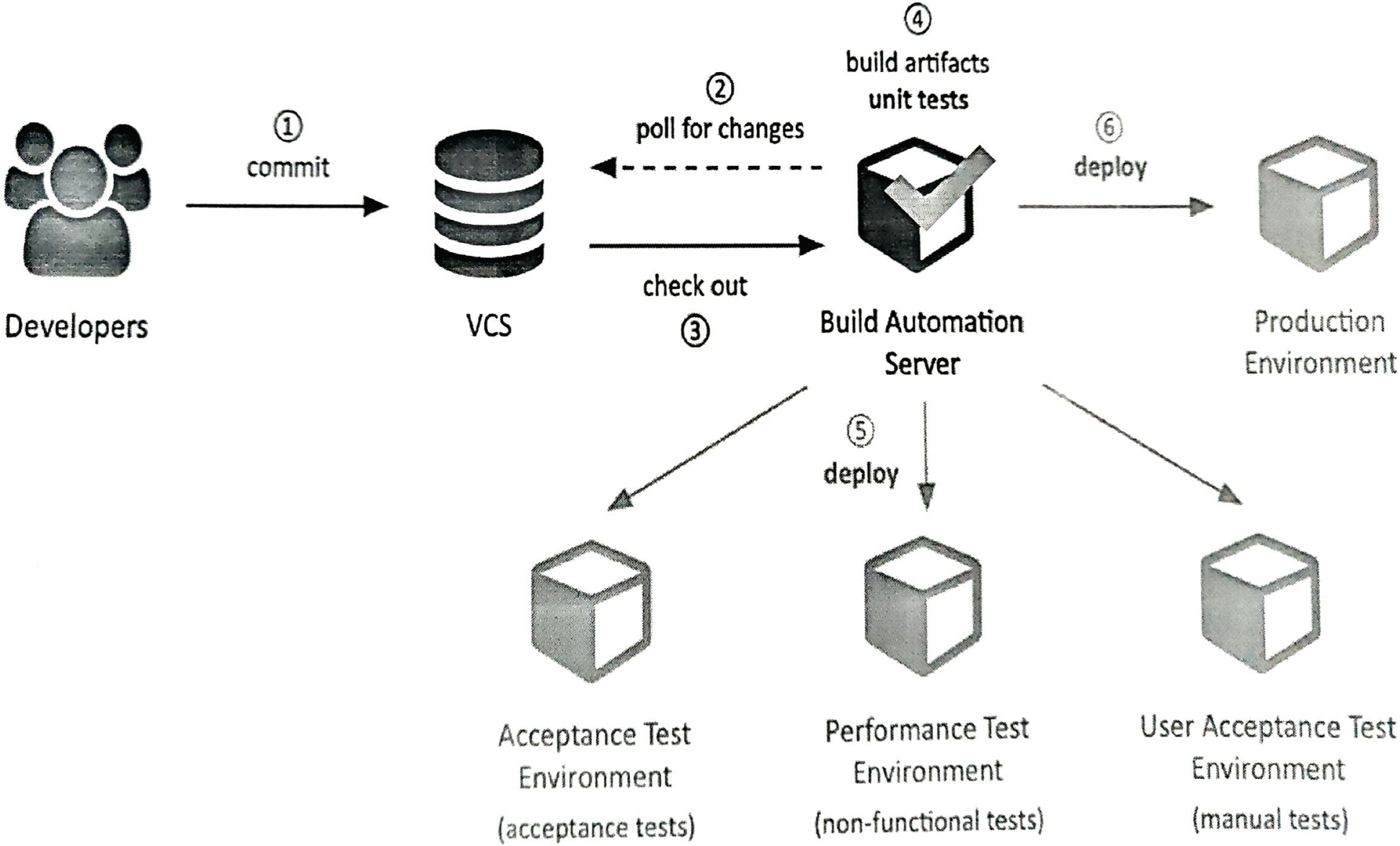


Continuous Integration:

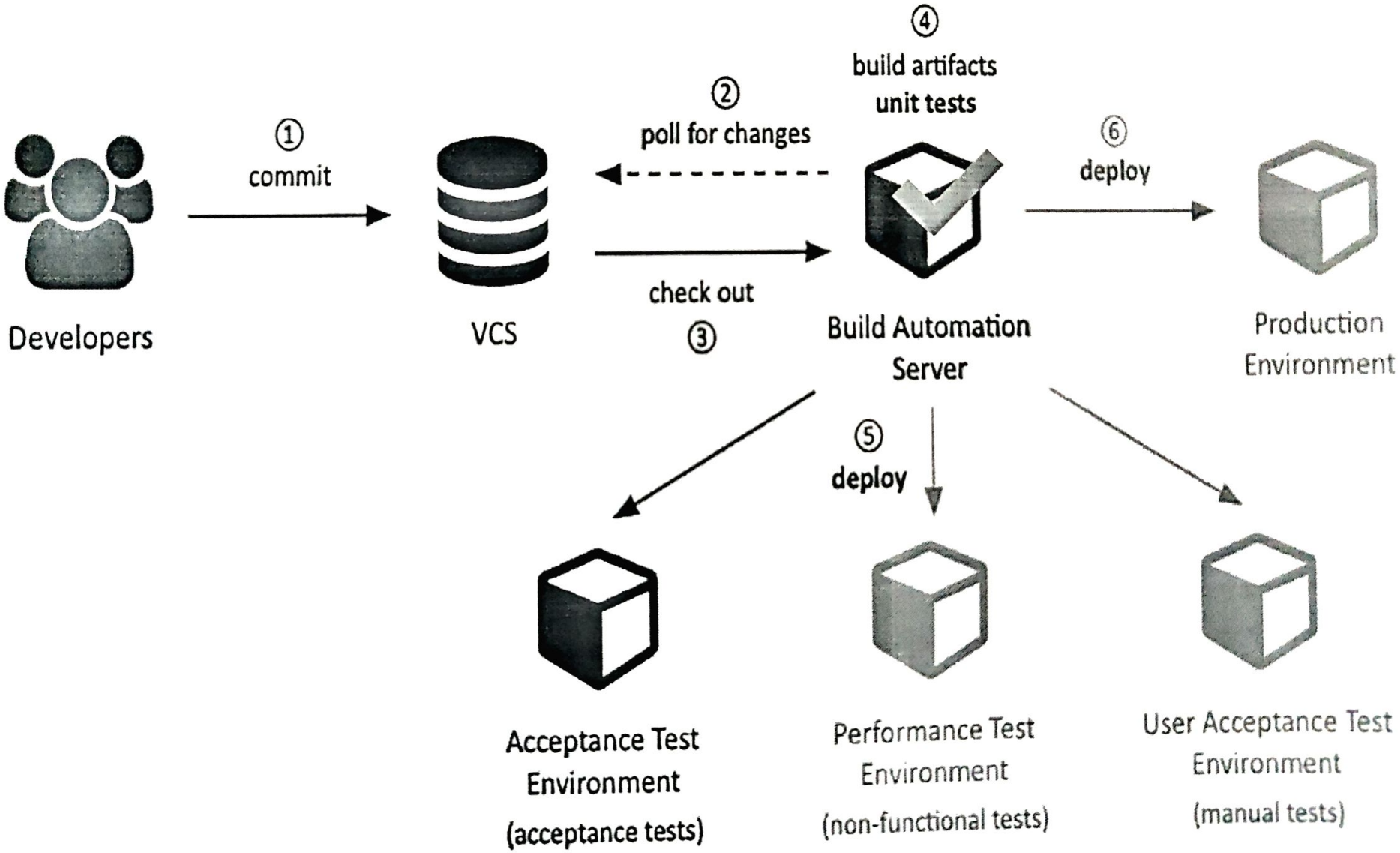
1. Pushing the code to the Source Code repository
2. Pull the code to the Build/CI Server and Start the Pipeline Process
3. Static analysis / Code verification
4. Pre-deployment testing / Unit Testing and Build Activity
5. Upload to the Artifacts Repository
6. Download and Deploy to the testing environments
7. Post-deployment testing / Integration Testing



