IT601 – System and Network Administration

# Course Introduction

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Introduction

- **Computers and networks are now a days a vital part of the daily life.**
  - Applications are every where e.g SOHO, SME, NGOs, Government and Educational Institutes
  - User of Computers have increased, esp the networked computers
  - Technologies are diversified and heterogeneous
  - Architectural approaches are also diversified
- **The need to have specialized people to operate and maintain computers and networks**
  - Build the infrastructure
  - Operate and maintain
  - Provide Support to users
- **System administration matters because computers and networks matter**

# Introduction

- **This course focuses on the principles of systems and network administration**
  - Introduction To System Administration
  - Servers/Datacenters
  - Services
  - Installation, configurations and verifications of server
  - Support and Processes
  - Network and Traffic Control
  - Best Practices

# Books

- **Following are text and reference books in addition to various articles from internet**
  - The Practice of System and Network Administration, Second Edition by Thomas Limoncelli, Christina Hogan and Strata Chalup, Addison-Wesley Professional; 2nd Edition (2007). ISBN-10: 0321492668
  - The Practice of System and Network Administration, Third Edition by Thomas Limoncelli, Christina Hogan and Strata Chalup, Addison-Wesley Professional; 2nd Edition (2017). ISBN-10: 0321492668
  - Red Hat Enterprise Linux 6 Bible: Administering Enterprise Linux Systems by William vonHagen, 2011
  - Studyguide for Practice of System and Network Administration by Thomas A. Limoncelli, Cram101; 2nd Edition (2011). ISBN-10: 1428851755
  - Networking Systems Design and Development by Lee Chao, CRC Press; 1st Edition (December 21, 2009). ISBN-10: 142009159X (TB2)

# Course Details

- **Credit Hours:**
  - Theory : 3
  - Practicals : 1

- **Assessment**
  - Quizzes
  - Assignments
  - Graded Discussion Board
  - Mid Term Exam
  - Final Term Exam
  - Practical Exam

# Prerequisites

- Operating Systems
- Data Communications  and Computer Networks

# Course Instructors


- Dr. Hasnain Ahmed Bukhari
  - Virtual University of Pakistan
  - Email : hasnain@vu.edu.pk
- Mr. Arif Husen
  - Virtual University of Pakistan
  - Email : arif.husen@vu.edu.pk
- Mrs. Fauzia Jumani
  - Virtual University of Pakistan
  - Email : fouziajumani@vu.edu.pk

# IT SYSTEM

Arif Husen

**Department of Computer Science and Information Technology,**
**Virtual University of Pakistan**

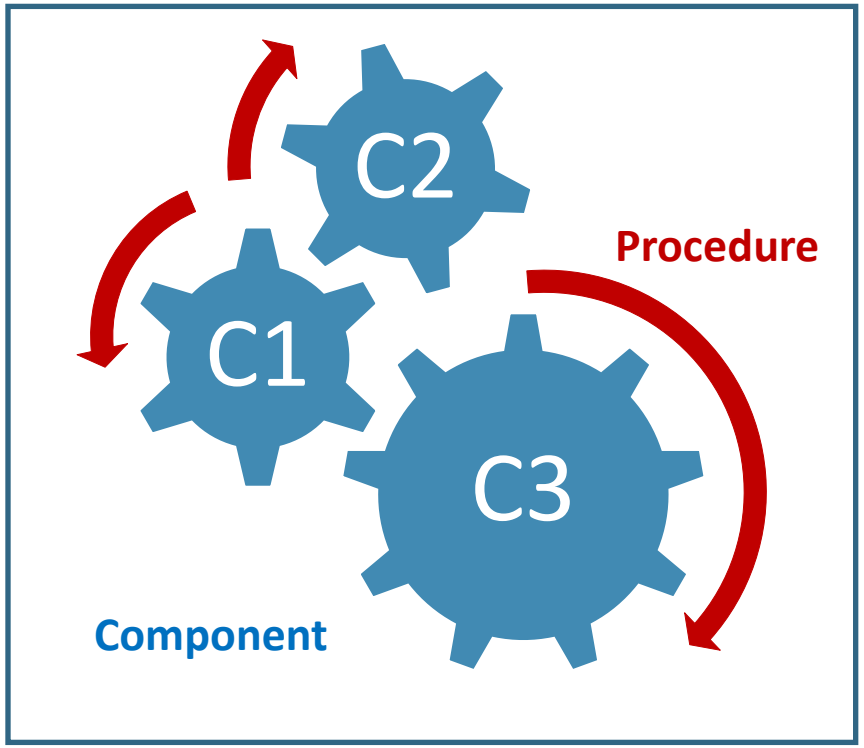
# System

- **System**
- **IT System**
- **General Definition**

# IT System

## A. What is a System?

A system is a collection of elements or components that are organized for a common purpose

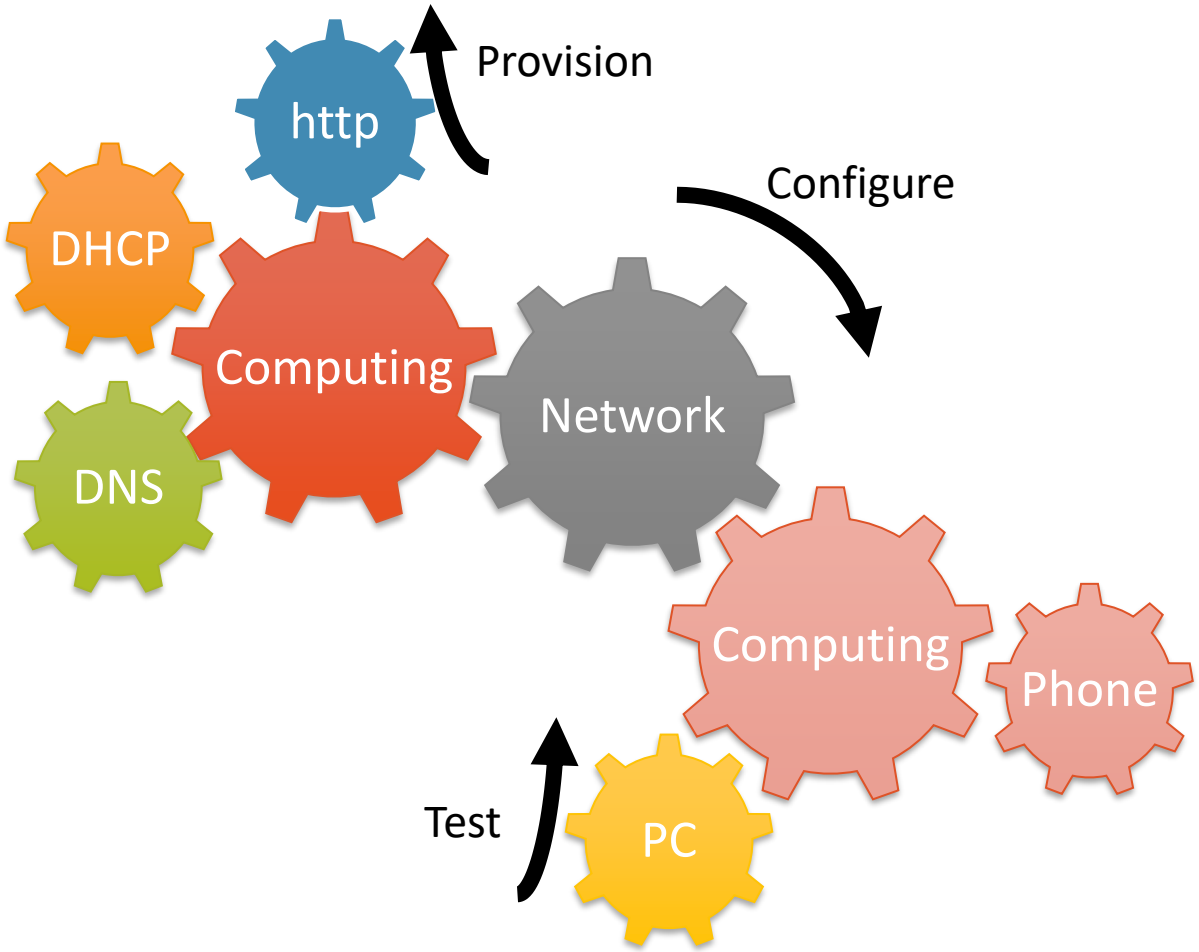



| Common Purpose | Organized |
| --- | --- |

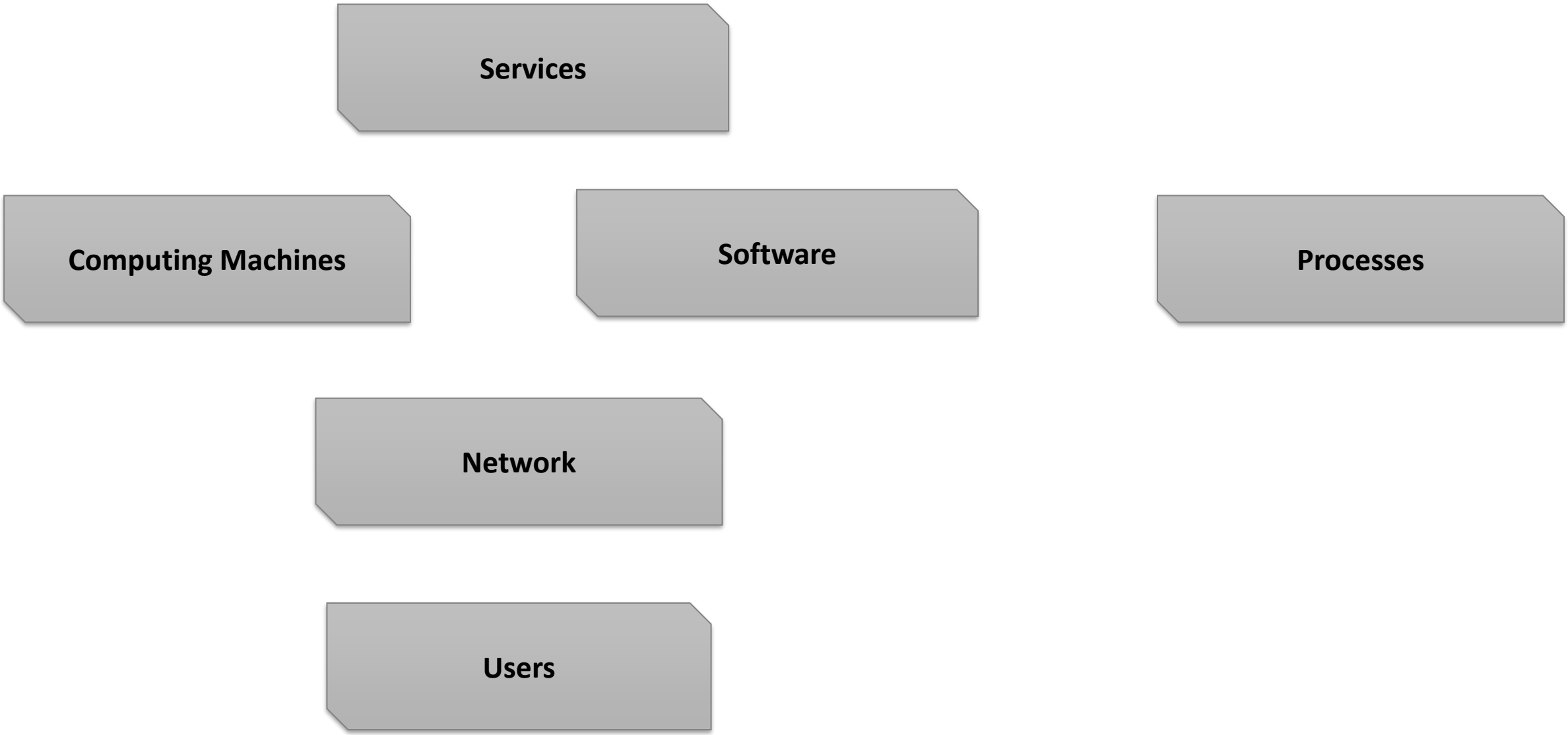
# IT System

## A. What is a IT System?

a collection of computing and/or communications components and other resources that support one or more functional objectives of an organization



# IT System Components



# IT System Components

- **Computing Machines**
- **Software**
- **Network**
- **Services**
- **Users**
- **Processes**

Computing Machines are important component of IT Systems and there are different forms of computing machines in an organization

**Servers/ Workstation**

**Dekstops**

**Laptops/ Mobile Devices**

# IT System Components

- Computing Machines
- **Software**
- Network
- Services
- Users
- Processes


Operating Systems


Applications


Utilities

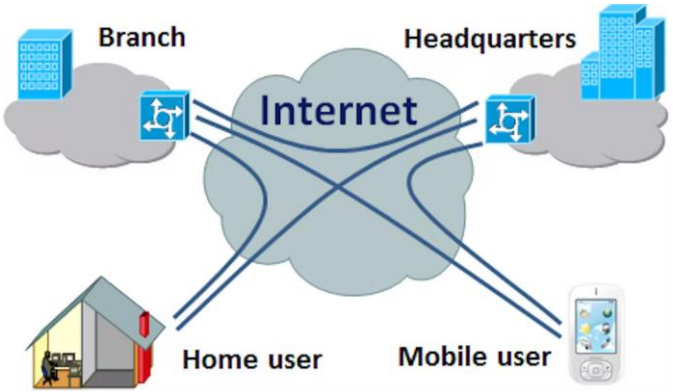
# IT System Components

- Computing Machines
- Software
- **Network**
- Services
- Users
- Processes

Wireless


Router
Switch
Firewall
DPI
Loadbalancers

Cable and Connectors

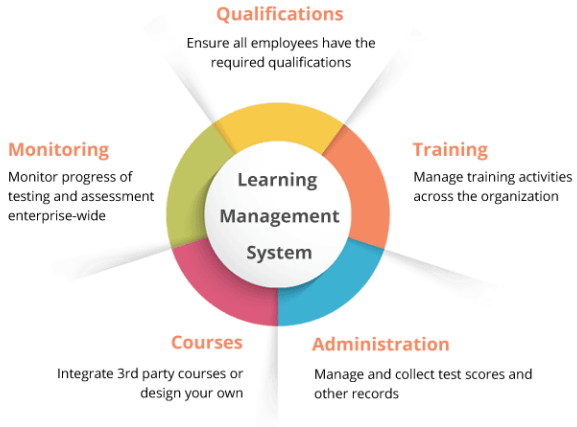



VPNs

# IT System Components

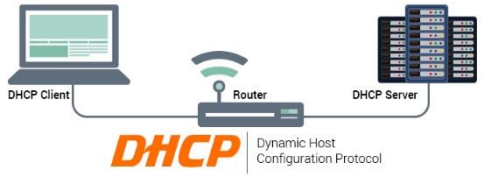
- Computing Machines
- Software
- Network
- **Services**
- Users
- Processes