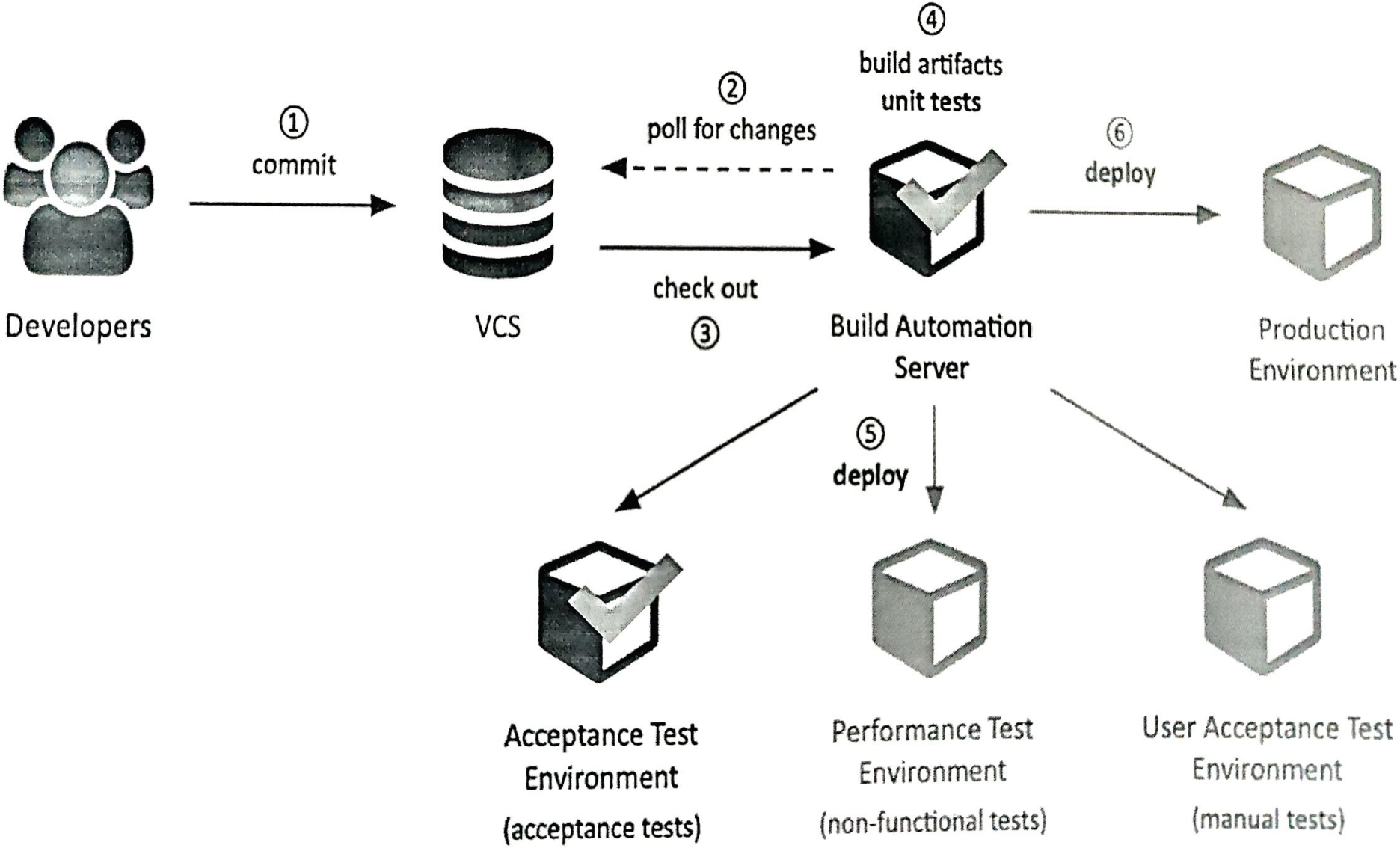
Continuous Delivery Deployment Pipeline



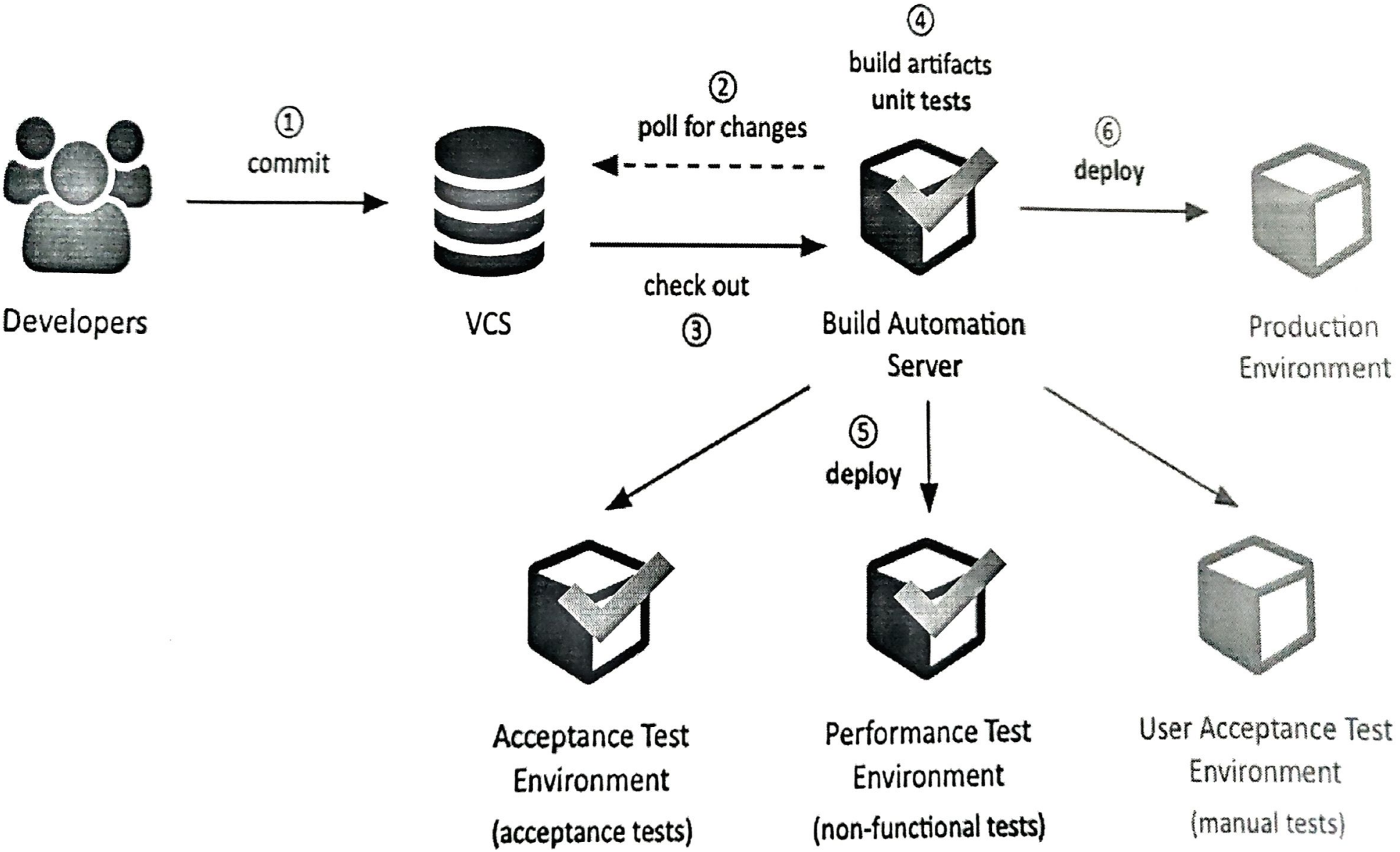
Continuous Delivery Deployment Pipeline



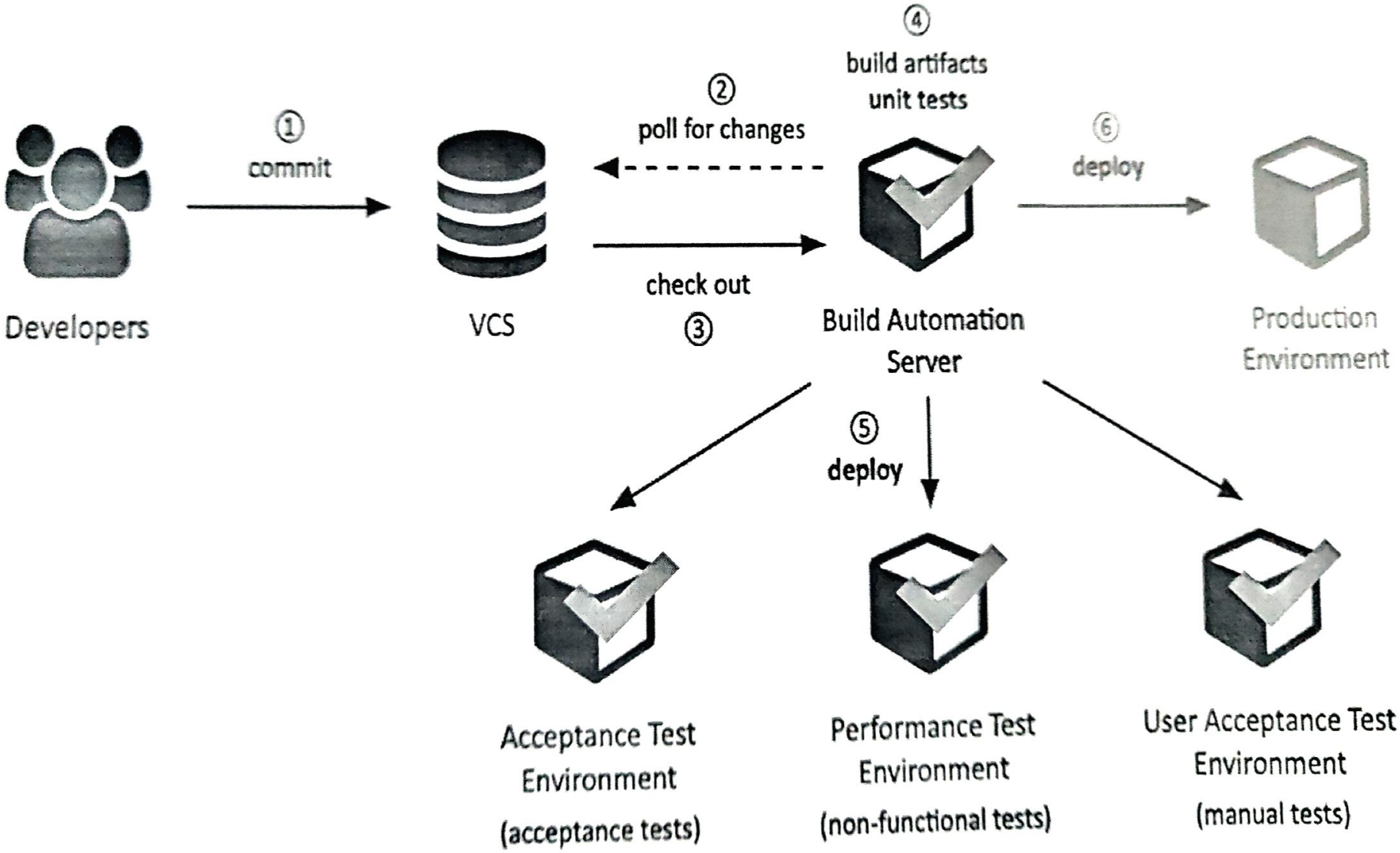
Continuous Delivery Deployment Pipeline



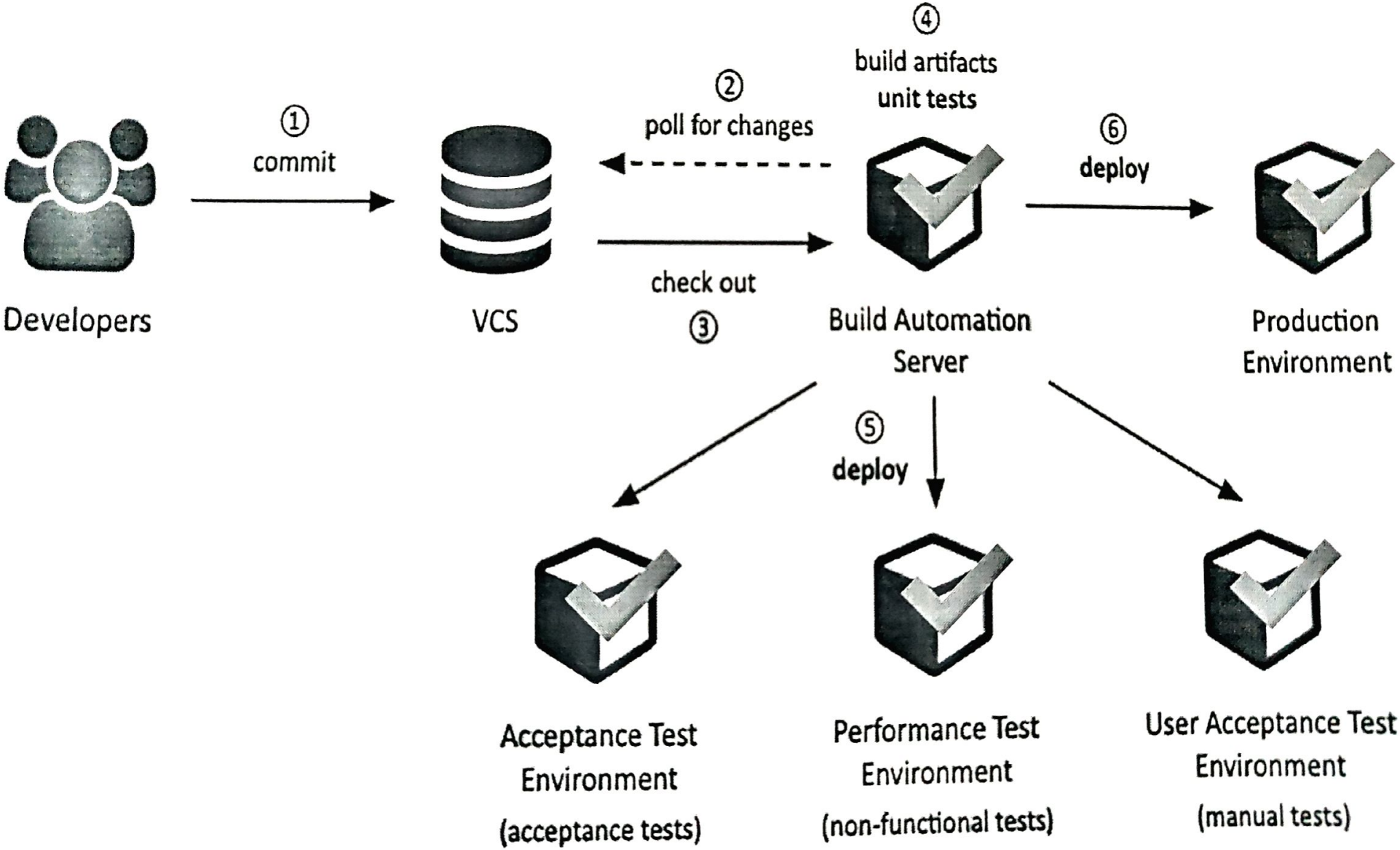
Continuous Delivery Deployment Pipeline



Continuous Delivery Deployment Pipeline

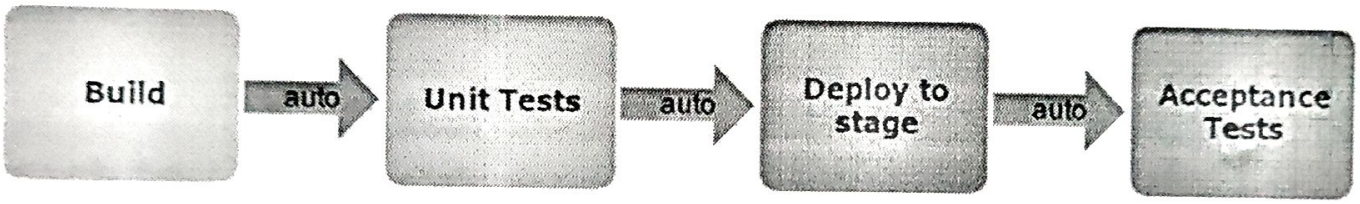


Continuous Delivery Deployment Pipeline

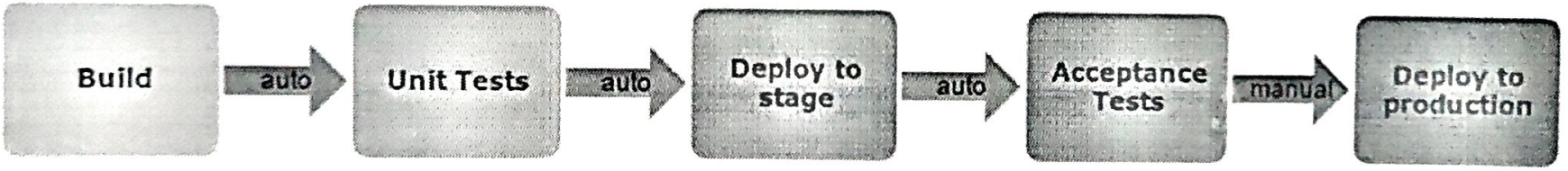


CI & CD

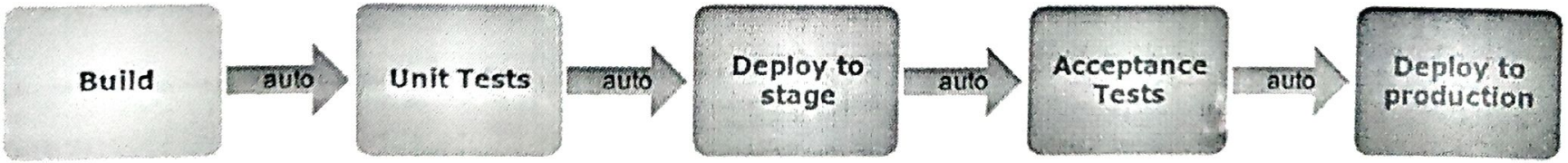
Continuous Integration



Continuous Delivery



Continuous Deployment



Combining the Power of Developers and Operations

- The operations complexity breaks down into a few main categories:
- Development Activities:
  - infrastructure automation
    - VMs, Storages, N/Ws….
  - configuration management
    - Platform Mgmt (Java, .Net, Python, PHP, DB…)
  - deployment automation
    - Application Changes
- Operational Activities:
  - log management
  - performance management
  - monitoring

Tools/Technologies

- Source Code Management –
  - SubVersion, CVS, Git, Mercurial etc...
- Build Tools:
  - Ant, Maven, Gradlle ... etc
- Code Review Tools:
  - Sonar, PMD, CheckStyle, Corbetura... etc
- Artifacts Repository:
  - Artifactory, Nexus, Archive...
- Continuous Integration Tools:
  - Jenkins, Hudson, Teamcity, Bamboo…

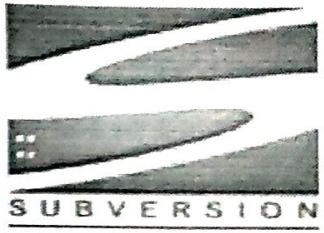
Tools/Technologies

- Deployment Platform:
  - Linux/Unix, Windows,...
- Configuration Mgmt Tools:
  - Ansible, Chef, Puppet, Salt...
- Application Servers:
  - Tomcat, WebLogic, JBoss, WebSphere...
- Test Automation Tools:
  - Selenium, Junit...
- Scripting:
  - Shell scripting, Perl Scripting, Python, Power Shell, Ruby etc...

Tools/Technologies

- Monitoring Tools:
  - Nagios
- Container Orchestration Engines (COE):
  - Docker, Kubernetes, Swarm…
- Cloud Infrastructures:
  - Amazon, Google Cloud, MS Azure
- Issue Tracking Tool
  - Jira
- Virtualization Software
  - Virtual Box, VMWare, Vagrant

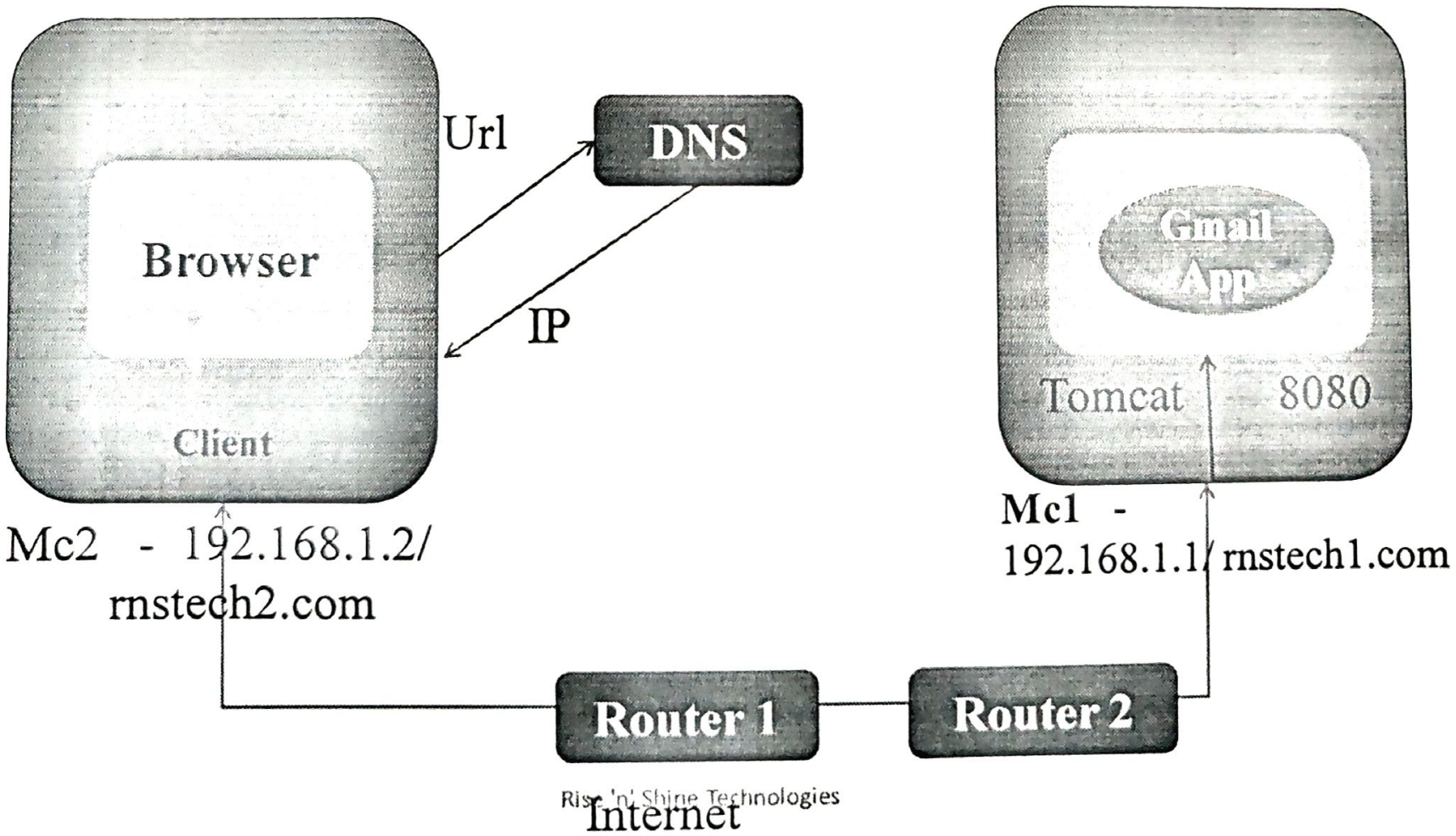
Common Elements of the Software SCM



Client Server Architecture

URL –

protocol://IP:Port/App_Name
http(s)://192.168.1.1:8080/Gmail
http://gmail.co.in (DNS register)