Application Services





Webservices

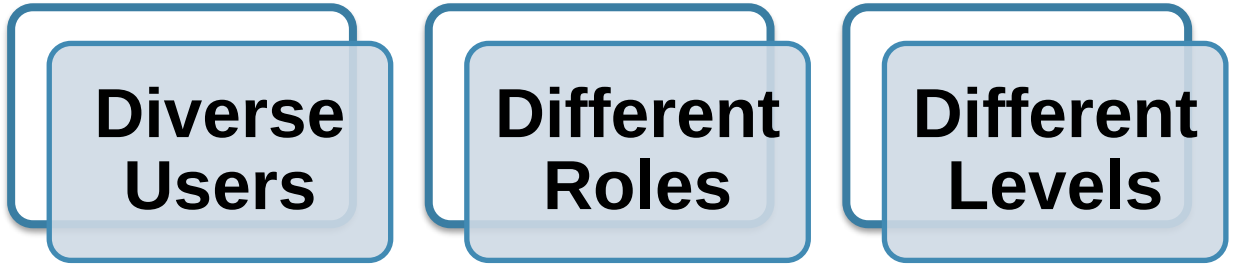




Network Sevices

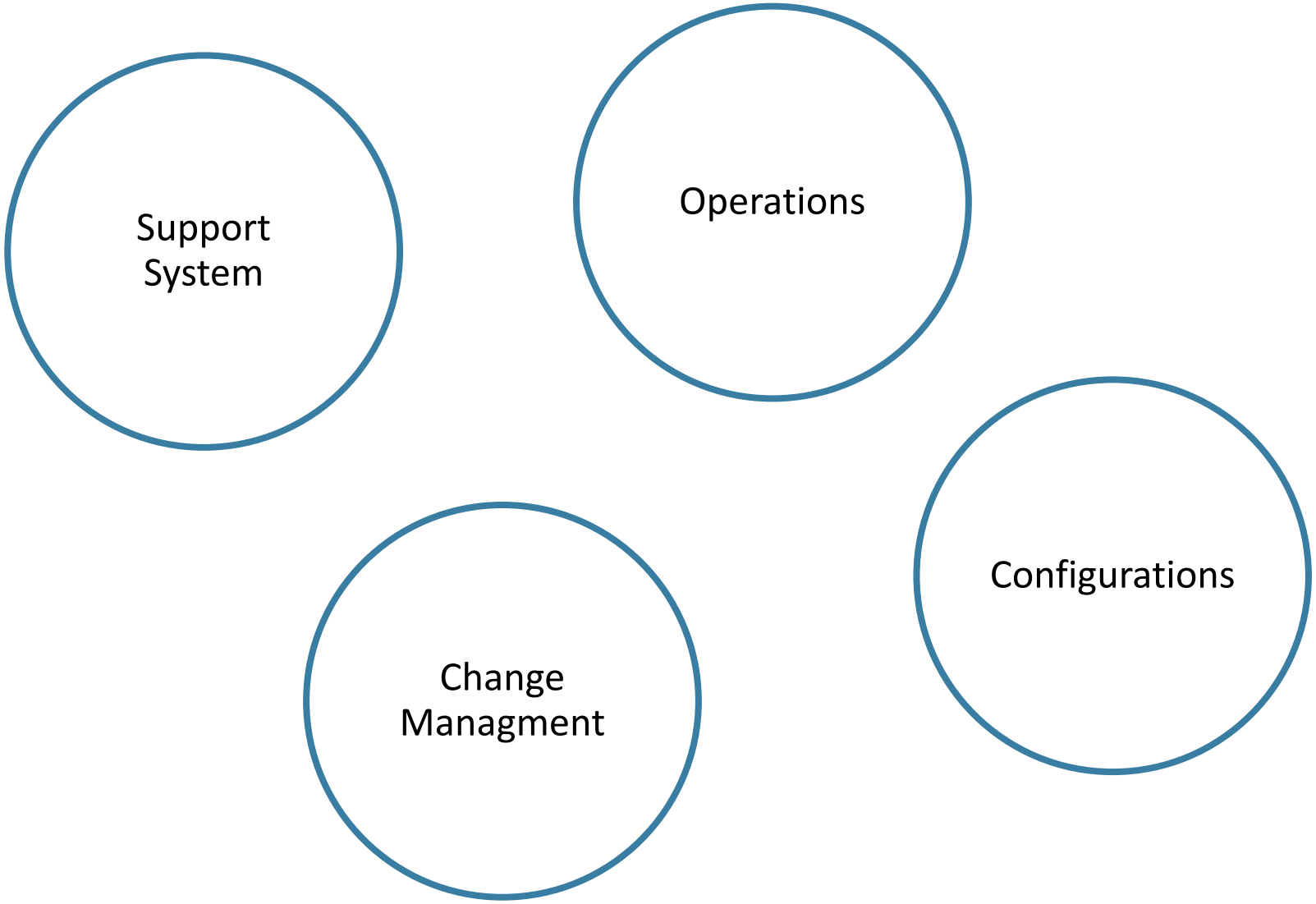
# IT System Components

- Computing Machines
- Software
- Network
- Services
- **Users**
- Processes



# IT System Components

- **Computing Machines**
- **Software**
- **Network**
- **Services**
- **Users**
- **Processes**



# IT System

Servers

Runs on

Webserver

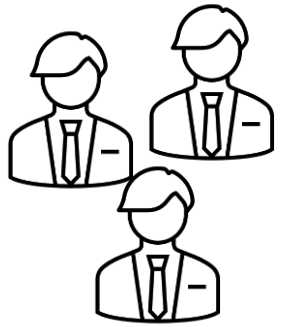
DHCP

Database

DNS

Services

Accessed With



Users

Do work

Workstations/Desktops

IT601 – System and Network Administration

# STATES OF COMPUTER MACHINES

Arif Husen

Department of Computer Science and Information Technology,

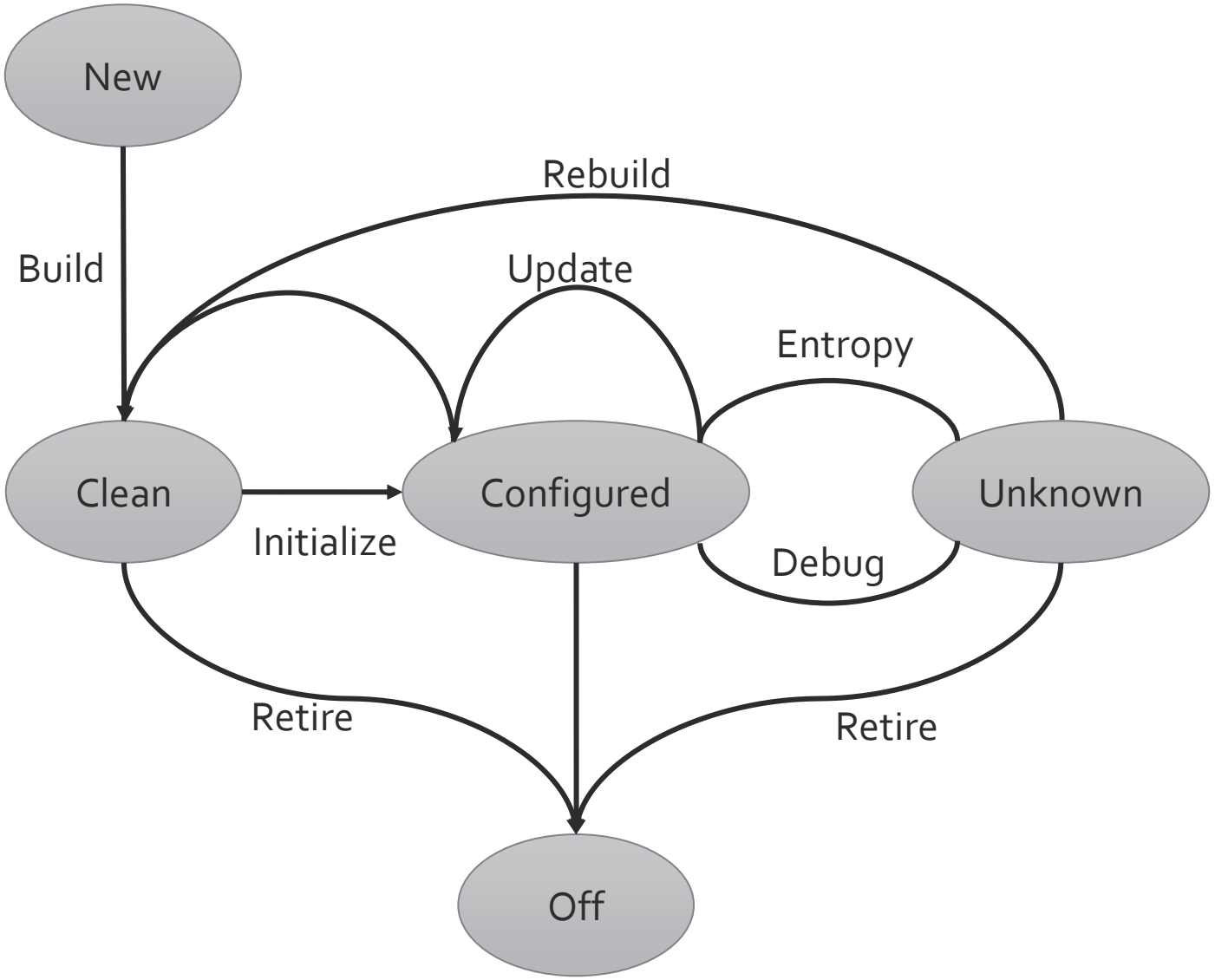
Virtual University of Pakistan

# Computing Machines

- **A Computer Machine Consists of ….**
  - Hardware Components
  - Operating System
  - Accessories
  - Has Different States and Several Processes Associated with it

- **Evard's Computer Machine Life Cycle**
  - A Computer Machine Goes through different states in their life.
  - Evard has outline different states and process that are performance on

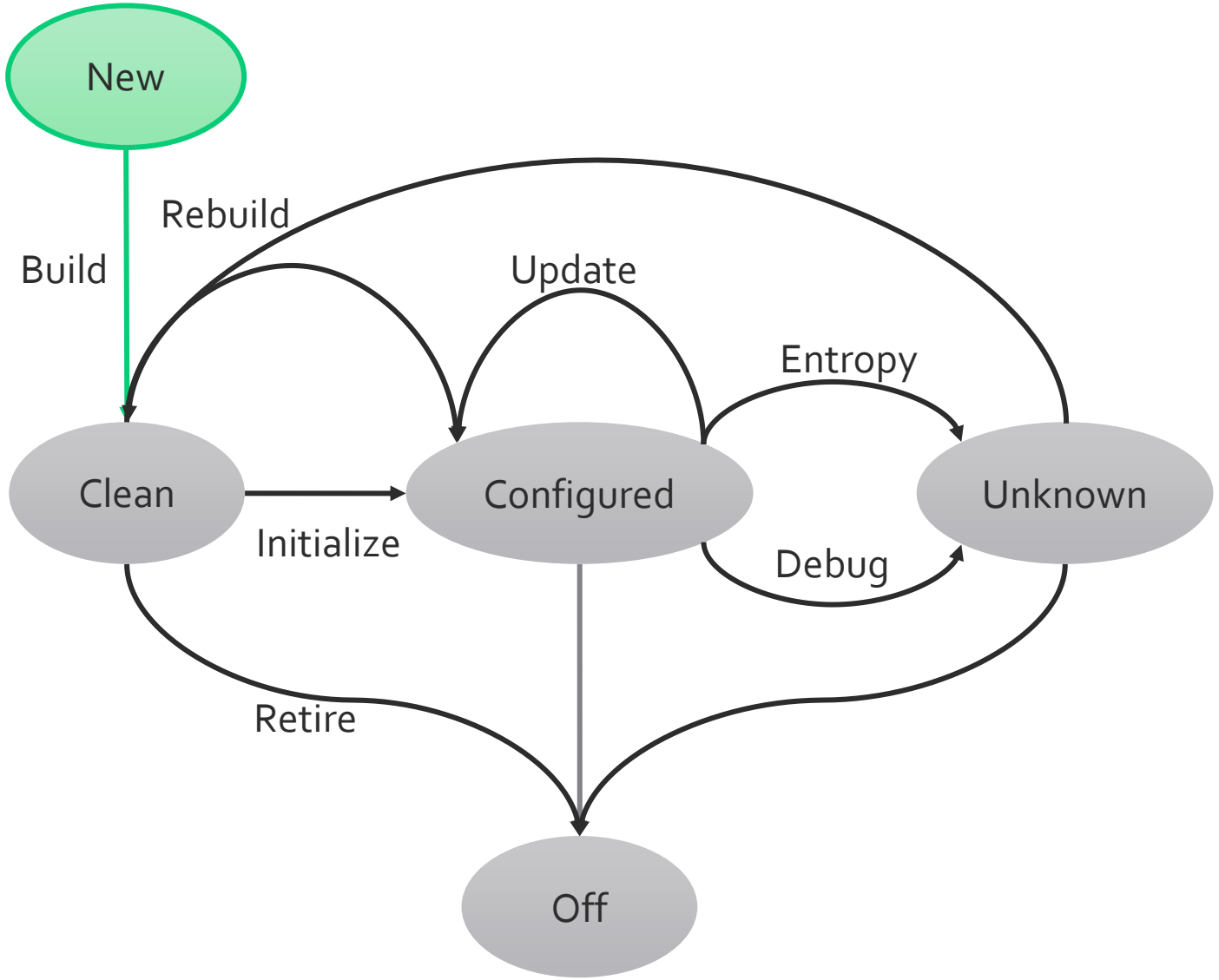
# Evards ' Life Cycle Of Computer Machine




- **Five States**
  - New
  - Clean
  - Configured
  - Unknown
  - Off

- **Several Processes**
  - Build
  - Initialize
  - Update
  - Entropy
  - Debug
  - Retire

# Evards 'S Life Cycle Of Computer Machine




**New State**

A new machine

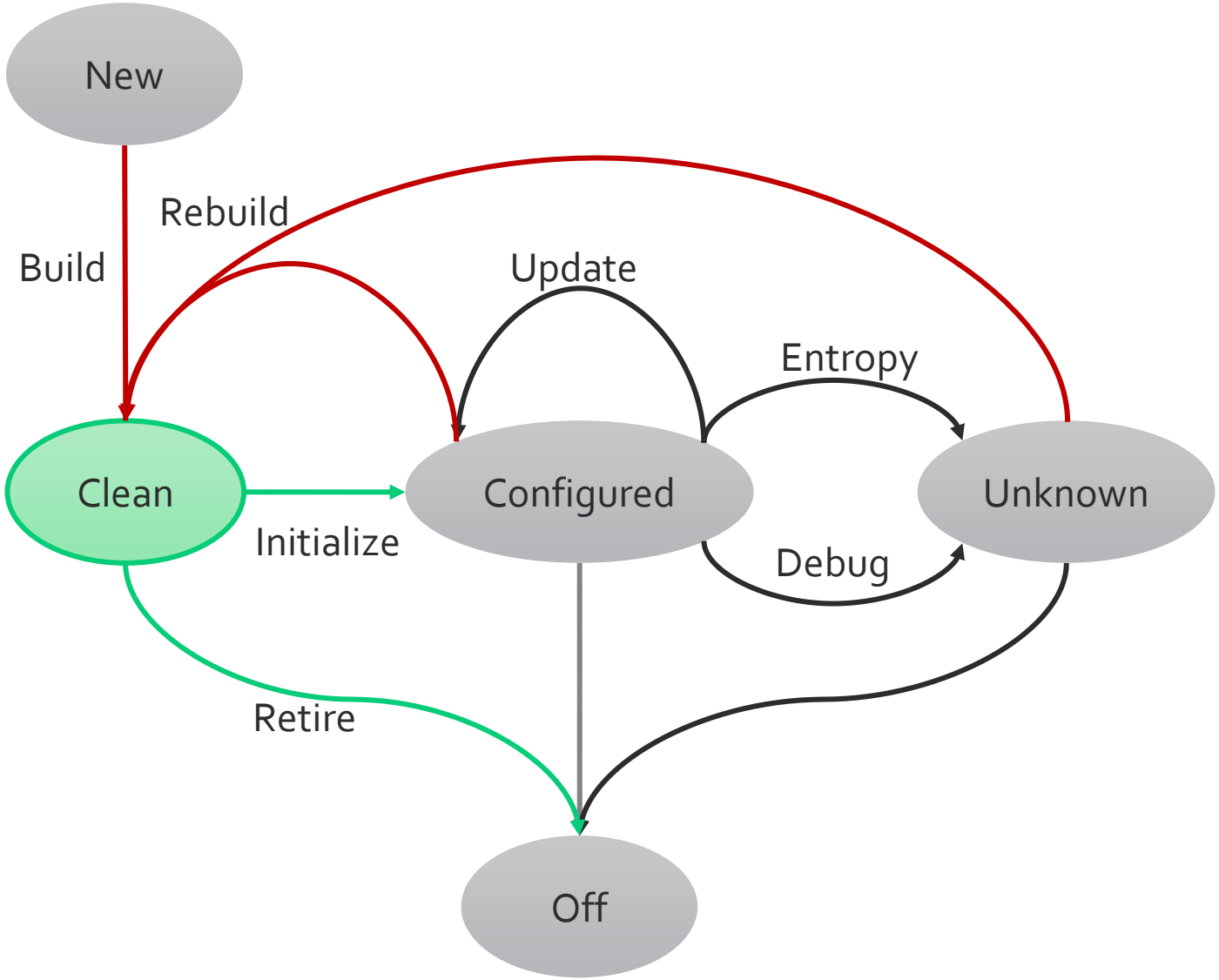
Unpacking

Physical Connections

Power up

Preinstalled OS or No OS

# Evards 'S Life Cycle Of Computer Machine




**Clean State**

A machine that has the operating system installed but isn't set up to function in the System.