Client-Server Architecture

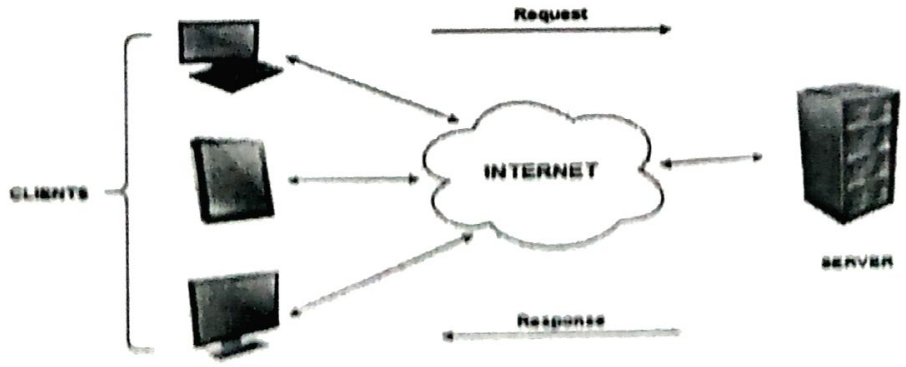
- The **clients requests** a resource and the **server respond** with that resource.

- A server may serve multiple clients at the same time while a client is in contact with only one server.

- There are two different structures :
  - Two - Tier Client/Server Structure
  - Three - Tier Client/Server Structure

Two – Tier Client/Server Structure

- The two tier architecture primarily has two parts, **a client tier** and a **server tier.**
- The client tier sends a **request** to the server tier and the server tier **responds** with the desired information.
- An example of a two tier client/server structure is a **web server**. It returns the required web pages to the clients that requested them.



Rise 'n' Shine Technologies

Advantages of Two - Tier Client/Server Structure

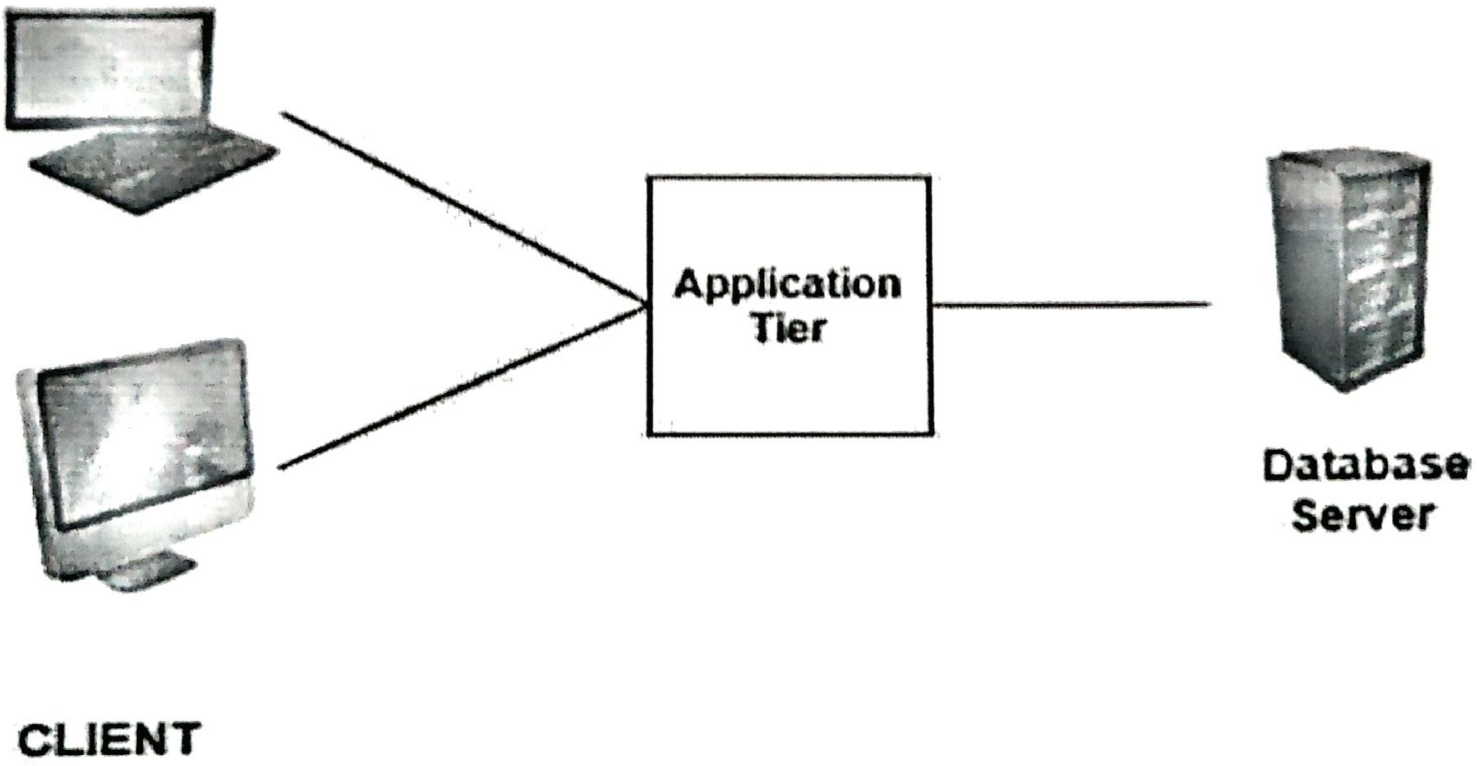
Advantages:

- This structure is quite easy to maintain and modify.
- The communication between the client and server in the form of request response messages is quite fast.

Disadvantages:

- If the client nodes are increased beyond capacity in the structure, then the server is not able to handle the request overflow and performance of the system degrades.

Three - Tier Client/Server Structure



Three - Tier Client/Server Structure

- It has three layers namely **client**, **application** and **data layer**.

- **The client layer** is the one that requests the information. In this case it could be the GUI, web interface etc.

- **The application layer** acts as an interface between the client and data layer. It helps in communication and also provides security.

- **The data layer** is the one that actually contains the required data.

Advantages of Three - Tier Client/Server Structure

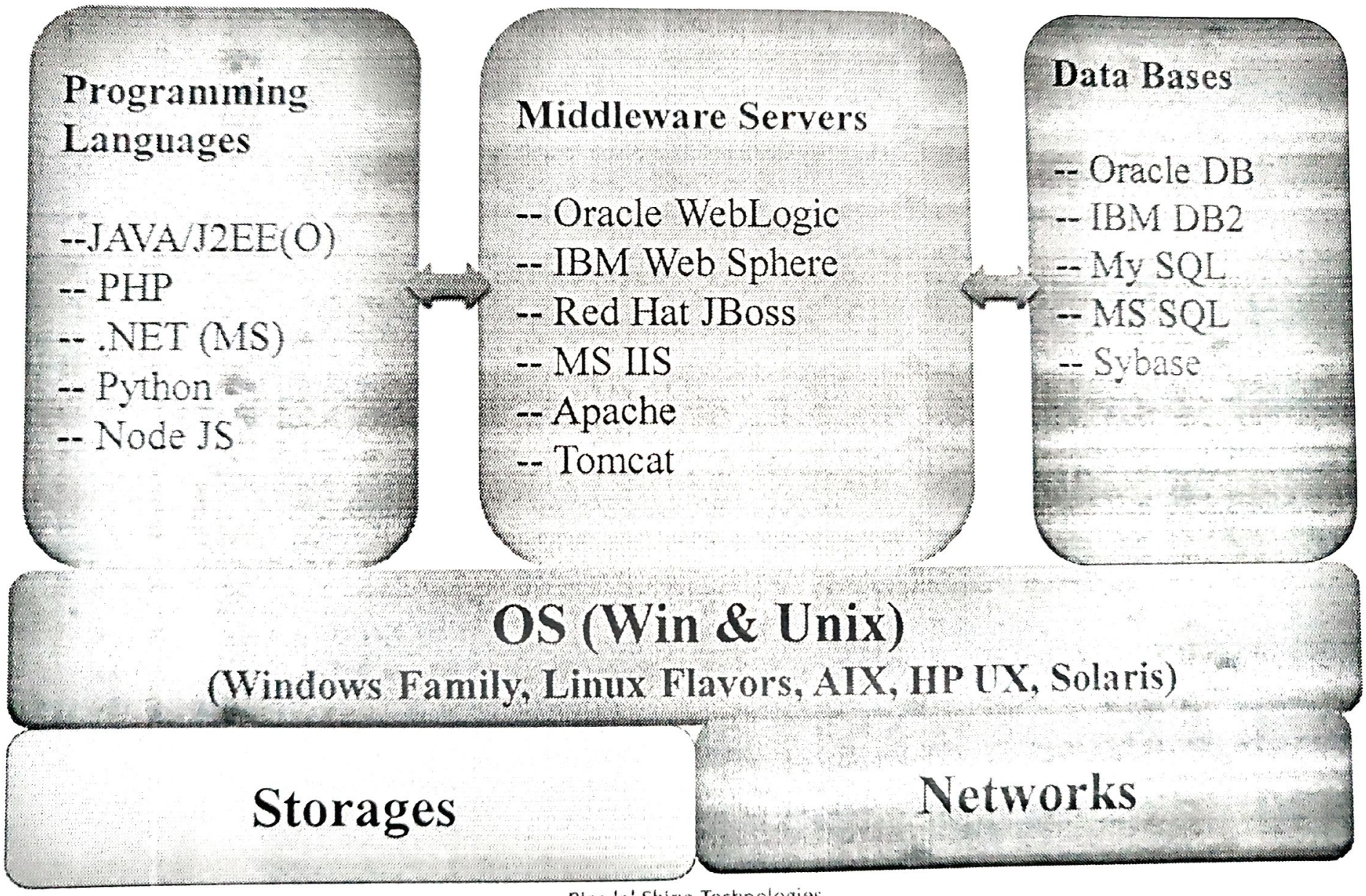
Advantages:

- The three tier structure provides much better service and fast performance.
- The structure can be scaled according to requirements without any problem.
- Data security is much improved in the three tier structure.

Disadvantages:

Three - tier client/server structure is quite complex due to advanced features.

Application Resources


Programming Languages
--JAVA/J2EE(O)
-- PHP
-- .NET (MS)
-- Python
-- Node JS
Middleware Servers
-- Oracle WebLogic
-- IBM Web Sphere
-- Red Hat JBoss
-- MS IIS
-- Apache
-- Tomcat
Data Bases
-- Oracle DB
-- IBM DB2
-- My SQL
-- MS SQL
-- Sybase
OS (Win & Unix)
(Windows Family, Linux Flavors, AIX, HP UX, Solaris)
Storages
Networks

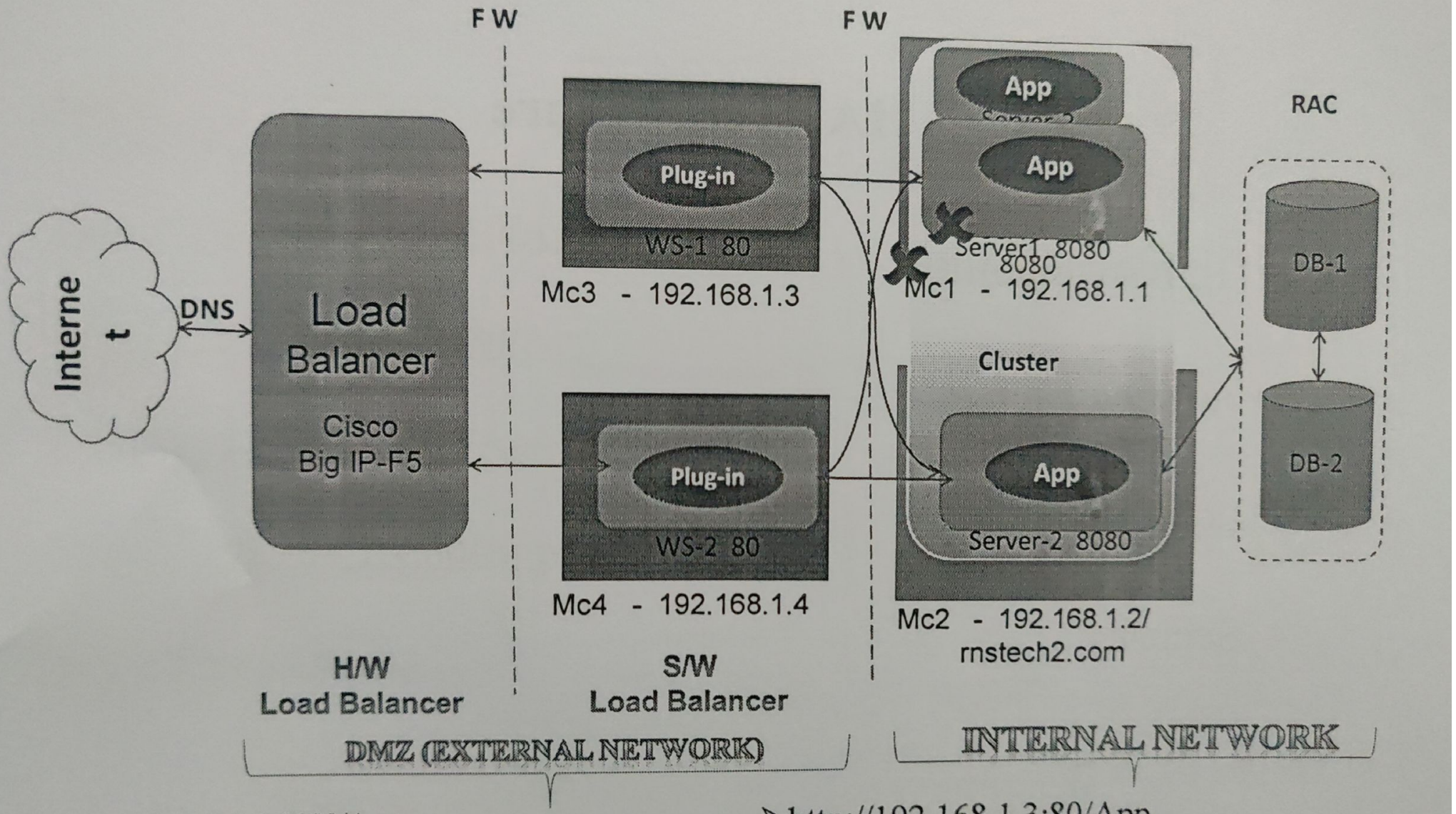
HA Architecture

1) High Availability

2) Fail Over

3) Load Balancing

4) Security



- http://192.168.1.1:8080/App
- http://192.168.1.2:8080/App

- http://192.168.1.3:80/App
- http://192.168.1.4:80/App
- http://gmail.co.in

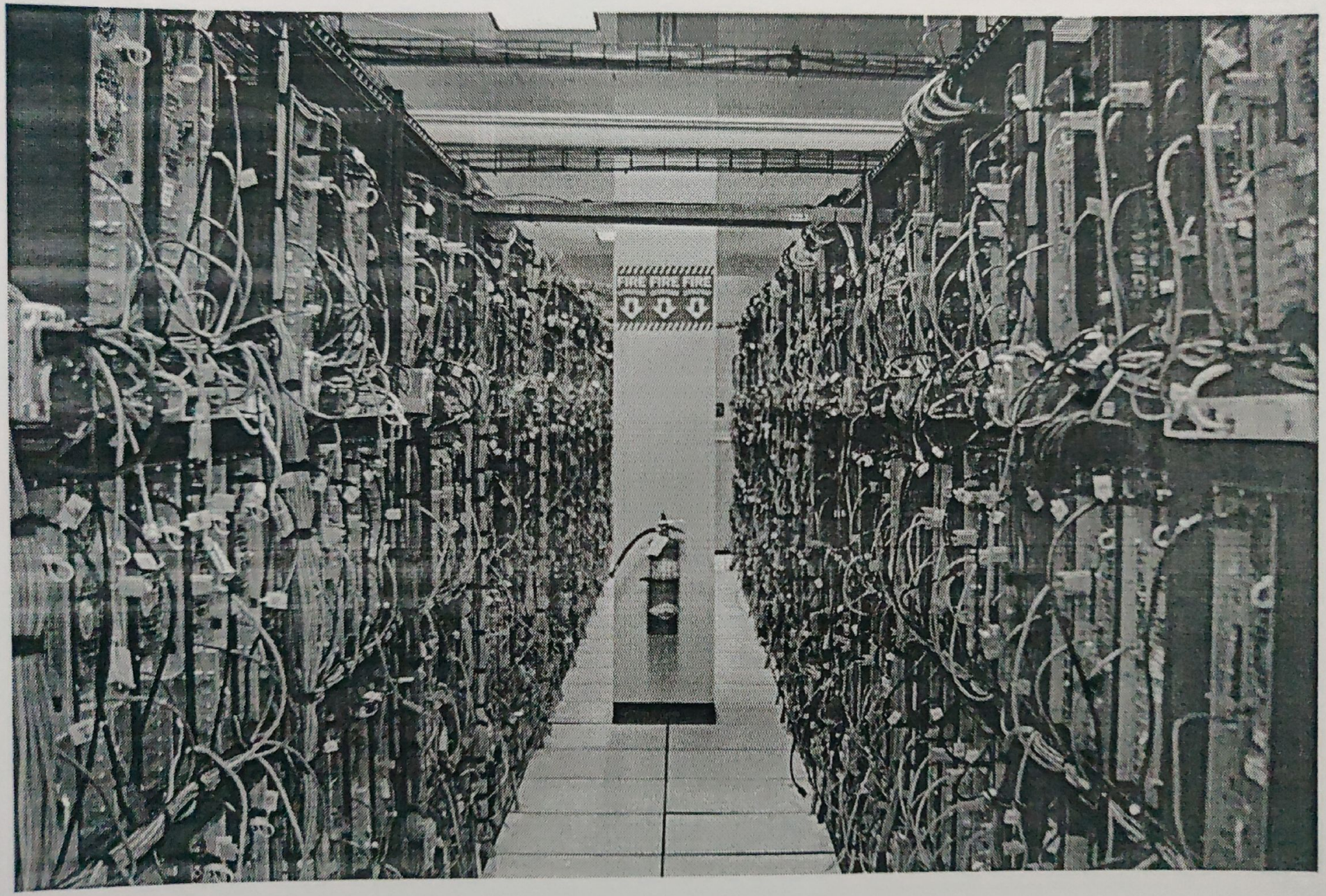
URLs:

- http://192.168.1.1:8080/App
- http://192.168.1.2:8080/App
- http://192.168.1.3:80/App
- http://192.168.1.4:80/App
- http://gmail.co.in


Cloud
Computing

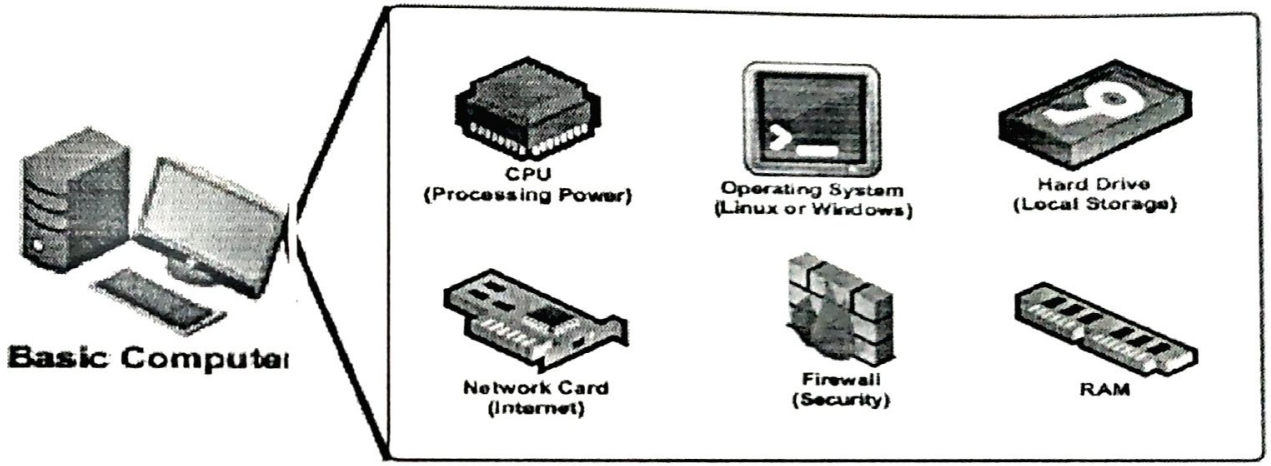
Common Use Cases for Infrastructure

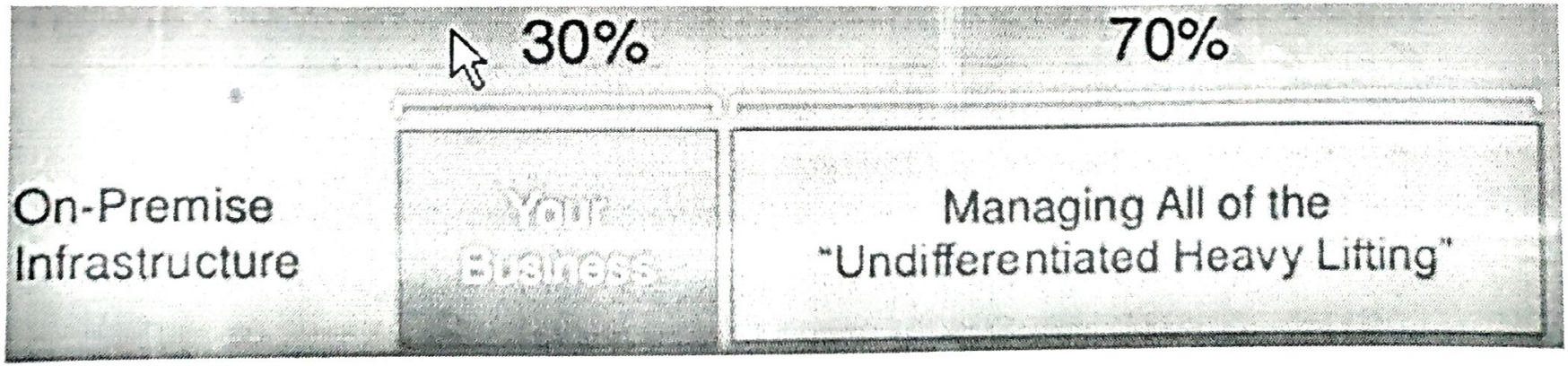
- Web site / Application hosting
- Mobile and Social Applications
- Internal IT application hosting
- Content delivery and media distribution
- High performance computing, batch data processing, and large scale analytics
- Storage, backup, and disaster recovery
- Development and test environments

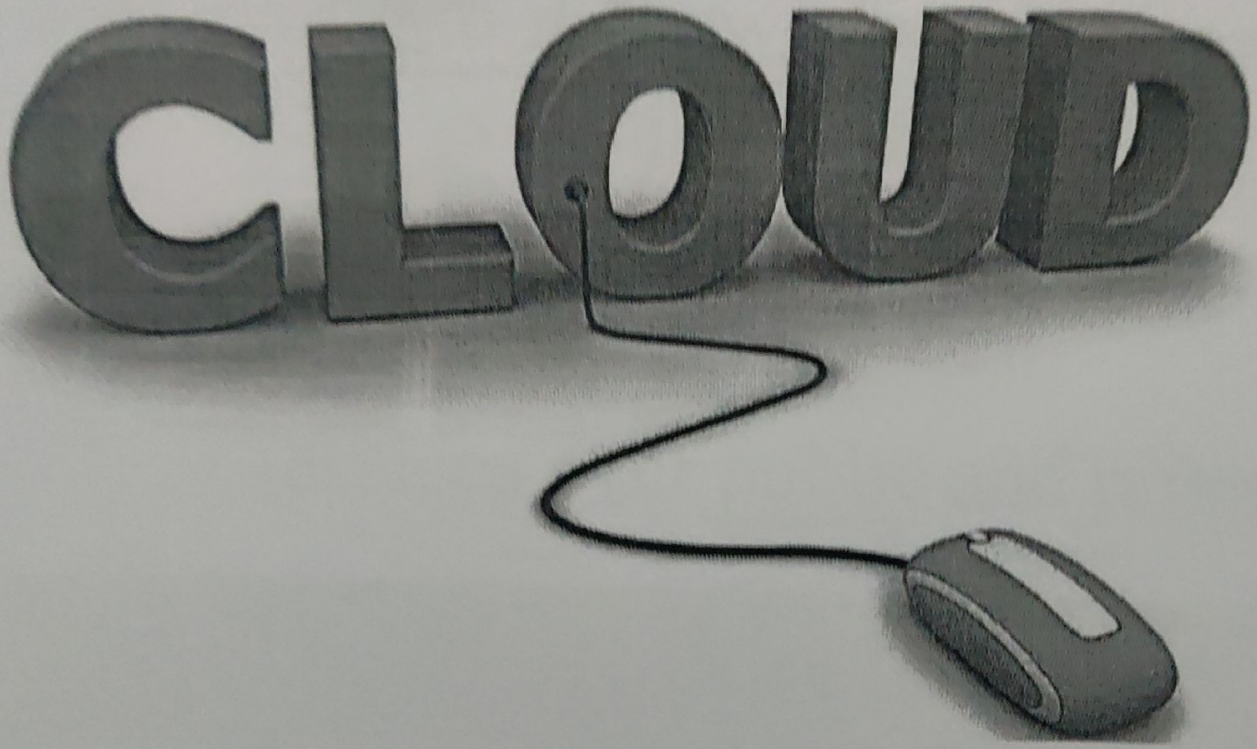


Rise 'n' Shine Technologies

- Basic Computer
- Networking
- Security
- Storage
- Data Base






CLOUD

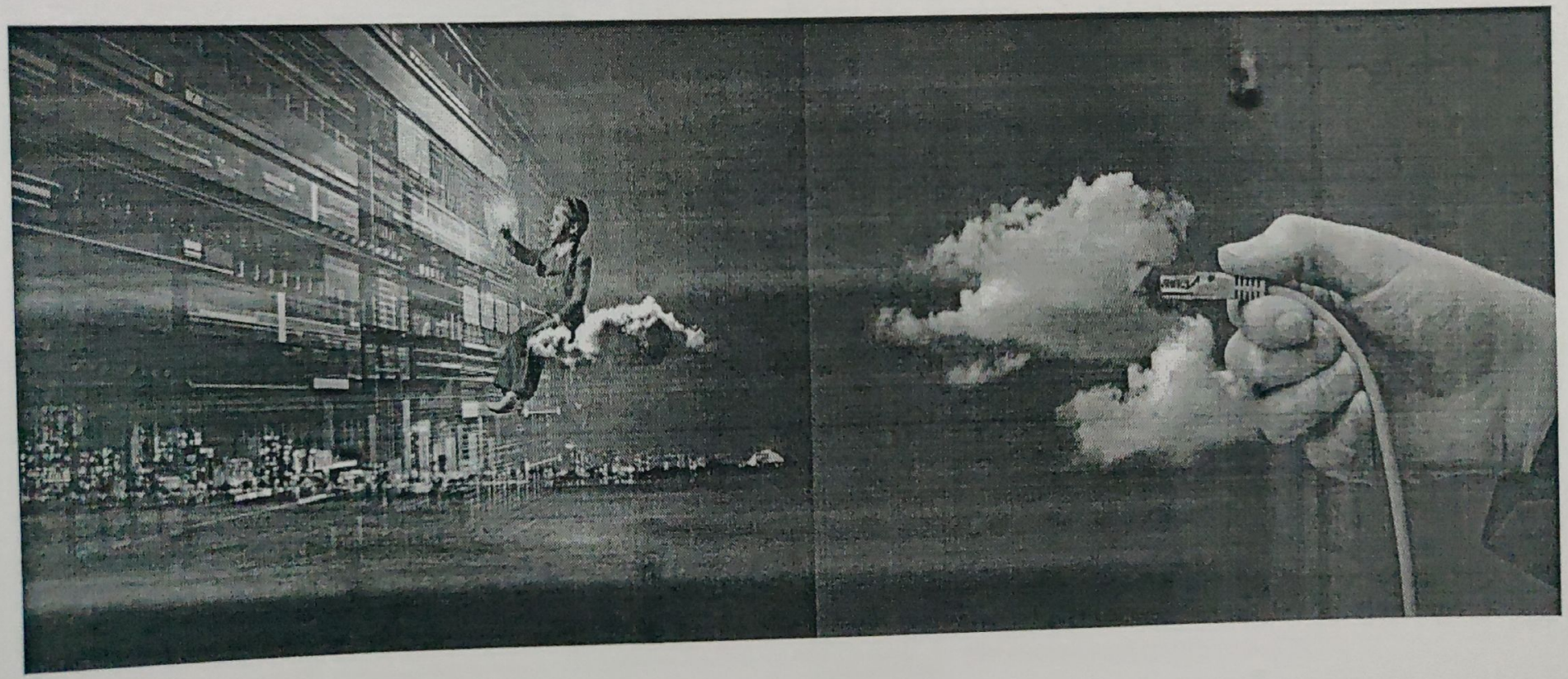
What is Cloud"?