**Build Process**

Select OS

Install OS

Drivers and Softwares

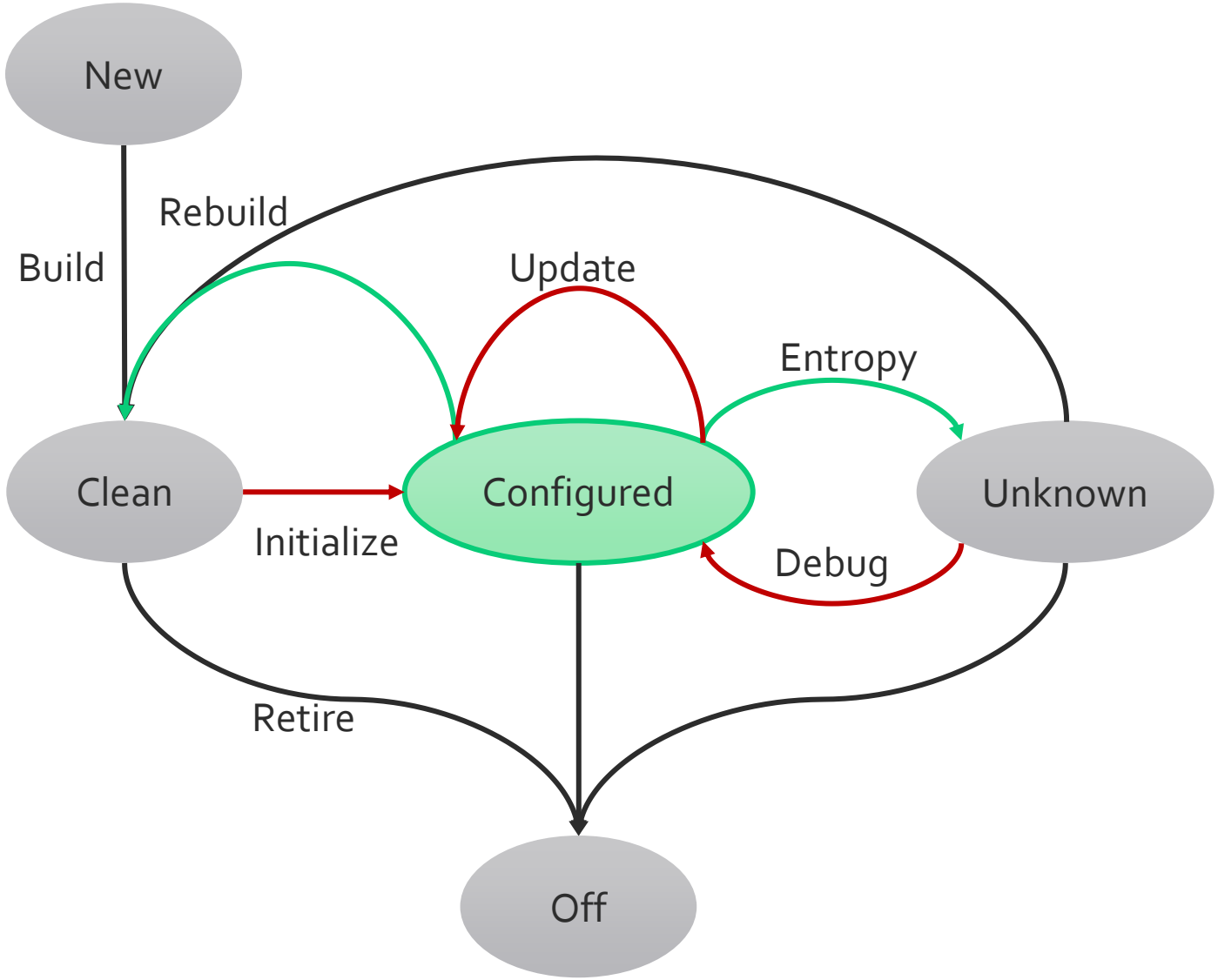
**Rebuild Process**

Recovered Data

Reformate Disks

Reinstall OS/Software

# Evards 'S Life Cycle Of Computer Machine




**Configured State**

- A machine that is properly configured to meet the needs of the System.
- User Roles and Permissions
- Updates

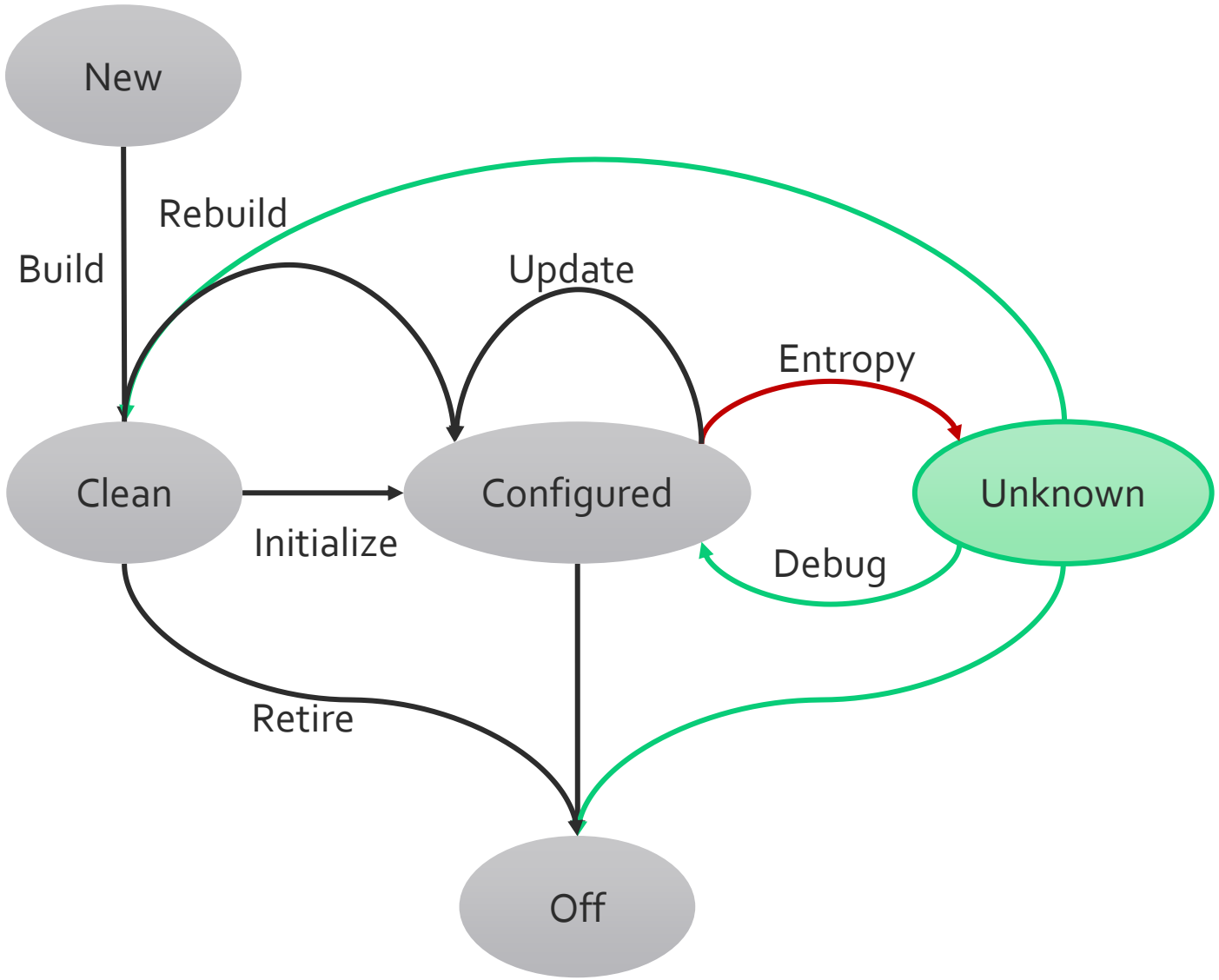
**Update Process**

Select Updates

Schedule Updates

Monitor

# Evards 'S Life Cycle Of Computer Machine




**Unknown State**

- **A machine that has been misconfigured, or that has become out of date, or that has been borrowed by an intern and returned stained**

**Debug Process**

Troubleshooting

Remove the bugs

Fix Drivers and Softwares

**Entropy Process**

Changes by user/admin/updates

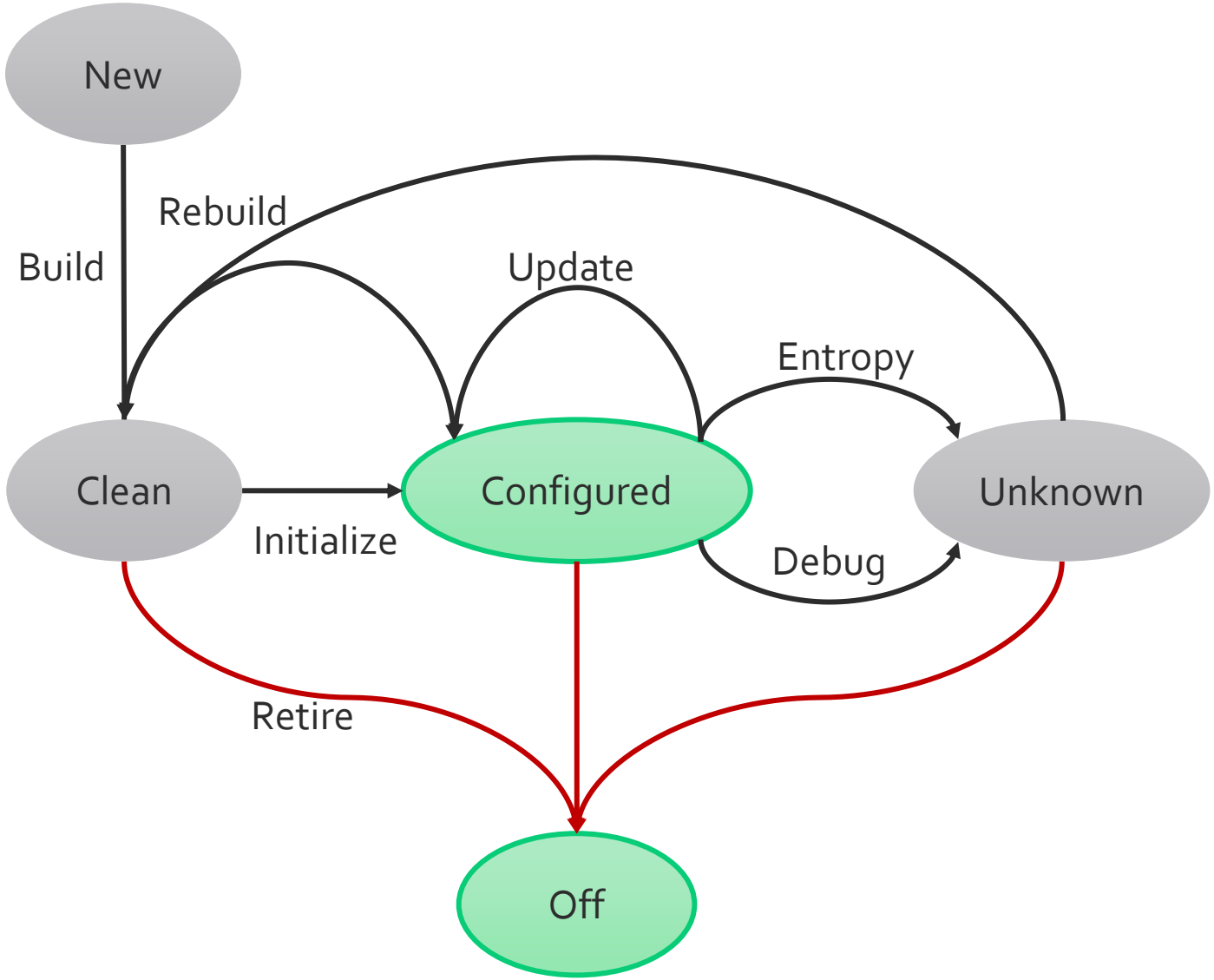
Compromised State

**Rebuild Process**

Recover User Data

Formate Disks and Reinstall OS

# Evards 'S Life Cycle Of Computer Machine




**Off State**

Powered Off

No User Data

OS installed or Removed

Disconnected from Infrasture

**Retire Process**

Recover User Data

Flush the Hard Disks

Isolate from Infrastructure

# Summary

- Computer Machine Passes through 5 states throughout their life cycle
- Different Processes are used for transition of the state of Computer Machines

IT601 – System and Network Administration

# System Administration

Arif Husen

Department of Computer Science and Information Technology,
Virtual University of Pakistan

# TOC

- What is System Administration?, its Importance?
- Orgnization Size play role
- Formal Definition of System Administration.

# What is SNA?

- **What is SNA and Who is System Administrator?**

  1. **Difficult to define because system administrators do so many things**

     - looks after computers, networks, and the people who use them.