The term **Cloud** refers to a Network or Internet. In other words, we can say that Cloud is something, which is present at remote location.

Cloud can provide services over public and private networks

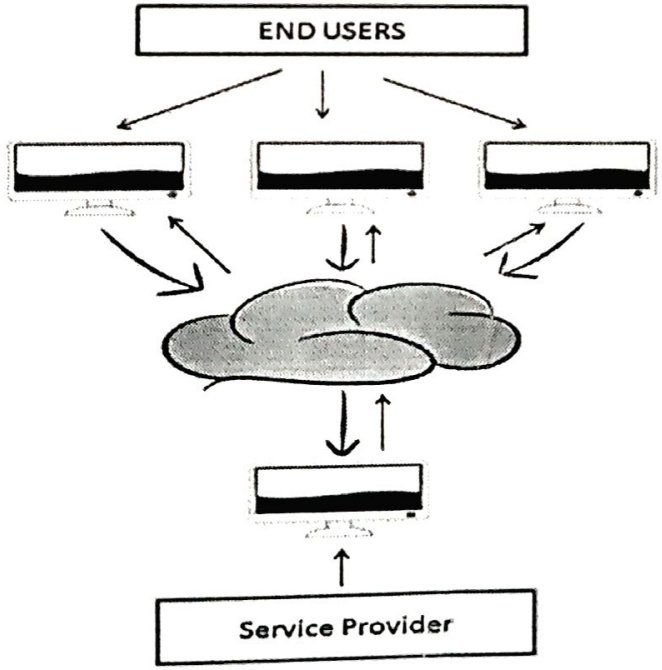
Cloud computing means storing and accessing data and programs over the Internet instead of your computer's hard drive.

Like this?



Why the Name "Cloud"?

- The term "Cloud" came from a network design that was used by network engineers to represent the location of various network devices and there inter-connection. The shape of this network design was like a cloud.



What is cloud computing?

- Cloud Computing can be defined as delivering computing power(CPU, RAM, Network Speeds, Storage OS software) a service over the internet rather than physically having the computing resources at the customer location.

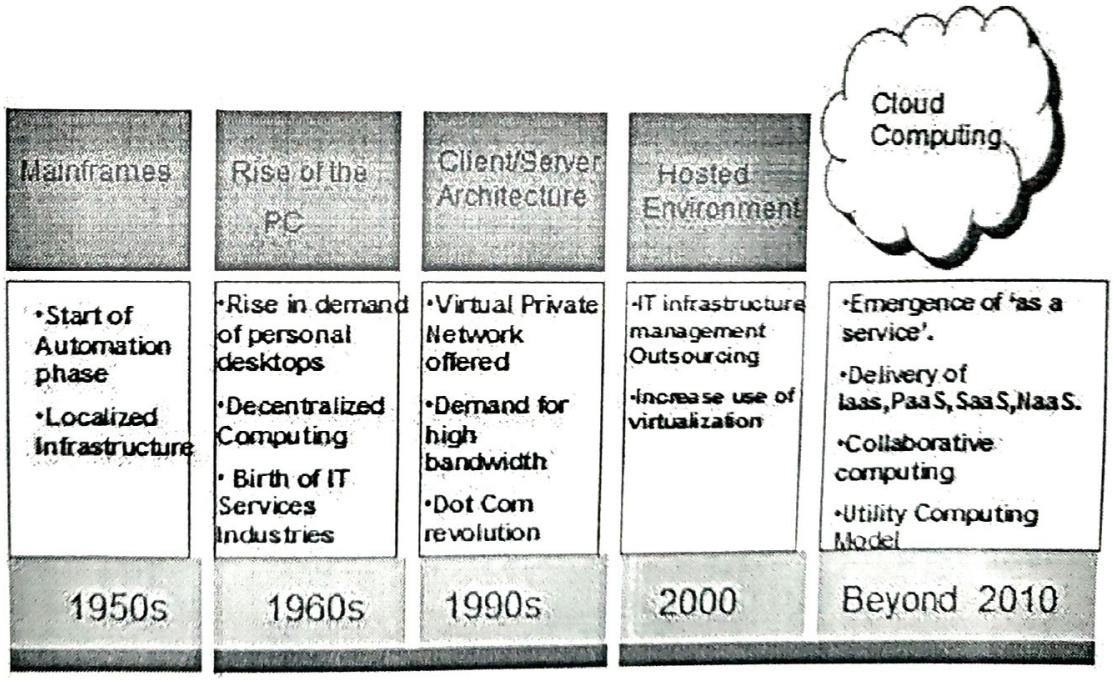
Or

- Cloud Computing refers to ***manipulating, configuring, and accessing the hardware and software resources remotely***. It offers online data storage, infrastructure, and application.

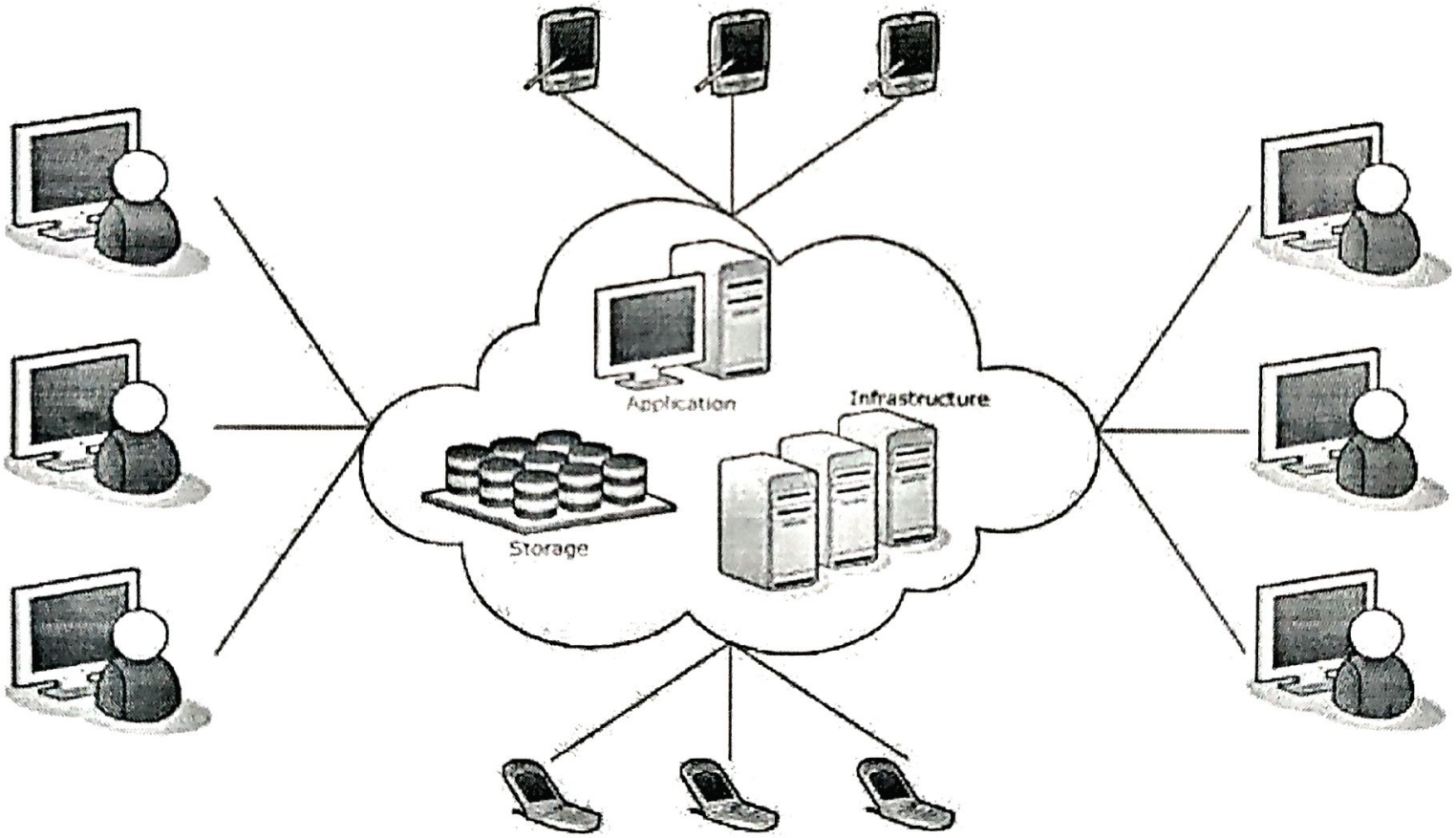
Example: AWS, Azure, Google Cloud

History of Cloud Computing

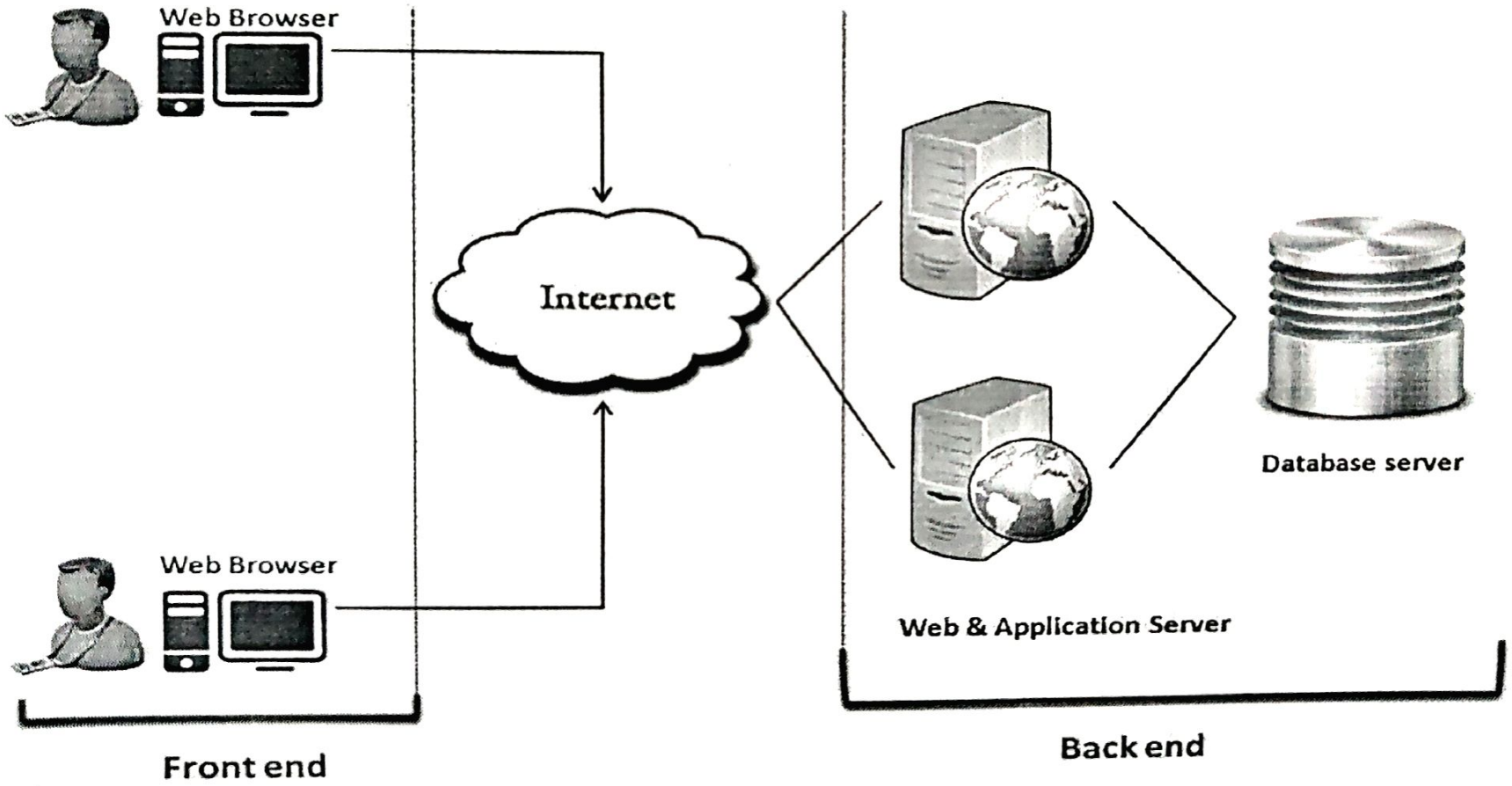
- The concept of **Cloud Computing** came into existence in the year 1950 with implementation of mainframe computers, accessible via **thin clients.** Since then, cloud computing has been evolved from thin clients to dynamic ones and from software to services. The following diagram explains the evolution of cloud computing:



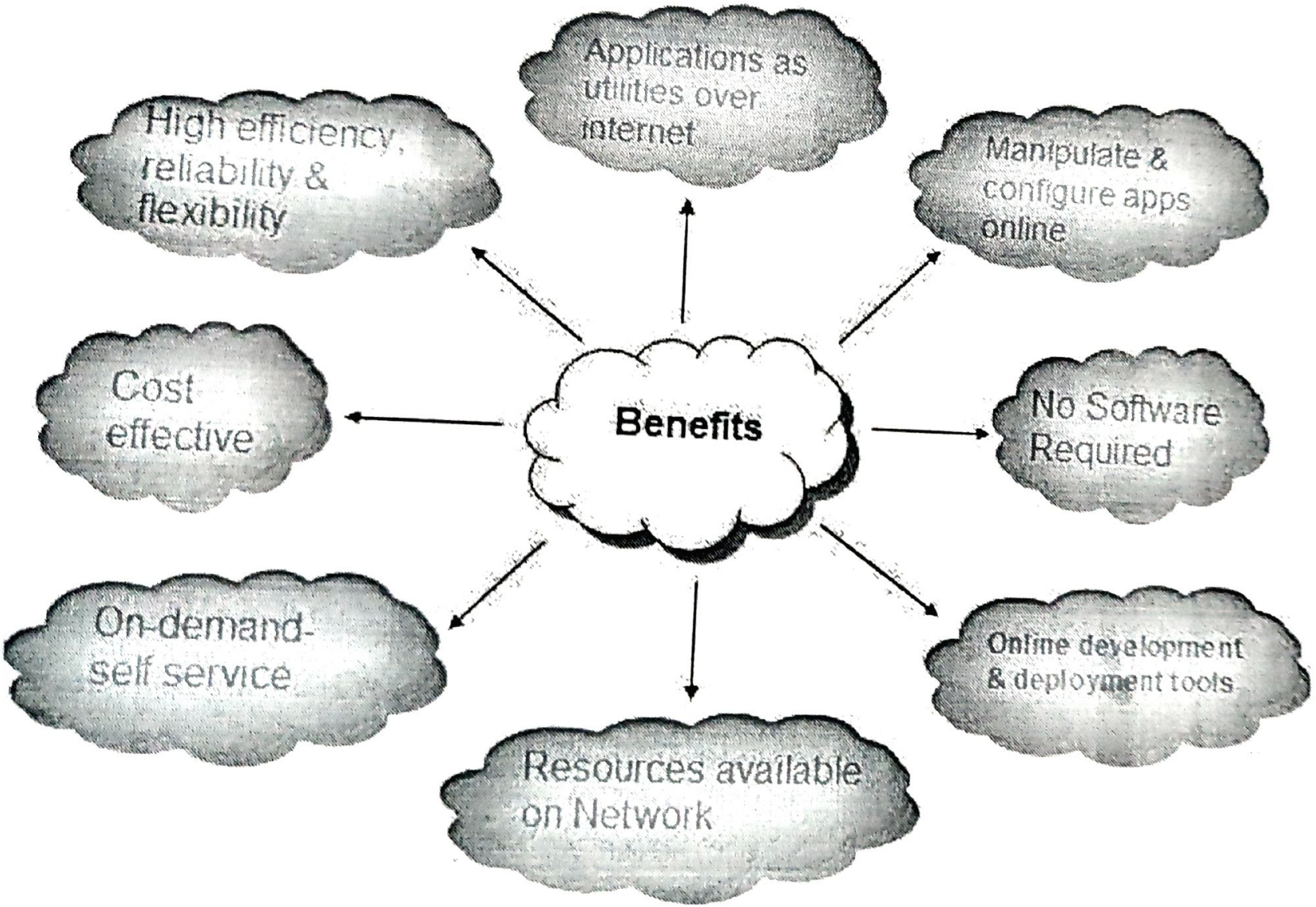
Conceptual view of cloud computing



What is Cloud Computing Architecture?



Benefits


Applications as utilities over internet
High efficiency, reliability & flexibility
Manipulate & configure apps online
Cost effective
Benefits
No Software Required
On-demand-self service
Online development & deployment tools
Resources available on Network

Benefits of Cloud Computing

- Lower IT infrastructure and computer costs for users
- Improved performance
- Fewer Maintenance issues
- Instant software updates
- Improved compatibility between Operating systems
- Backup and recovery
- Performance and Scalability
- Increased storage capacity
- Increase data safety

Risks related to Cloud Computing

- **Security and Privacy**

It is the biggest concern about cloud computing. Since data management and infrastructure management in cloud is provided by third-party, it is always a risk to handover the sensitive information to cloud service providers.

- **Lock In**

It is very difficult for the customers to switch from one **Cloud Service Provider (CSP)** to another. It results in dependency on a particular CSP for service.

- **Isolation Failure**
- **Management Interface Compromise**

In case of public cloud provider, the customer management interfaces are accessible through the Internet.

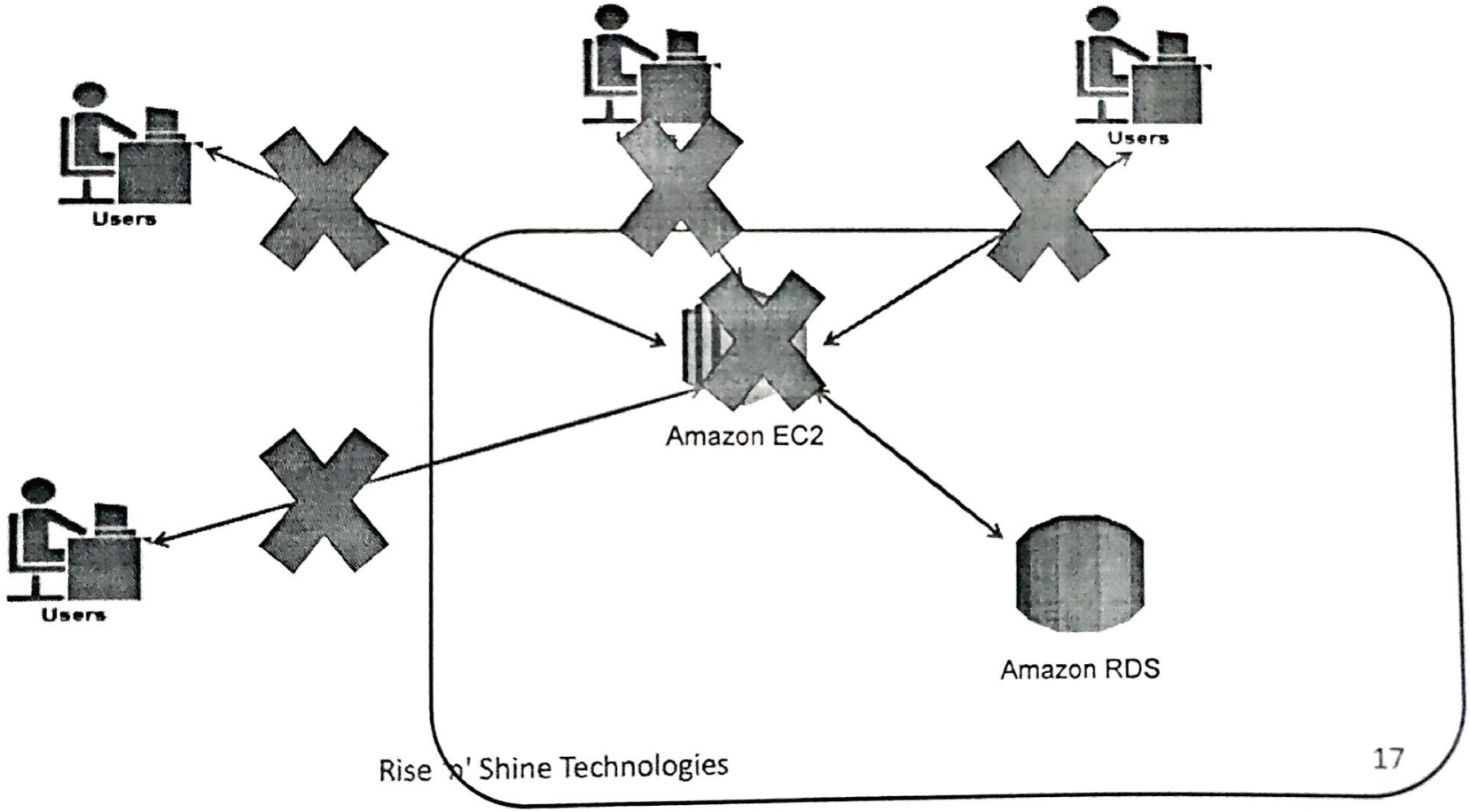
- **Insecure or Incomplete Data Deletion**
  - Extra copies of data are stored but are not available at the time of deletion
  - Disk that stores data of multiple tenants is destroyed.

Cloud Terminology:

- **High Availability** In computing, the term availability is used to describe the period of time when a service is available
- **Fault Tolerant:** is the property that enables a system to continue operating properly in the event of the failure of some (one or more **faults** within) of its components.
- **Scalability:** "Increasing" the capacity to meet the "increasing" workload.
- **Elasticity:** "Increasing or reducing" the capacity to meet the "increasing or reducing" workload.

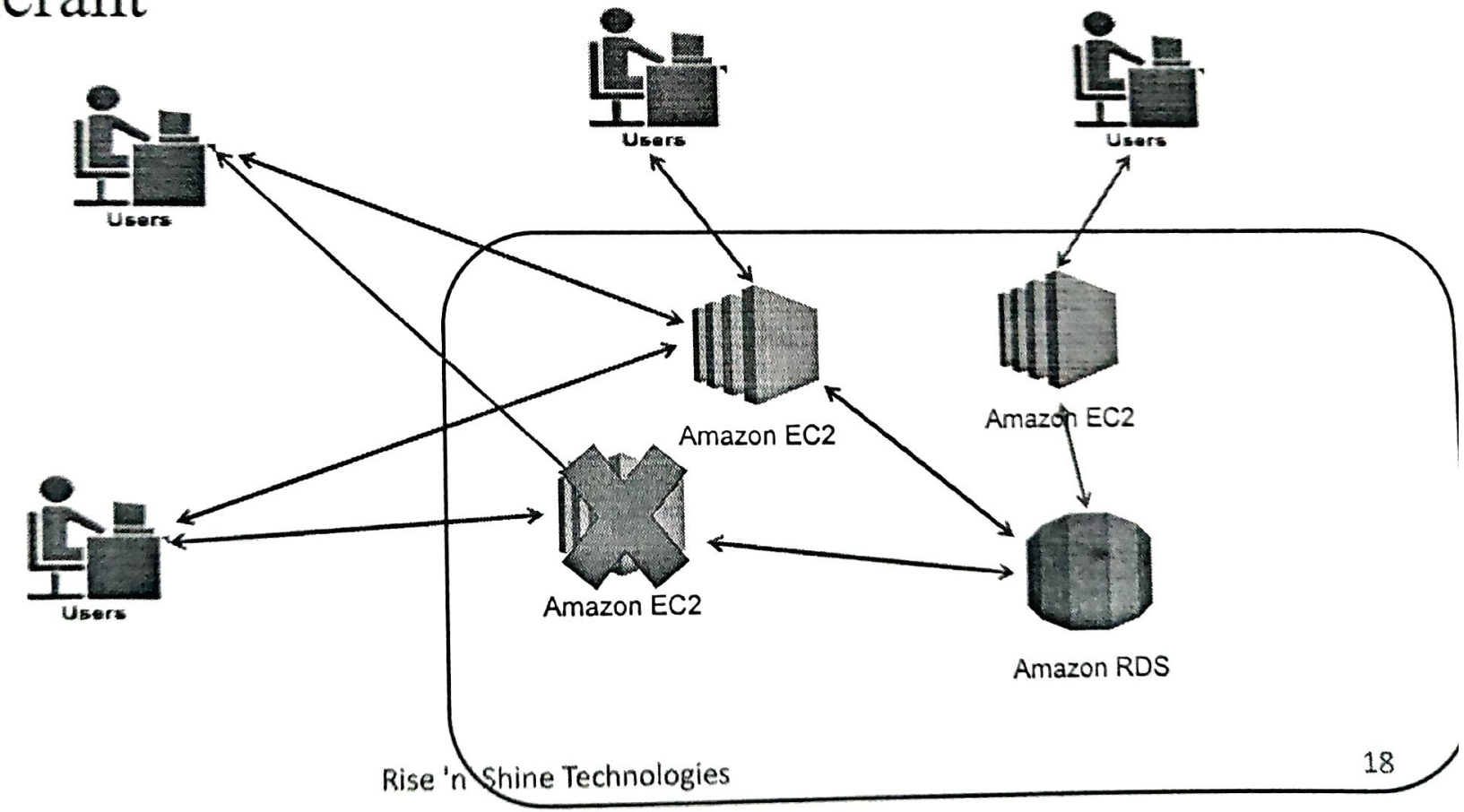
Cloud Terminology:

- High Availability



Cloud Terminology:

- High Availability
- Fault Tolerant

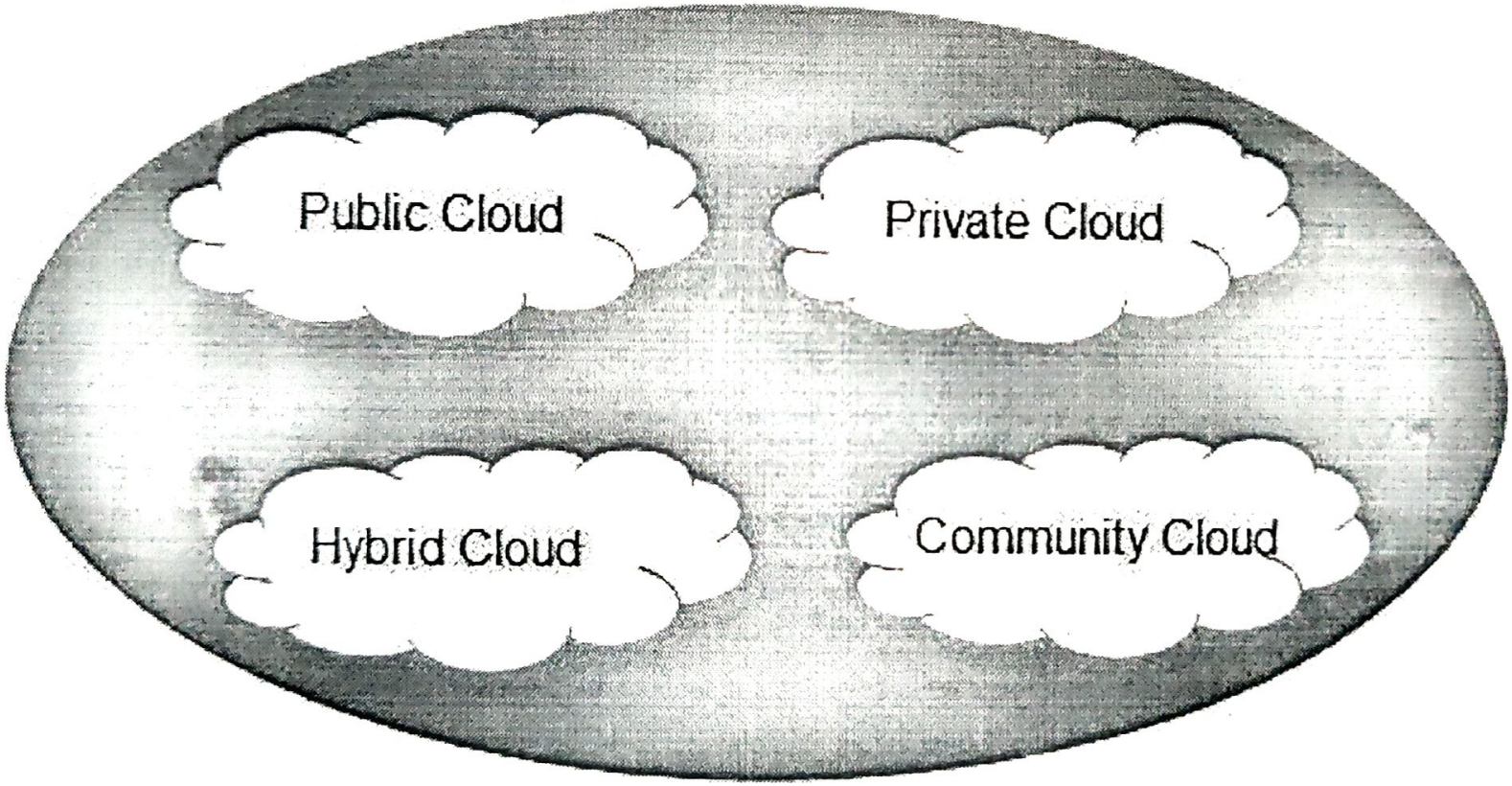


Basic Concepts

There are certain **services** and **models** working behind the scene making the cloud computing feasible and accessible to end users. Following are the **working models** for cloud computing:

- **Deployment Models**
  - Public Cloud
  - Private Cloud
  - Hybrid Cloud
  - Community Cloud
- **Service Models**
  - IAAS
  - PAAS
  - SAAS
  - Anything-as-a-Service (XaaS) is yet another service model, which includes Network-as-a-Service, Business-as-a-Service, Identity-as-a-Service, Database-as-a-Service or Strategy-as-a-Service.