     - look after hardware, operating systems, software, configurations, applications, or security

**Computers**

**Networks**

**People**

**Hardwares**

**OS**

**Security**

**Configs**


**SW & Apps**

# What is SNA?

**2 - A system administrator influences how effectively other people can or do use their computers and networks.**

- Help Employees
- Support Others
- Guide Others

**3 - A system administrator sometimes needs to be a**

- business-process consultant
- corporate visionary
- janitor
- software engineer
- electrical engineer
- psychiatrist
- mindreader
- economist

**4 - Known with different names.**

- network administrators
- system architects
- system engineers
- operators
- system programmers

# General Definition

**System administration is the field of work which involves…**

- **Managing one or more systems,**
- **System could be software, hardware, servers or workstations.**
- **The objective are efficiency and effectiveness**

# Importance of System Administration

- **System administration matters because computers and networks matter.**

  Business Operations | Colloborations | Solve Many Problems

- **Business Depends on**

  Internet | Interanet | World Wide Web

- **Management now has a more realistic view of computers, More people becoming direct users of Computers**
- **Unreliable machine room power system that caused occasional but bearable problems now prevents sales from being recorded.**
- **Computers matter more than ever. If computers are to work and work well, system administration matters**

# Orgnization Size and System Administrtaion

- **Small and Home Offices:**
  - Most businesses in the U.S. fall under this category.
  - The typical characteristics of a small business are having no more than 1,500 employees

- **Mid-market enterprise:**
  - These organizations are larger than small businesses but smaller than large enterprises.
  - They generally employ between 1,500 and 2,000 people

- **Large enterprise:**
  - These organizations are relatively few,

# Managing an IT System means?

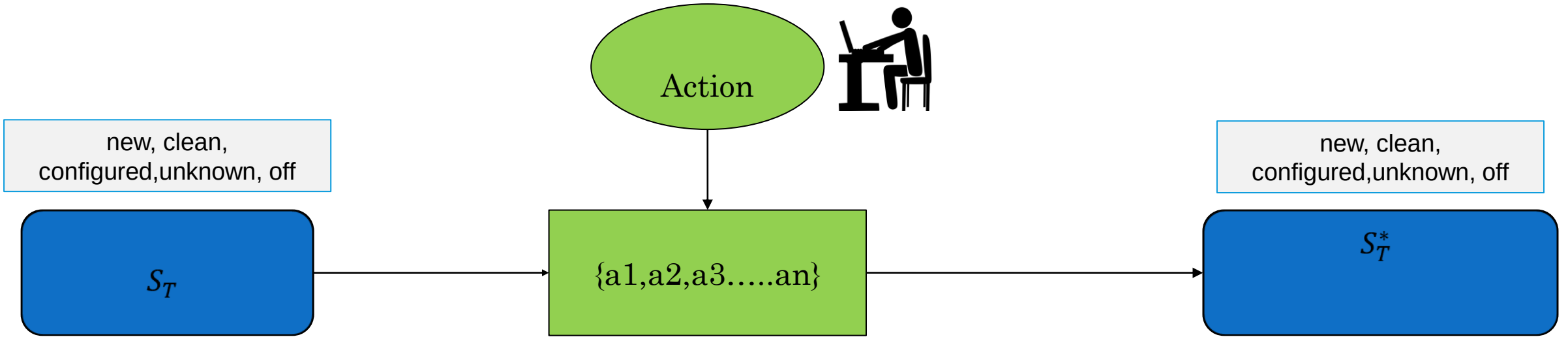
**Applying Tasks and actions**

- **Installation and Configuration**
- **Monitoring, Logging and Reporting**
- **Changing**
- **Supporting**
- **Provisioning**
- **Troubleshooting**

**Changes System State**

# Formal Definition of System Administration

specification of a system Components 's configuration and its acceptable usage.




At Given Time (T) configuration (files, kernel, memory or CPU usage) of a system.

modifying the system to bring it closer to S*(t).

states of the system that match the system policy. Over time, the system state shifts away from the ideal state.

IT601 – System and Network Administration

# System Administration Maturity

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

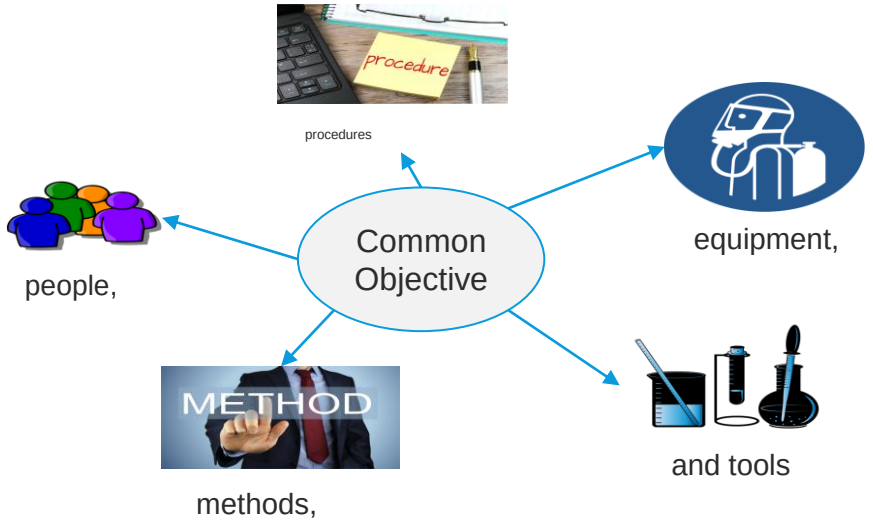
# Maturity Model

**What is a process?**

- **Maturity model in IT?**
- **System Administration Maturity Model**

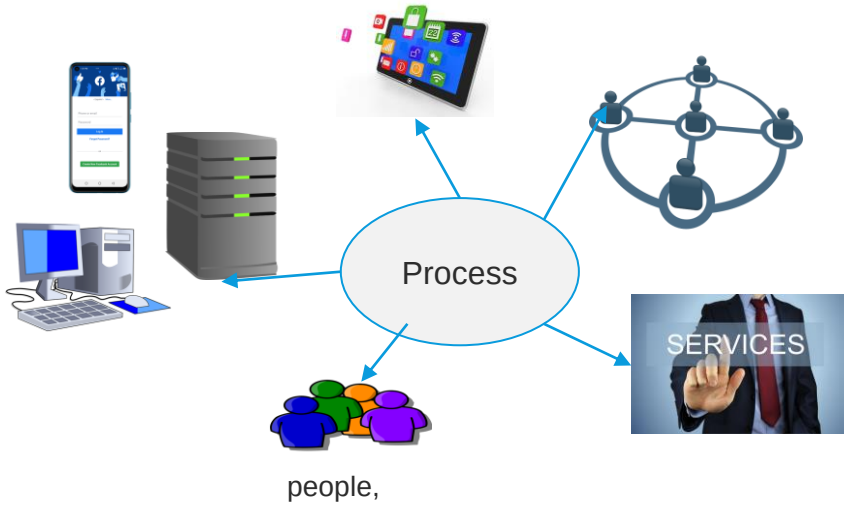
# Process and its Maturity

- The term process describes the means by which the system components are integrated or interact with each other  to produce a desired end result.
- IT System have several components which interacts or integrate with each other for the common objectives.




## Maturity

- Process maturity is a measure of how well defined and controlled a system 's processes are.
- A high level of process maturity shows that processes are documented well, users understand and follow procedures, and there is continuous process improvement.



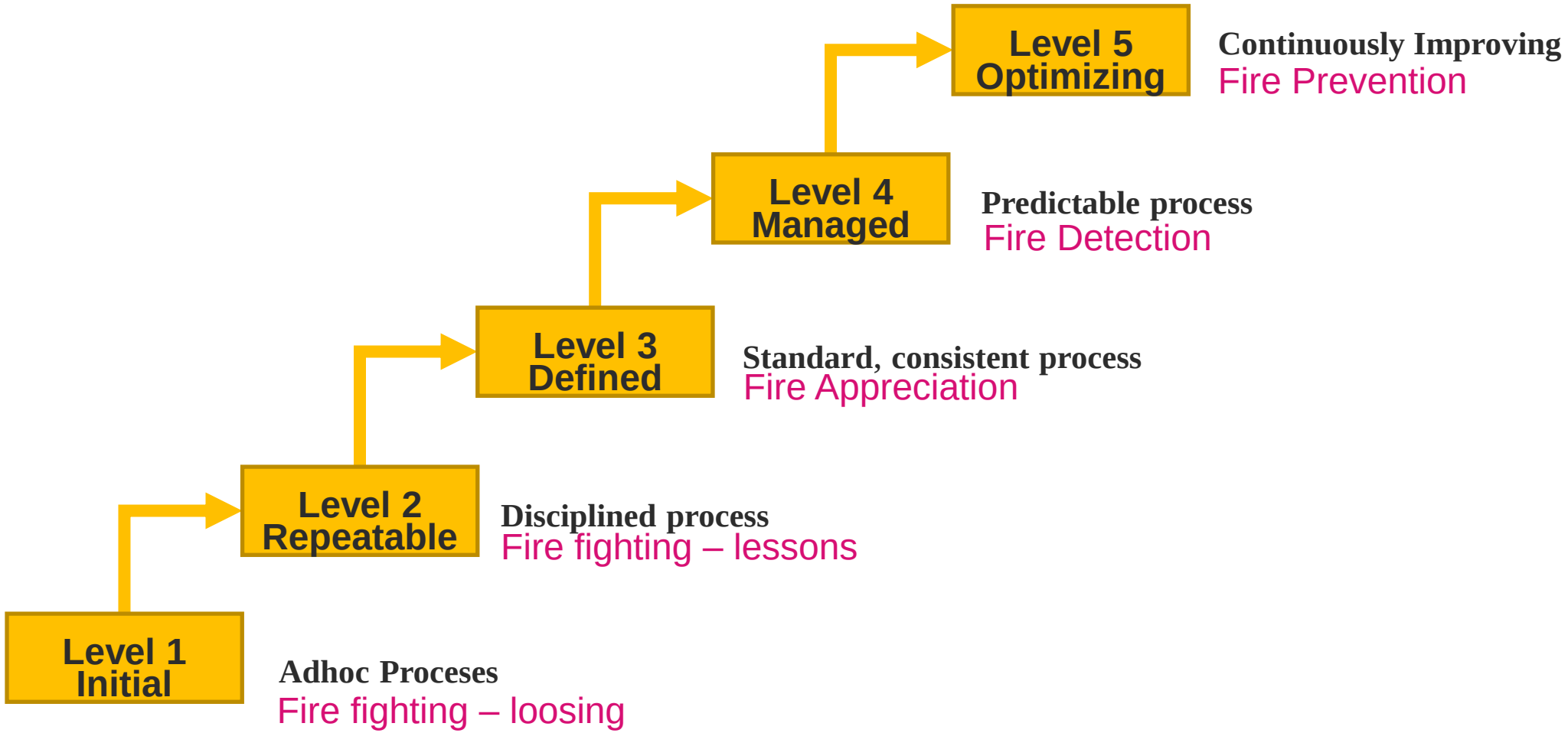
# Maturity Models IT

**Several maturity models  exists for IT and Business, Examples are….**

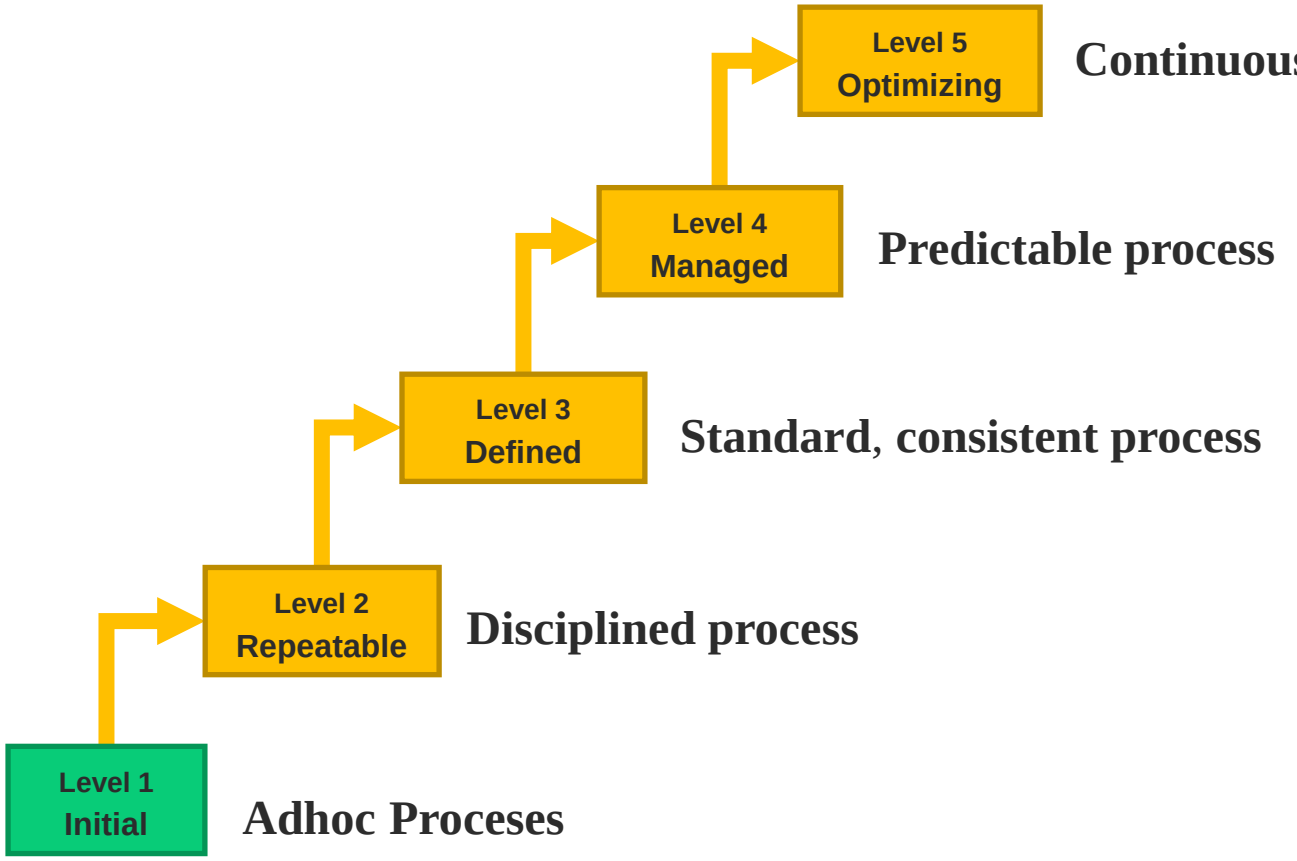
- CMMI
- ITIL Maturity Model
- Agile Maturity Model
- DevOps Maturity Model
- MDM Maturity Model

**We Focus on System Administration Maturity Model (SAMM) which is quite similar to ITIL Maturity Model.**

# SAMM Maturity Levels



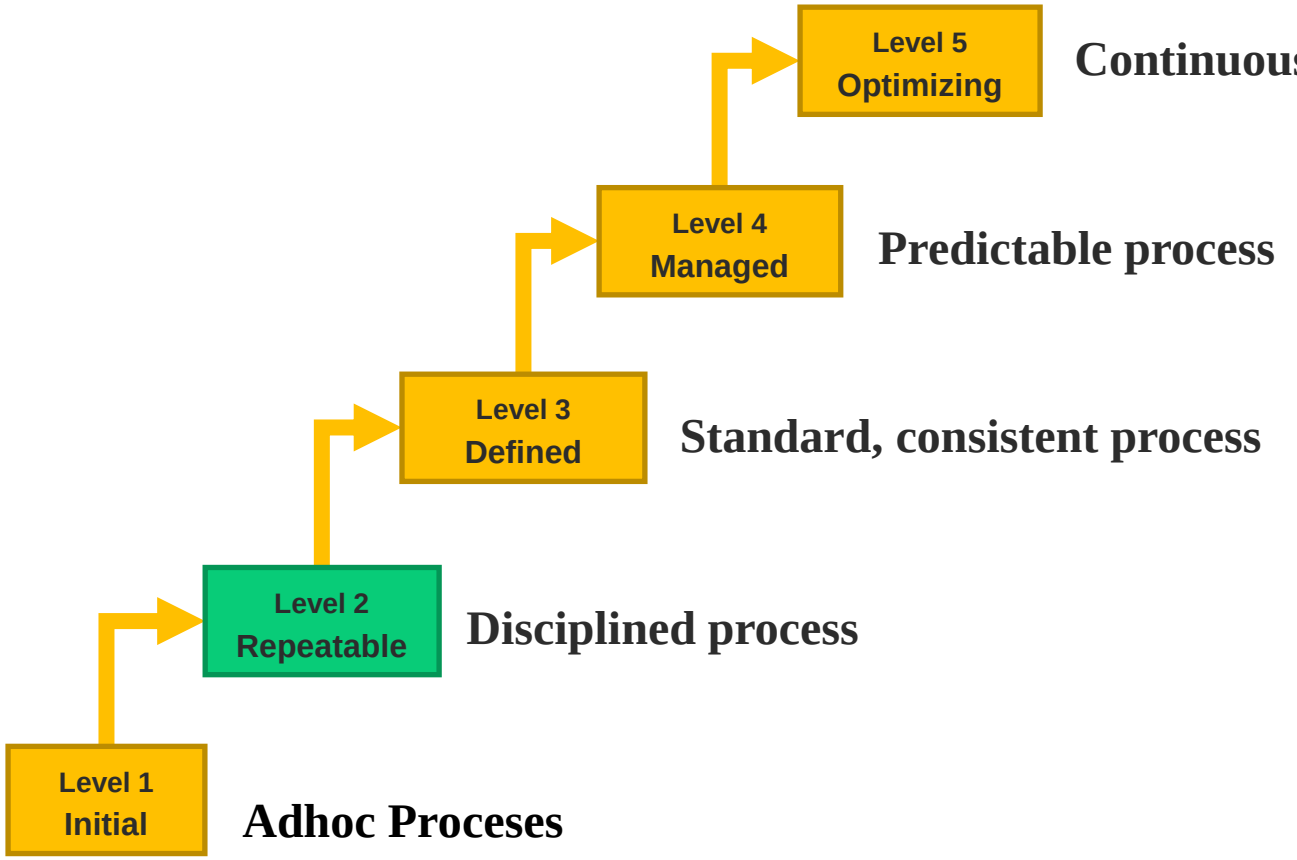
# SAMM Maturity Levels




**Key Process Area**

- The user is given the authority to decide what to do and when to do.
- They must also be able to assign work to other people creating interactions.