Types of Clouds



Types of Cloud Explained

- **PUBLIC CLOUD**

The **public cloud** allows systems and services to be easily accessible to the general public. Public cloud may be less secure because of its openness.

- **PRIVATE CLOUD**

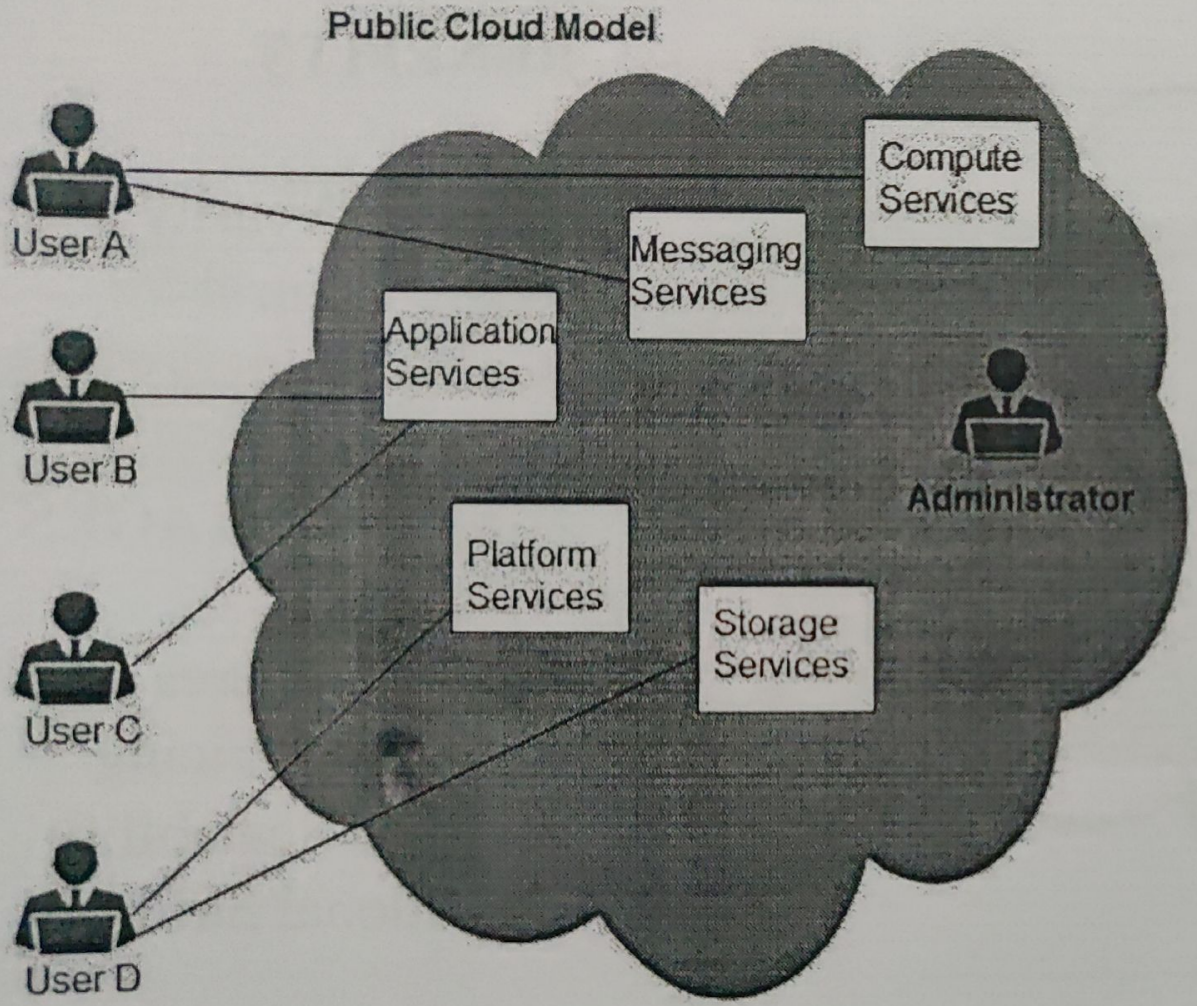
The **private cloud** allows systems and services to be accessible within an organization. It is more secured because of its private nature.

- **COMMUNITY CLOUD**

The **community cloud** allows systems and services to be accessible by a group of organizations.

- **HYBRID CLOUD**

The **hybrid cloud** is a mixture of public and private cloud, in which the critical activities are performed using private cloud while the non-critical activities are performed using public cloud.

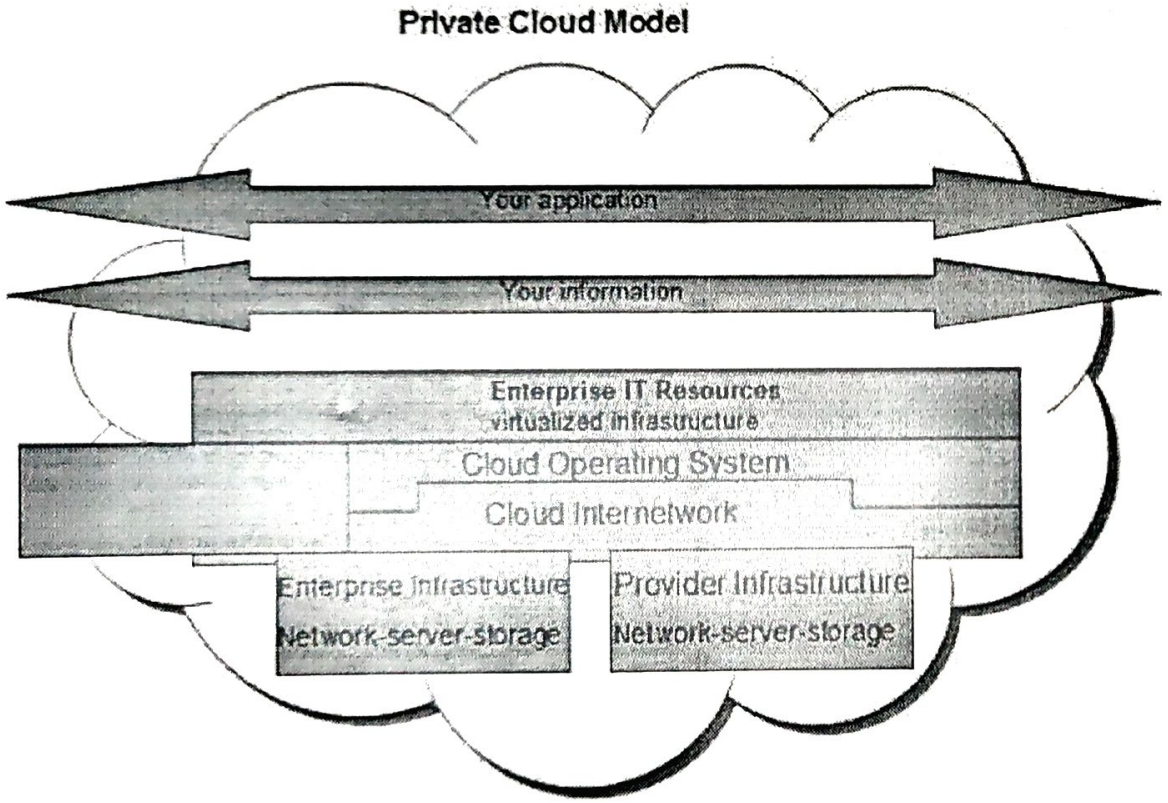


BENEFITS

- Cost Effective
- Reliability
- Flexibility
- Location Independence
- Utility Style Costing
- High Scalability

DISADVANTAGES

- Low Security
- Less customizable

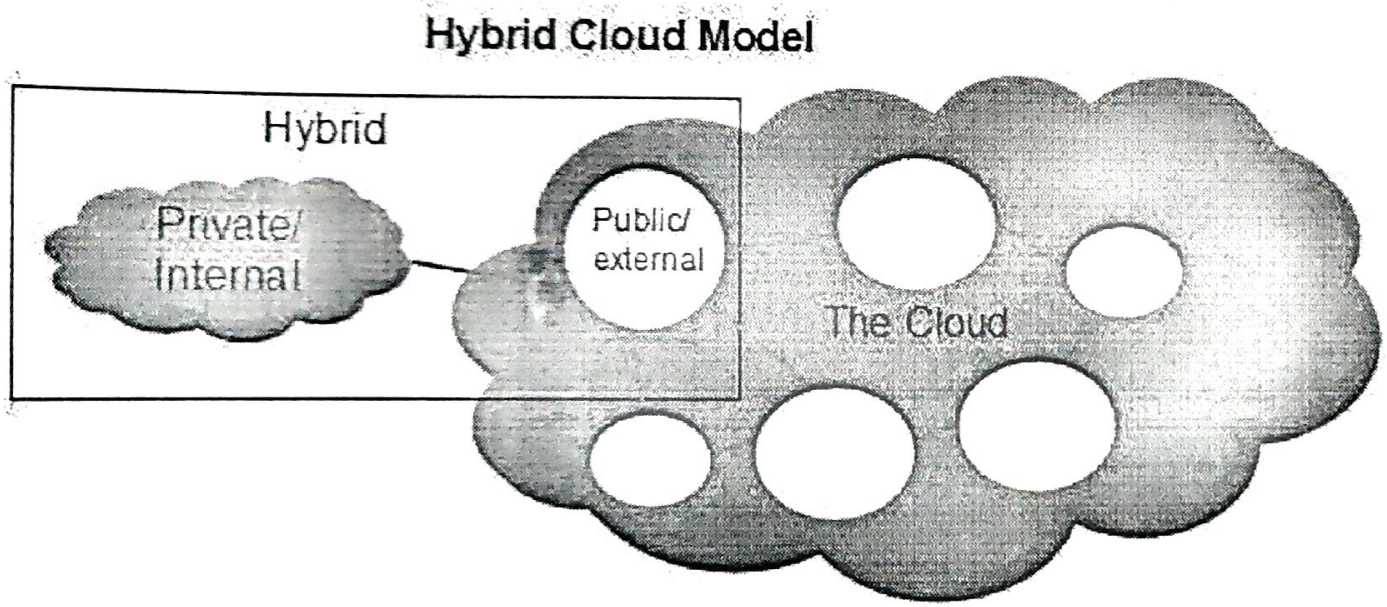


BENEFITS

- Higher Security and Privacy
- More Control
- Cost and energy efficiency

DISADVANTAGES

- Restricted Area
- Inflexible Pricing
- Limited Scalability
- Additional Skills

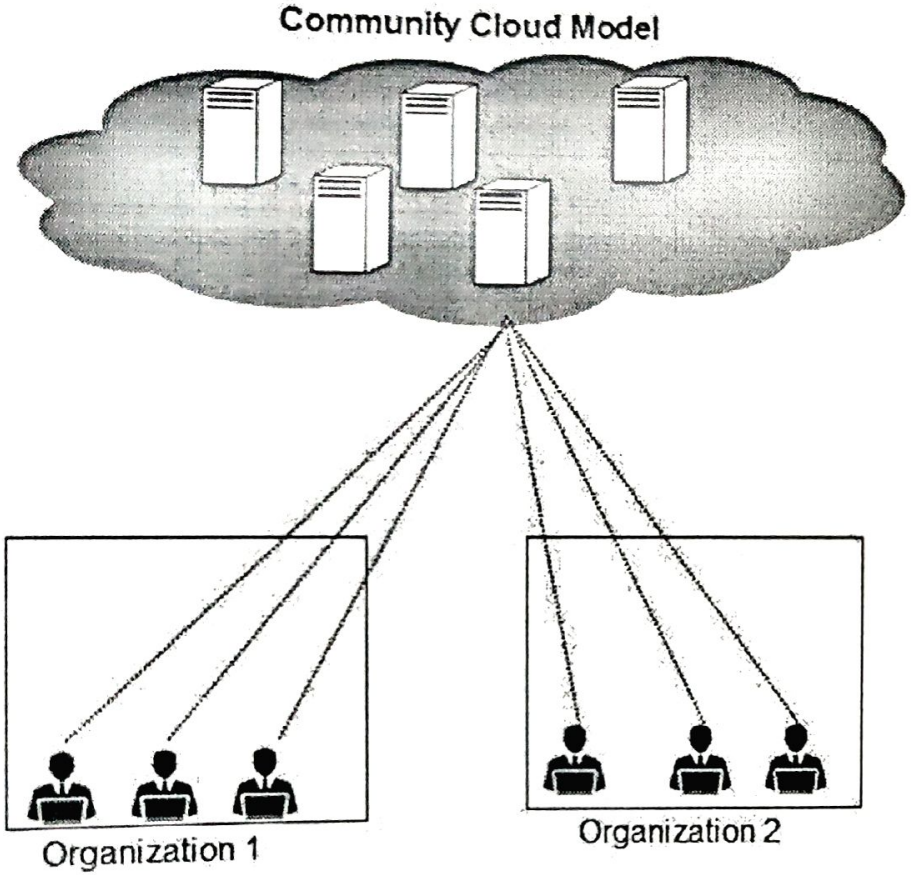


BENEFITS

- Scalability
- Flexibility
- Cost Efficiencies

DISADVANTAGES

- Networking Issues
- Security Compliance
- Infrastructural Dependency



BENEFITS

- Cost effective
- Sharing Between Organizations
- Security

ISSUES

- Since all data is housed at one location, therefore one must be careful in storing data in community cloud because it might be accessible by others.
- It is also challenging to allocate responsibilities of governance, security and cost.

Service Models

- **INFRASTRUCTURE-AS-A-SERVICE (IAAS)**

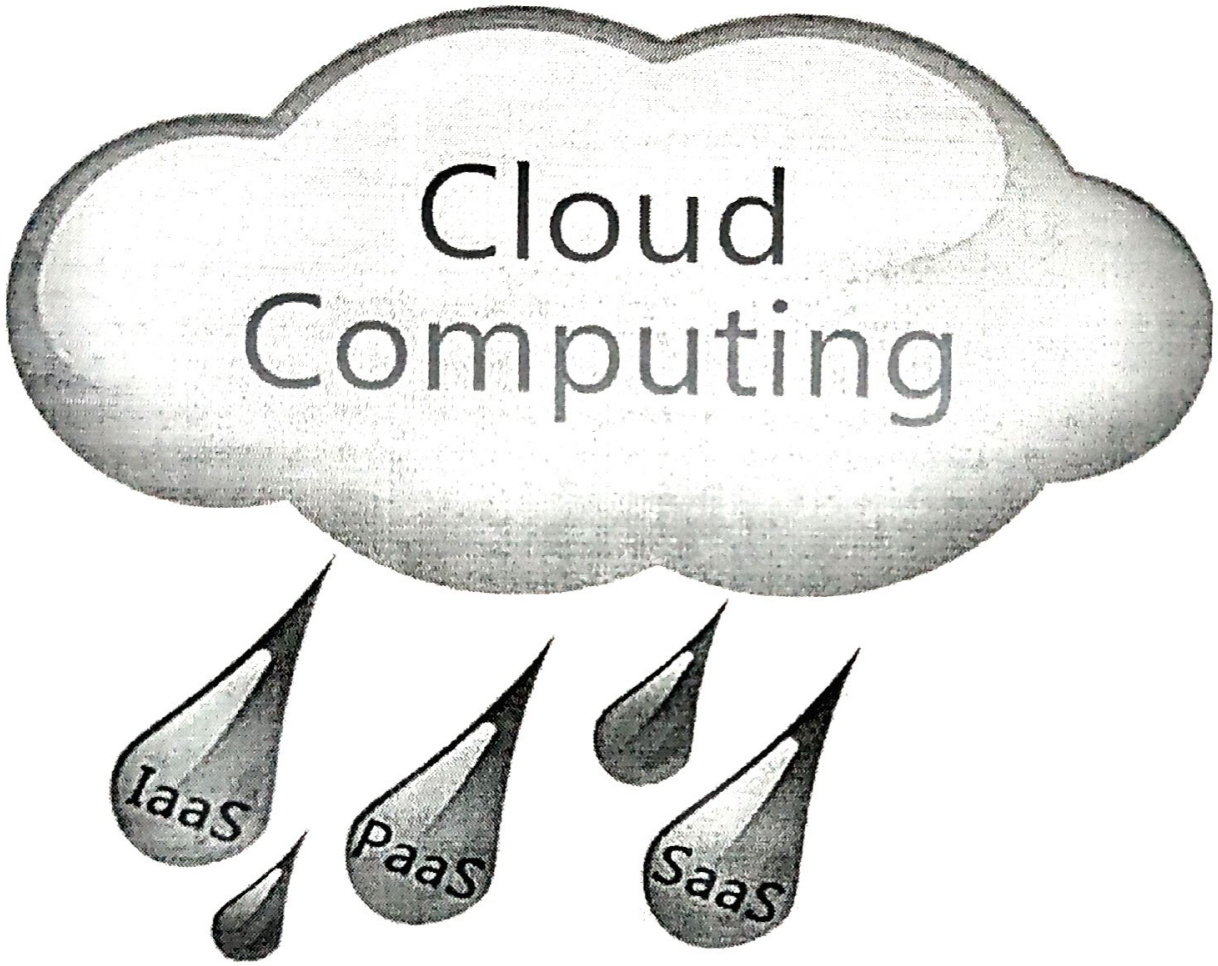
IaaS provides access to fundamental resources such as physical machines, virtual machines, virtual storage, etc.

- **PLATFORM-AS-A-SERVICE (PAAS)**

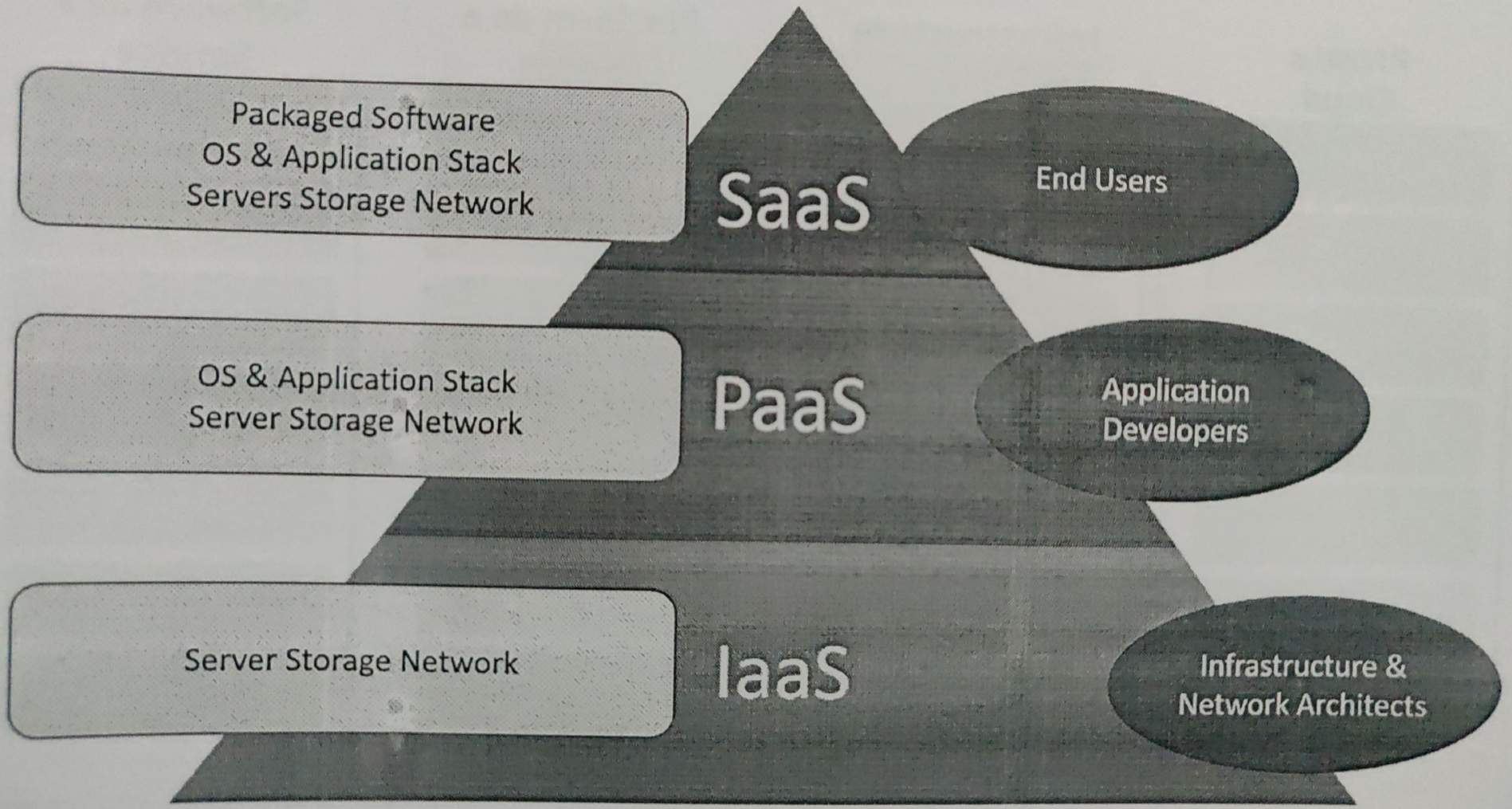
Deploy application without managing virtual servers (Google App Engine, , AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com)

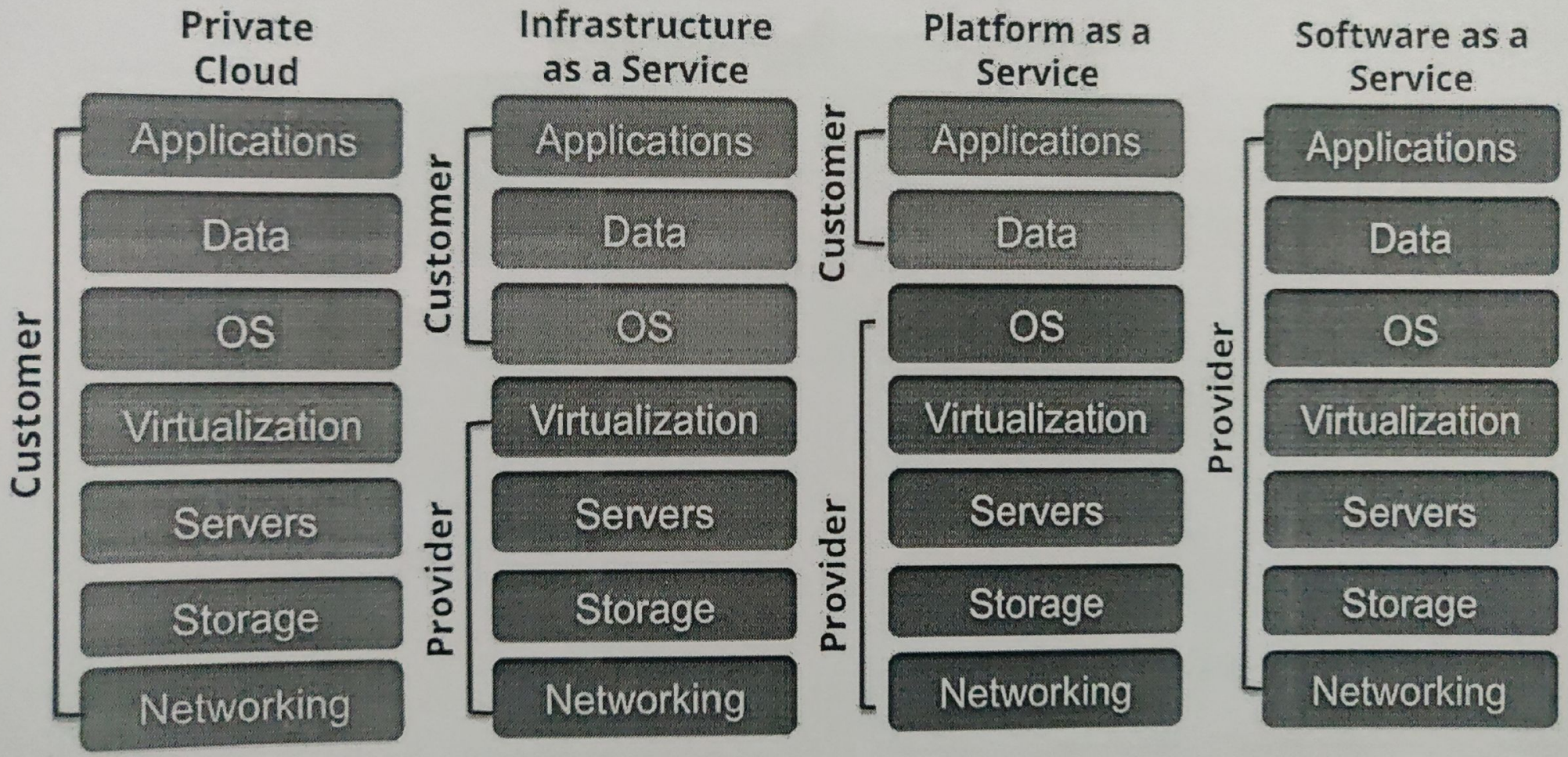
- **SOFTWARE-AS-A-SERVICE (SAAS)**

Ready to use software applications (Gmail, Office365, Google Apps, Dropbox, Salesforce, Cisco WebEx, Concur, GoToMeeting)

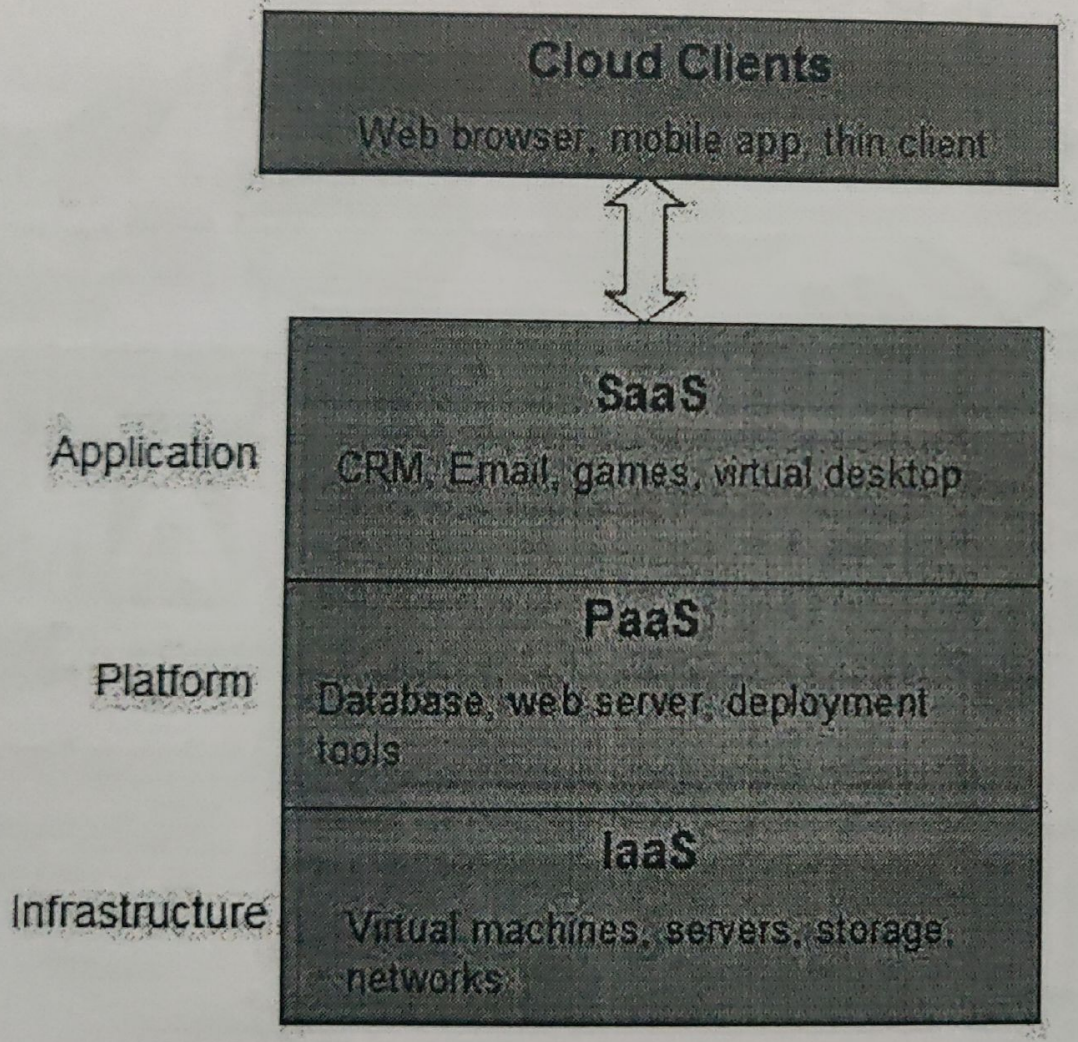

Cloud
Computing
IaaS
PaaS
SaaS

Cloud Service Models




Private Cloud
Customer
Applications
Data
OS
Virtualization
Servers
Storage
Networking
Infrastructure as a Service
Customer
Applications
Data
OS
Provider
Virtualization
Servers
Storage
Networking
Platform as a Service
Customer
Applications
Data
Provider
OS
Virtualization
Servers
Storage
Networking
Software as a Service
Provider
Applications
Data
OS
Virtualization
Servers
Storage
Networking

Service Models



The Cloud Scales: Customers in 190Countries

Summary

- Common use cases of Infrastructure
- What is Cloud & Cloud Computing
- History of Cloud Computing
- Cloud Computing Architecture
- Deployment Models(Public, Private, Hybrid & Community Clouds)
- Service Models(Iaas, Paas, SaaS & Xaas)
- Benefits of Cloud Computing
- Risks of Cloud Computing
- High Availability, Fault Tolerance, Scalability & Elasticity