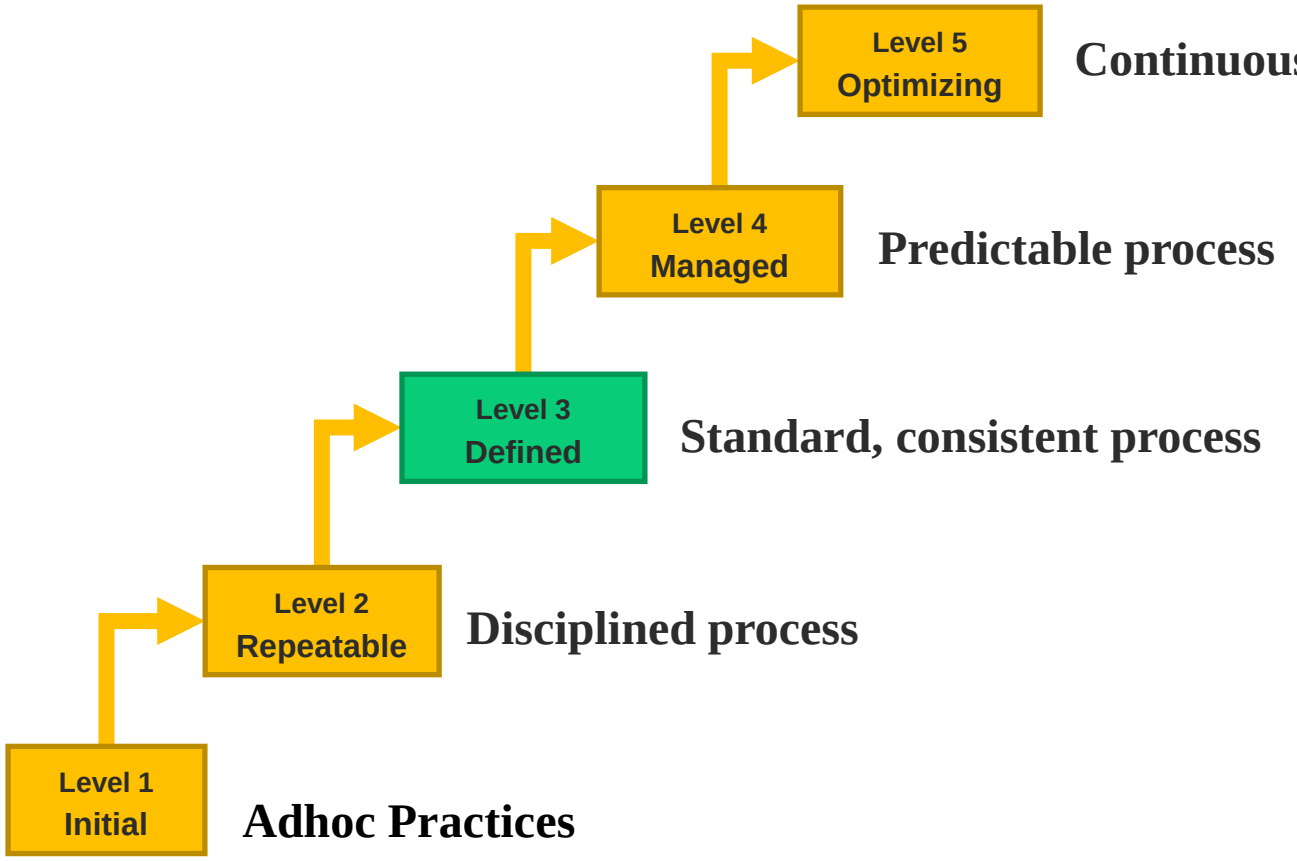
# SAMM Maturity Levels




**Key Process Area**

- Requirements Management
- Project Planning
- Project Tracking
- Subcontract/Vendor Management
- Quality Assurance
- Configuration Management

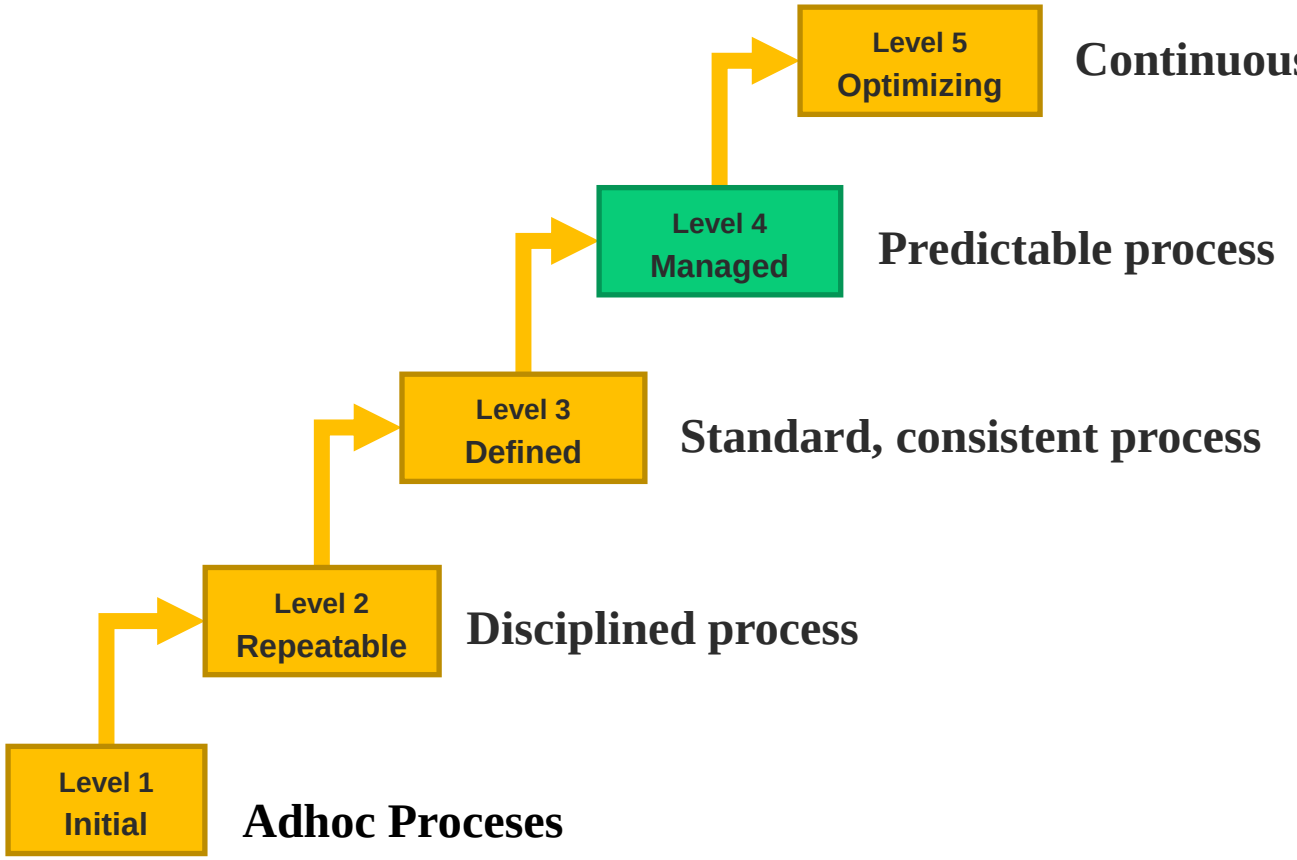
# SAMM Maturity Levels




**Key Process Area**

- Process Focus
- Process Definition
- Training
- Integrated Management
- System & Network Engineering
- Intergroup Coordination

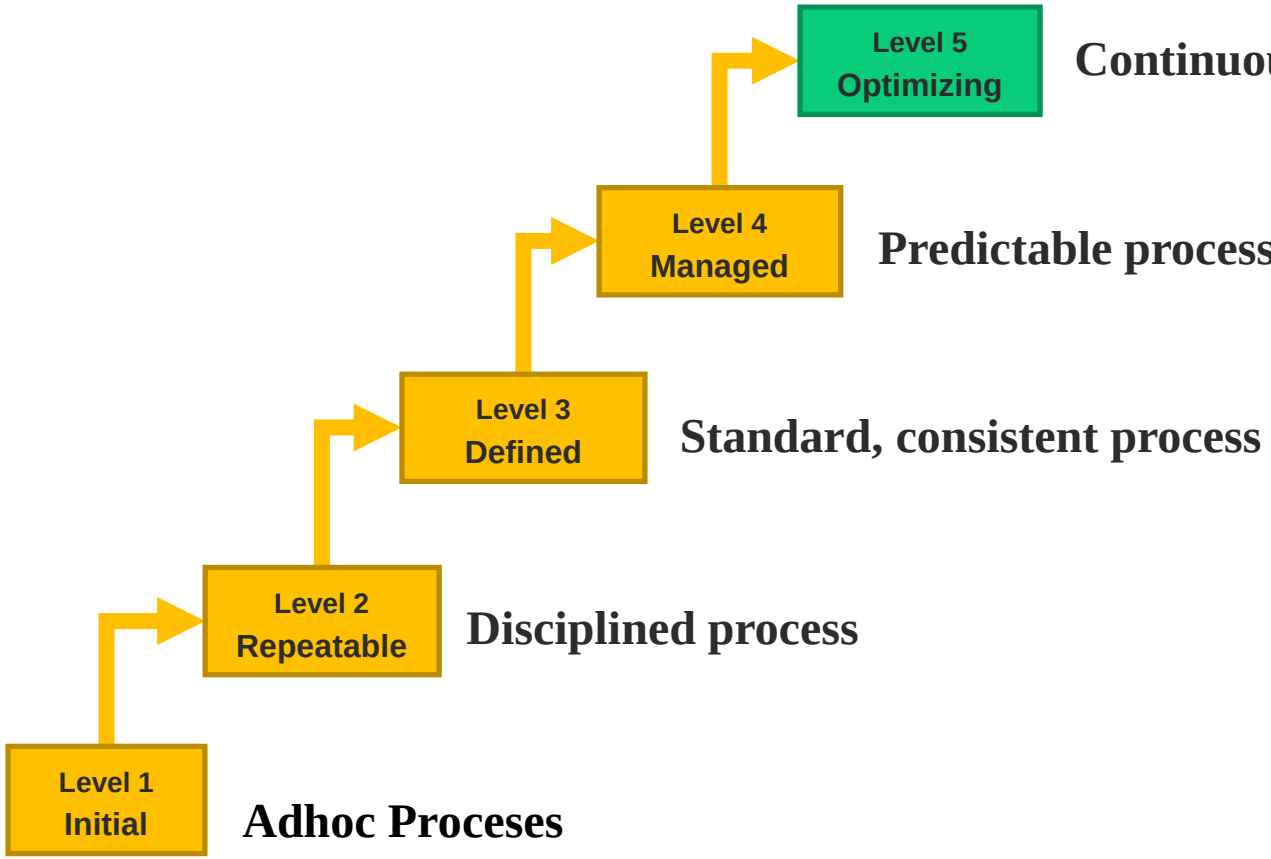
# SAMM Maturity Levels




**Key Process Area**

- Quantitative Process Management
- Quality Management

# Maturity Levels




**Key Process Area**

Defect Prevention

Technology Change Management

Process Change Management

IT601 – System and Network Administration

# SAMM – Common Practices

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Common Practices at different SAMM Levels

- Five Levels
  - Initial
  - Repeatable
  - Defined
  - Managed
  - Optimizing

Lets see the common Proactices of System Administration at above Levels.

# Level 1 – Initial - Practices

**New user** – Verbal requests are addressed as time permits or escalated with management.

**Software install** – New or upgraded software is installed whenever & where ever makes the most sense at the moment.

**Hardware install** – New or upgraded hardware is installed whenever & where ever makes the most sense at the moment.

**Problem report** – Problems are sometimes reported by users by mail or phone to a random administrator.

**Security** – No specific security standards or policies exist.

**Disk capacity** – Disk space is in short supply. No information on usage rates is available. Project managers of supported organizations often fight among themselves regarding disk space.

**Backups** – Backups are usually done according to a weekly schedule.

# Level 2 – Repeatable - Practices

**New user** – Procedure to request and create an account is well documented. Cycle time for requests is monitored.

**Software install** – Installation guidelines are understood. Time spent installing and configuring software is tracked.

**Hardware install** – Installation standards are understood. Time spent installing and configuring hardware is tracked.

**Problem report** – Process to report problems is well understood by users and cycle time for problem resolution is monitored.

**Security** – Various security standards are clearly documented. Security violations are monitored.

**Disk capacity** – Acceptable disk capacity levels are established. Capacity is periodically monitored.

**Backups** – Failure conditions for backups are understood. Failure rates and effort to resolve problems are tracked.

# Level 3 – Defined - Practices

**New user** – Head count expansion information is provided to signal new account requests are expected. Revision of procedure.

**Software install** – Installations are planned with the  supported organizations Reasons for install/upgrade are recorded.

**Hardware install** – Installations are coordinated with supported organizations and vendors. Reasons for install/upgrade are recorded.

**Problem report** – Problems are mapped to root causes. Group reviews solutions to resolutions suggested to address root causes.

**Security** – Group reviews security incidents for root vulnerabilities. Resolutions are discussed, tested and released.

**Disk capacity** – Capacity planning is addressed in project plans written by supported organizations and reviewed by the network & systems group.

**Backups** – Training is provided for network and systems group to enhance backup programs and procedures.

# Level 4 – Managed - Practices

**New user** – Cycle time numbers are used to adjust staffing to meet demand and requirements.

**Software install** – Productivity measures and goals for installation created.

**Hardware install** – Productivity/problem data is compare to goals for installs, tests and demos.

**Problem report** – Cycle time and resolution quality information is used on a regular basis to access effectiveness of problem reporting and resolution system.

**Security** – Effectiveness of group reviews is studied.

**Disk capacity** – Metric for disk availability is created and used.

**Backups** – Backup system is certified with high reliability rating.

# Level 5 – Optimizing - Practices

**New user** – Accounts are electronically requested and verified.

**Software install** – Problems with software installs are documented and avoidable with new procedures.

**Hardware install** – New hardware technologies are evaluated & integrated.

**Problem report** – New problem reporting system installed to better meet user requirements for ease of use.

**Security** – Internal contest to establish better security practices is established.

**Disk capacity** – Project tracking information combined with metrics of utilization are used to predict needs.

**Backups** – Backup process is revisited with input from supported organizations re: production schedules.

# SAMM Process Evaluation Ratings

| Rating | Score | Characterization |
|---|---|---|
| Poor | 0 | No ability No interest<br>Ineffective results |
| Weak | 2 | Partial ability Fragmented usage Inconsistent results |
| Fair | 4 | Implementation Plan defined Usage in major areas Consistent positive results |
| Marginal | 6 | Implementation across organization Usage in most areas<br>Positive measurable results |
| Qualified | 8 | Practice is integral part of process Consistent use across organization<br>Positive long term results |
| Outstanding | 10 | Excellence in practice well recognized Consistent long term use<br>Consistent world class results |

IT601 – System and Network Administration

# Building Reliable Systems

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Introduction to Reliability

- All systems will eventually fail.
  - Many Reasons

- An important term that quantifies dependability of a system during its life-cycle  is the reliability.

- Generally, Reliability is defined as the probability of success.

Hardware Failure

Process Failure`


Software Bug

Security Breach

Network Failure

# Objectives of Reliability

- The objective of studying reliability

  To quantify how “reliable” a product or service is,

  To understand the causes for poor reliability,

  To deploy actions to improve the reliability of Orgnizations.

- Unreliable Systems contributes

  Increase cost of ownership of the systems.

# Defining Reliability

- Reliability is defined as the probability that a system will perform its intended function(s) during a specified period of time under defined conditions.
  - Specified Time Period

  - Defined Conditions

# Examples of Reliability

Reliability usually changes as a function of time and is denoted as R(t).

Examples of reliability statements are:

The basic coverage warranty lasts for 36 months or 36,000 miles.

We warrant the bulb will be free from defects and will operate for 3 years based on 3 hours/day.

# Calculating Reliability

The Reliability is calculated …..

$$r(t) = \frac{N_w(t)}{N(t_0)}$$

$\mathrm{r}(T)$ Reliability at time T

$N_w(T)$ Working Systems at time T

$N(T_0)$ Total Systems at time T0

# Failure Probability

The formula for calculating reliability is:

$$f(t) = \frac{N_{nw}(t)}{N(t_0)}$$

$$= 1 - R(t)$$

$F(T($ Failure Probability at time T

$N_{nw}(T)$ Failed Systems at time T

$N(T_0)$ Total Systems at time T0

# Failure Rate

- The frequency of component failure per unit time. It is usually denoted by the λ.

$$\lambda(t) = \frac{N_w(t) - N_w(t+dt)}{dt * N(t)}$$

$\lambda(t)$ Failure rate at time t

dt interval

# Weibull distribution

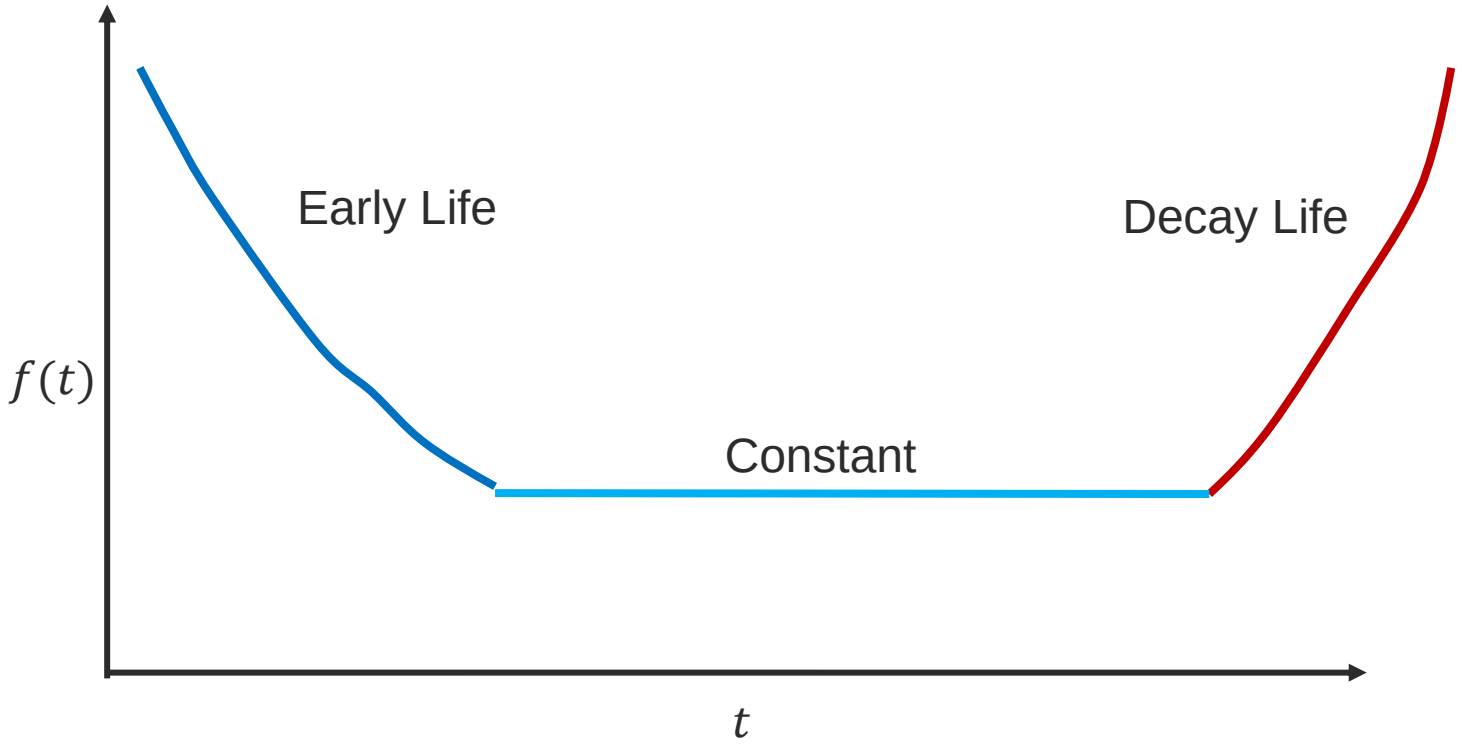
$$r(t) = e^{-\left(\frac{t}{\alpha}\right)^{\beta}}$$

$\alpha$ is called the scale factor which represents the characteristic life of the product

- which is the time at which 63% of the products have failed.

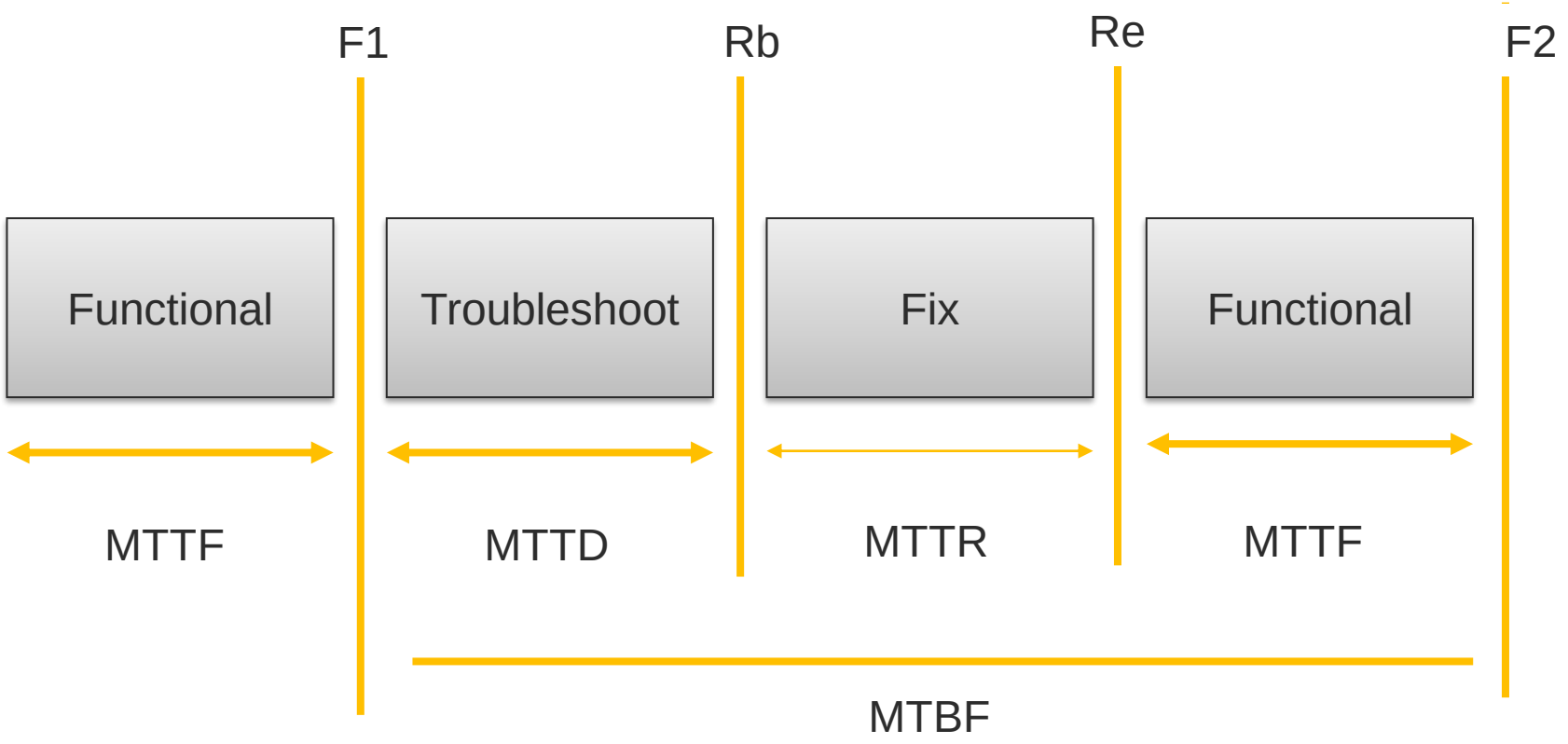
$\beta$ is called the shape factor and represents the different shapes for the Weibull distribution.

- $\beta < 1$ represents early failures
- $\beta = 1$ represents constant failure rate
- $\beta > 1$ represents wear-out failures



# System/Service Metrics

- MTTF
  - Mean time to failure
- MTTD
  - Mean time to detection
- MTTR
  - Mean time to recovery
- MTBF
  - Mean time between failure



# Failure Rate

- The frequency of component failure per unit time denoted by λ.
- Failure rate is considered as forecasted failure intensity given that the component is fully operational in its initial condition.

$$\lambda = \frac{1}{MTBF} = \frac{1}{MTTF}$$

# Repair rate

- The frequency of successful repair operations performed on a failed component per unit time and denoted μ .

- Repair rate is defined mathematically as follows:

$$\mu = \frac{1}{MTTR}$$

# Mean time to failure (MTTF)

The average time duration before a non-repairable system component fails.

$$MTTF = \frac{Total\ Functional\ Hours}{Total\ Units} = \frac{1}{\lambda}$$

# Mean time between failure (MTBF)

The average time duration between inherent failures of a repairable system component.

$$MTBF = \frac{Total\ Functional\ Hours}{Total\ Failures} = \frac{1}{\lambda}$$

$$MTBD = MTTF + MTTR$$

# Mean time to recovery (MTTR)

- The average time duration to fix a failed component and return to operational state.

- It includes the time spent during the alert and diagnostic process before repair activities are initiated.

$$MTTR = \frac{Total\ Maintenance\ Hours}{Total\ Repairs} = \frac{1}{\mu}$$

# Mean time to detection (MTTD)

The average time elapsed between the occurrence of a component failure and its detection.

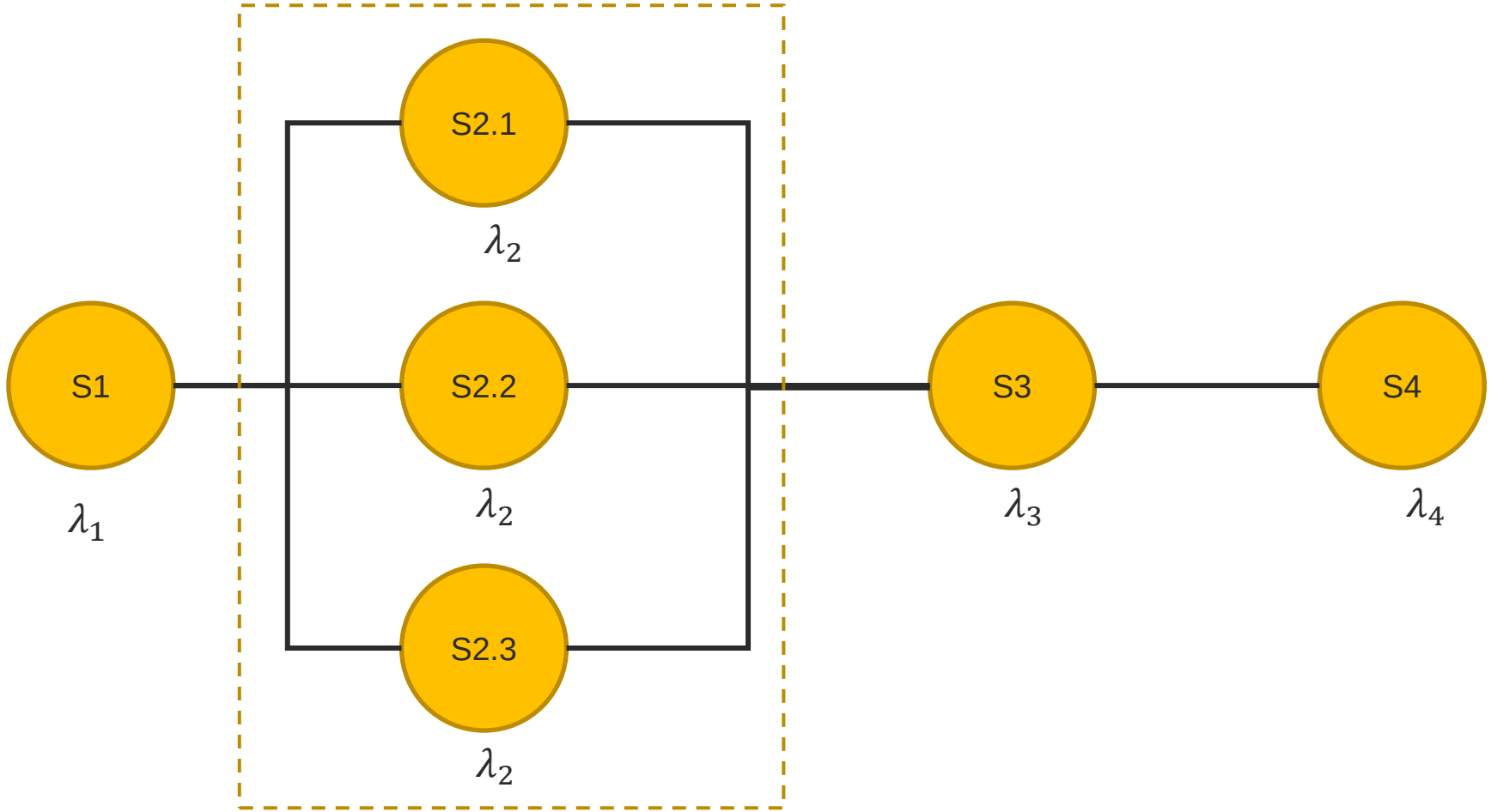
$$MTD = \frac{Total\ Hours\ of\ Incident\ Detection}{Total\ Incidents}$$

# Availability

It Determines the instantaneous performance of a component at any given time based on time duration between its failure and recovery

$$Availability = \frac{MTBF}{MTBF + MTTR}$$

# Reliability in Practical IT Systems



- For series connected components, the effective failure rate is determined as the sum of failure rates of each component. Given n series-connected components:

$$\lambda = \sum_{i=1}^{n} \lambda_i$$

- For parallel connected components, MTTF is determined as the reciprocal sum of failure rates of each system component. Given n parallel-connected components:

$$MTTF = \sum_{i=1}^{n} \frac{1}{\lambda_i}$$

- For hybrid systems, the connections may be reduced to series or parallel configurations first.

# Availability and Reliability specifications

- Calculate reliability and availability of each component individually. For series connected components, compute the product of all component values. for N series-connected components.

$$R(t) = \prod_{i=1}^{n} R_i(t) \quad , \quad A(t) = \prod_{i=1}^{n} A_i(t)$$

- For parallel connected components, for N parallel-connected components.

$$R(t) = 1 - \prod_{i=1}^{n} (1 - R_i(t)) \quad , \quad A(t) = 1 - \prod_{i=1}^{n} (1 - A_i(t))$$

- For hybrid connected components, reduce the calculations to series or parallel configurations first.

# Caveats

- It’s important to note a few caveats regarding these incident metrics and the associated reliability and availability calculations.
    - These metrics may be perceived in relative terms. Failure may be defined differently for the same components in different applications, use cases, and organizations.
    - The value of metrics such as MTTF, MTTR, MTBF, and MTTD are averages observed in experimentation under controlled or specific environments. These measurements may not hold consistently in real-world applications.
- Organizations should therefore map system reliability and availability calculations to business value and end-user experience.
- Decisions may require strategic trade-offs with cost, performance and, security, and decision makers will need to ask questions beyond the system dependability metrics and specifications followed by IT departments.

# Rferences


- https://www.bmc.com/blogs/system-reliability-availability-calculations/
- https://sigmamagic.com/blogs/introduction-to-reliability/

IT601 – System and Network Administration

# Common Tasks

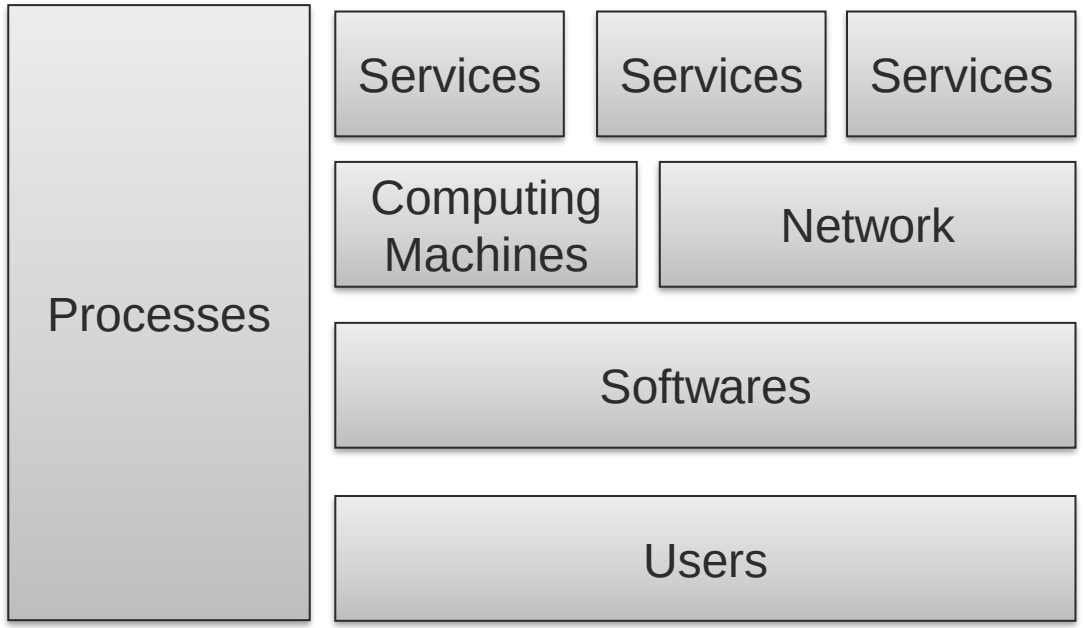
Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# IT System Components

1. Services
2. Computing Machines
3. Network
4. Processes
5. Softwares
6. Users



# Common Tasks

## Planning and Design

1) Architecuture
2) Physical Planning
3) IP Addressing
4) VLANS
5) Internet vs Intranet

## Routine Tasks

1) User Managment
2) Hardware Managment.
3) Backup Managment.
4) Software Managment.
5) Troubleshooting.
6) System monitoring.
7) Security Managment.
8) Users Assistance.
9) Communication.

# User Management

Adding User Accounts

- Consistency
- automation
- Startup (dot) files

Namespace management

- Usernames
- UIDs
- SSI
- Multiple namespaces

Removing user accounts

- Consistency
- automation
- Many accounts
- different systems

# 4.2 - Hardware Management

| Adding and removing hardware | Purchase | Capacity planning | Data Center management | Virtualization |
| --- | --- | --- | --- | --- |
| Configuration, cabling, etc. | Evaluate and purchase servers + other hardware | How many servers? How much bandwidth, storage? | Power, racks, environment (cooling, fire alarm) | When can virtual servers be used vs. physical? |

# 4.3 - Backups

## Backup strategy and policies

- Scheduling: when and how often?
- Capacity planning
- Location: on-site vs. off-site.

## Monitoring backups

- Checking logs
- Verifying media

## Performing restores when requested

- Time
- Automated?

# 4.4 - Software Installation

## Automated consistent OS installs

- Desktop vs. server OS image needs.

## Installation of software

- Purchase, find, or build custom software.
- Managing multiple versions of a software pkg.

## Managing software installations

- Distributing software to multiple hosts.

## Patching and updating software

- When?
- Updates vs Upgrades

# 4.5 - Troubleshooting

## Problem identification

- By user notification
- By log files or monitoring programs

## Tracking and visibility

- Ensure users know you're working on problem
- Provide an ETA if possible

## Finding the root cause of problems

- Provide temporary solution if necessary
- Solve the root problem to permanently eliminate

# 4.6 - System Monitoring

Automatically monitor systems for

- Problems (disk full, error logs, security)
- Performance (CPU, mem, disk, network)

Provides data for capacity planning

- Determine need for resources
- Establish case to bring to management

# 4.7 - Helping Users

## Request tracking system

- Ensures that you don’t forget problems.
- Ensures users know you’re working on their problem; reduces interruptions, status queries.
- Lets management know what you’ve done.

## User documentation and training

- Policies and procedures

## Schedule and communicate downtimes

- Effected Systems
- Donwtimes

# 4.8 - Communicate

## Customers

Keep customer appraised of process.

- When you've started working on a request with ETA.
- When you make progress, need feedback.
- When you're finished.

Communicate system status.

- Uptime, scheduled downtimes, failures.

Meet regularly with customer managers.

## Managers

Meet regularly with your manager.

Write weekly status reports.

IT601 – System and Network Administration

# Server Operating Systems

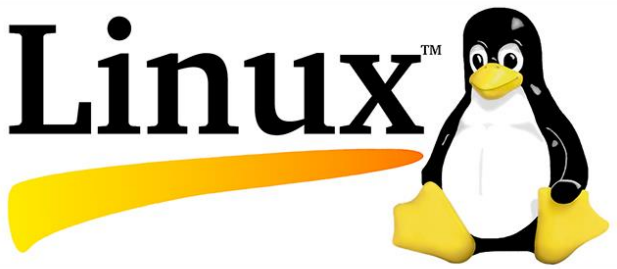
Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

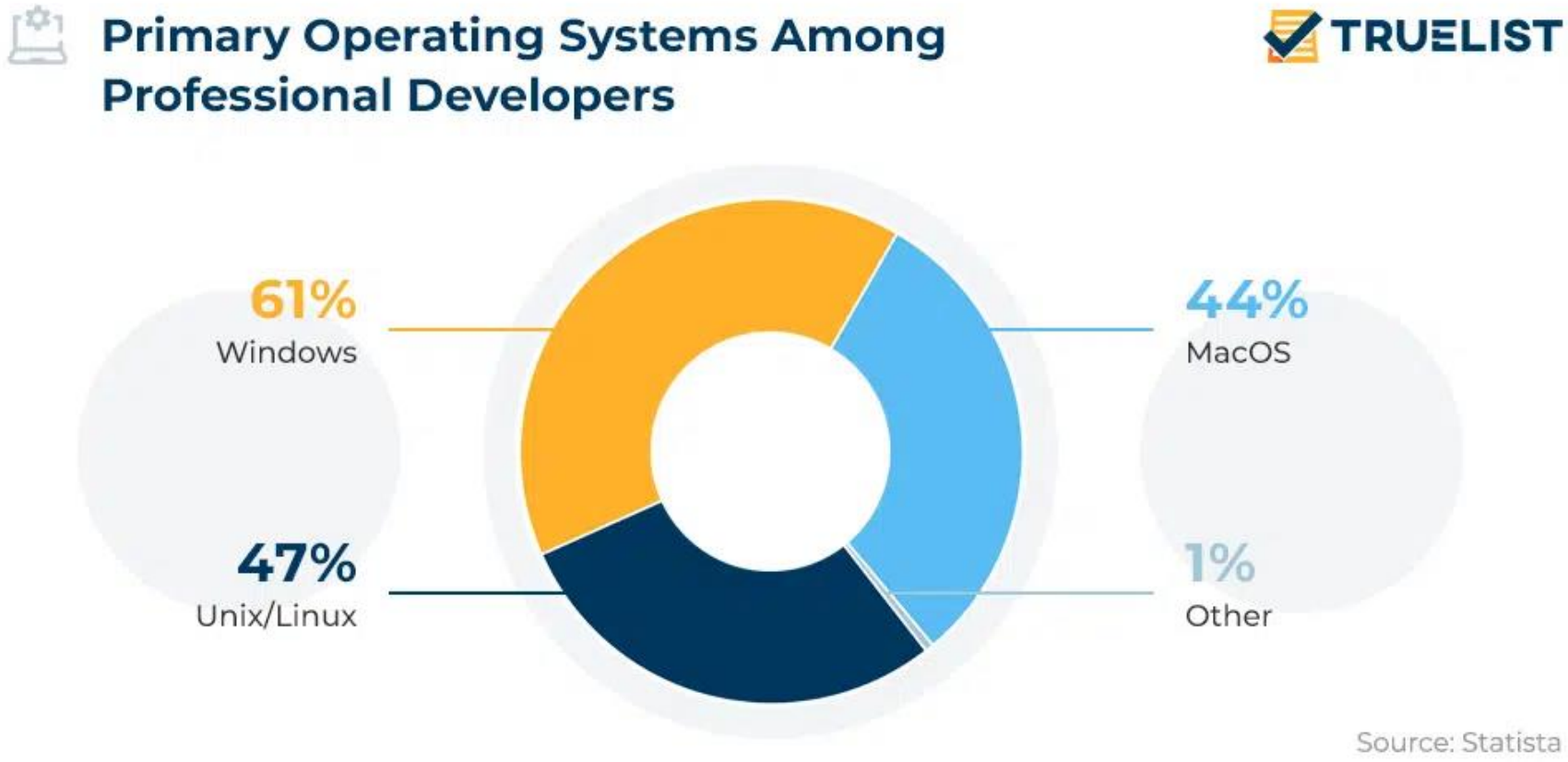
# Two Major Server Operating Systems

- Linux Operating Systems
  - Servers
  - Desktops
  - Mobiles

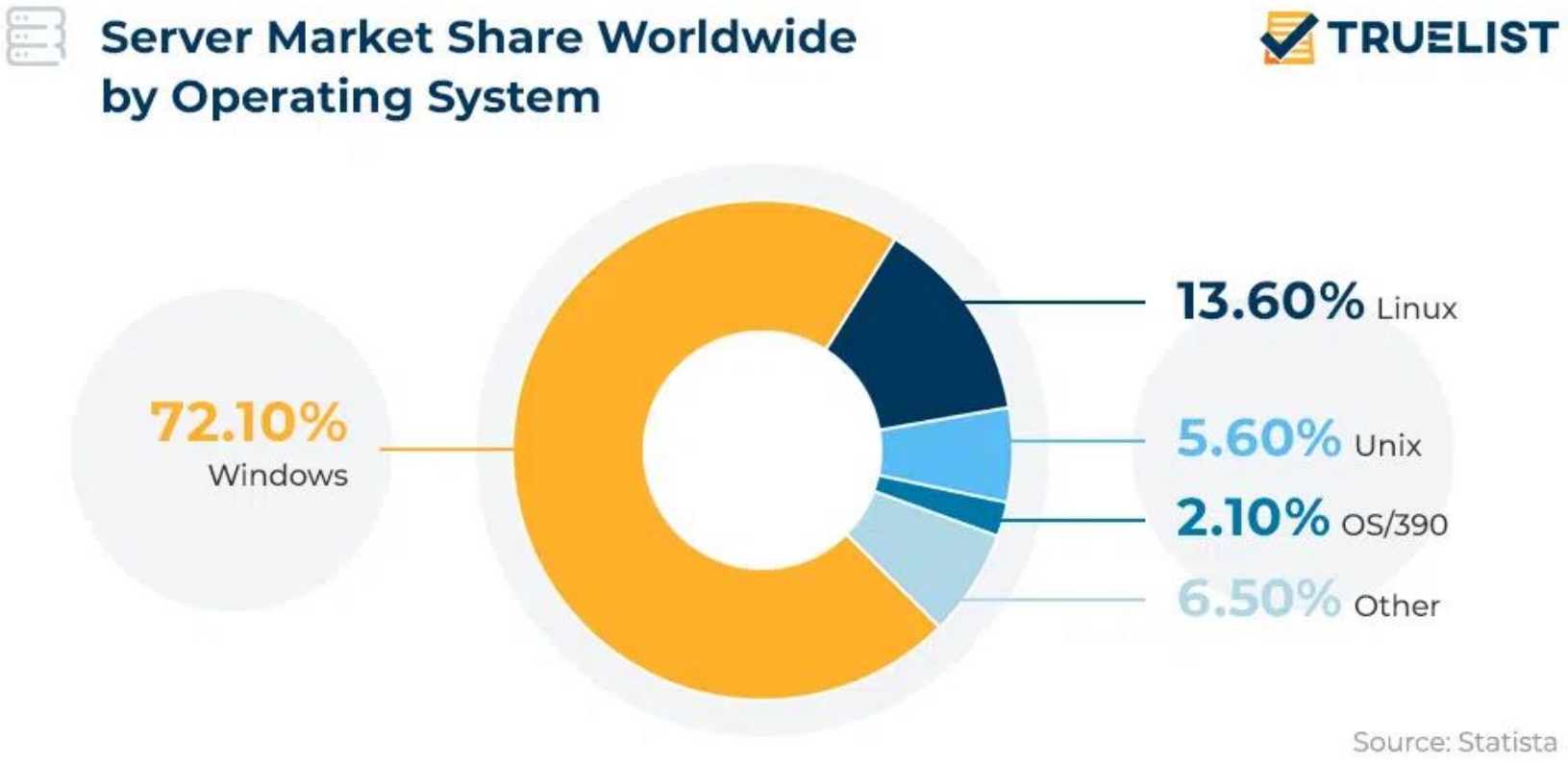



- Windows Operating Systems
  - Servers
  - Desktops
  - Mobiles

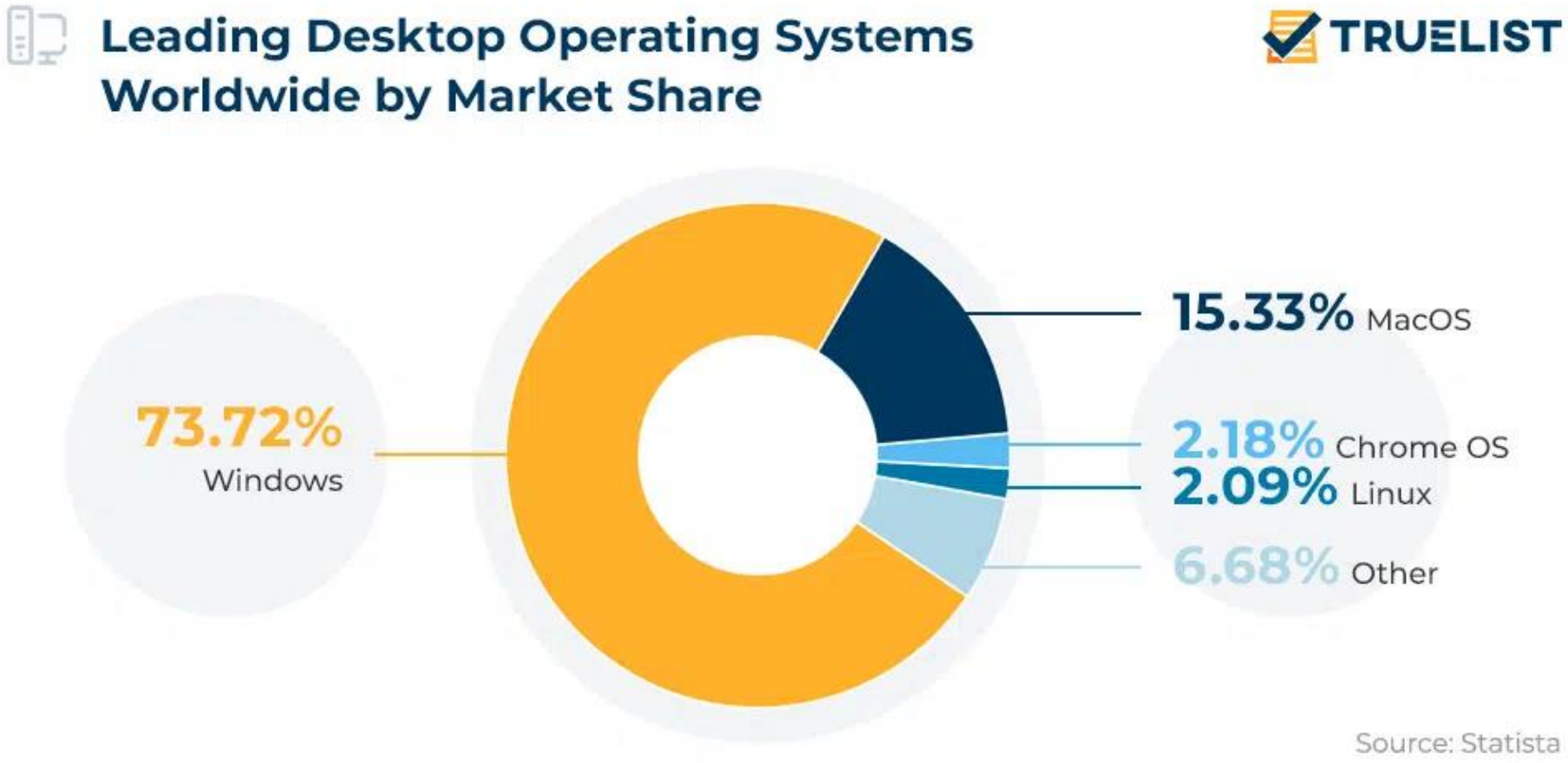
# Linux Choice on Developers



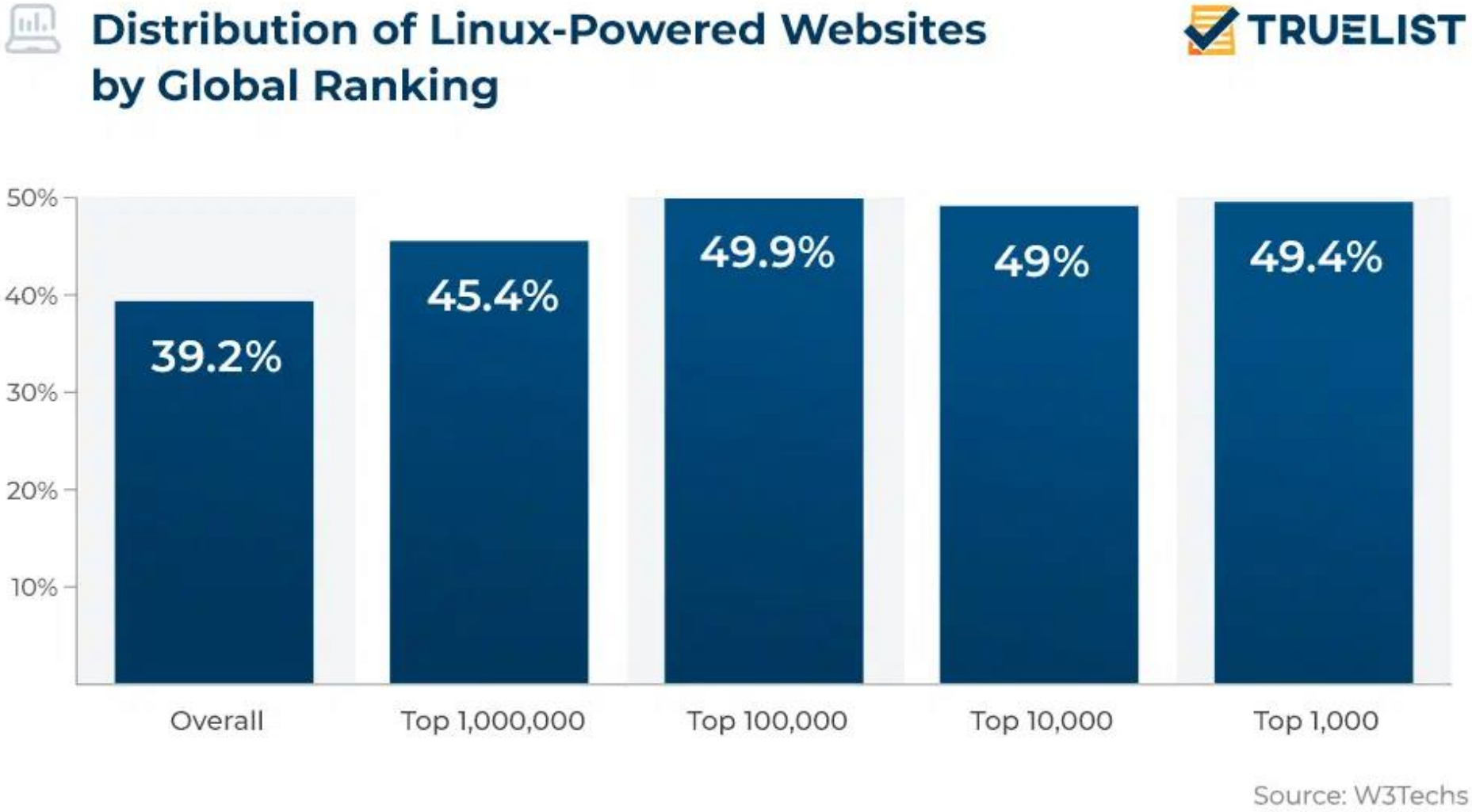
# Linux Usage Worldwide on Servers



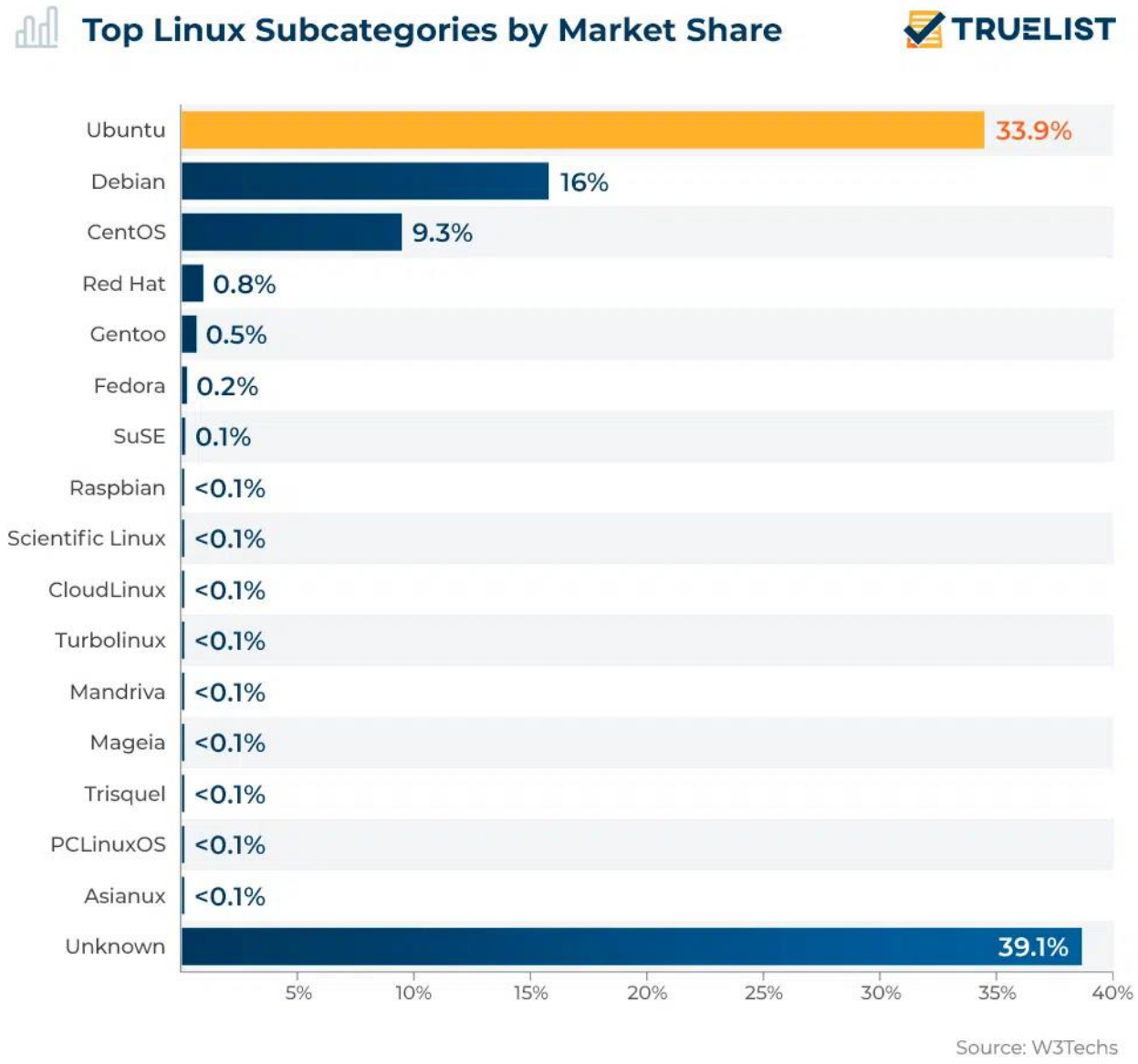
# Linux and Desktop Systems



# Linux Usage and Websites



# Usage of Linux Distributions



# Quick Comparison

| OS | Architecture | Cost | Security | Support |
|---|---|---|---|---|
| Linux | Centered around the Linux kernel | Free<br><br>open-source software | ▪ Highly secure against<br><br>malware<br>cyber threats | Large comm. supports<br><br>Answer commonly asked questions |
| Microsoft Windows Server System | ▪ Based on the Windows NT architecture | ▪ Owned by Microsoft,<br>▪ a licensing fee per user | ▪ More prone to<br>▪ hacking attempts<br>▪ cyber threats | ▪ Community and long-term customer support<br>▪ Great documentation |

# Quick Comparison

| OS | Operation | User Experience | Database Support | Scripting Support |
|---|---|---|---|---|
| Linux | command line | requires an relatively experienced Linux administrator | MySQL, PostgreSQL | Python<br>PHP,<br>Perl,<br>Other Unix languages |
| Microsoft Windows Server System | graphical user interface | more beginner-friendly | Microsoft SQL<br>Microsoft Access | ASP<br>ASP.NET |

# Advantages/Disadvantages of Linux Servers

- **Advantages**
  - No additional licensing fee as the operating system is free.
  - More reliable - it rarely experiences malware, cyber threats, or other security errors.
  - Not demanding on the client hardware and lower resource consumption.
  - Due to its low infrastructure requirements, it shows excellent performance rates.
  - System administrators have the freedom and opportunity to customize the system.
  - Seamless use of open-source software on the server.
  - Supports cooperative work without exposing the program's core.

- **Disadvantages**
  - Operating via a command line instead of a GUI requires some learning or experience.
  - Not all versions have long-term support.
  - Updating from one major version to another can sometimes be complex.
  - Some third-party and professional programs may not have support or require admin privileges.

# Advantages/Disadvantages of Windows Servers

- **Advantages**
  - Beginner-friendly due to its intuitive graphical user interface and out-of-the-box functionality.
  - Guaranteed five years of maintenance + five years of extended support.
  - Supports third-party applications and is compatible with Microsoft applications.
  - Requires less admin monitoring and maintenance thanks to its robust approach and automated updates.

- **Disadvantages**
  - Higher costs due to the obligatory licensing fee for the OS.
  - More prone to malware, cyber-threats, and other security-related errors.
  - Its mandatory GUI makes it more resource intensive.

IT601 – System and Network Administration

# Linux Server

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# What is Ubuntu Server?

- Ubuntu Server is a server operating system developed by Canonical that runs on all major architectures


- Ubuntu is a server platform that anyone can use for the following and much more:


- Ubuntu Server has these minimum requirements:

| Install Type | Install Method | CPU | RAM | Hard Drive Space | |
|---|---|---|---|---|---|
| | | | | Base System | All Tasks Installed |
| Server (Standard) | debian-installer | 1 gigahertz | 512 megabytes | 1.5 gigabyte | 2.5 gigabytes |
| | live server | 1 gigahertz (amd64 only) | 1 gigabyte | 1.5 gigabyte | n/a |
| Server (Minimal) | debian-installer | 300 megahertz | 384 megabytes | 1.5 gigabytes | 2.5 gigabytes |

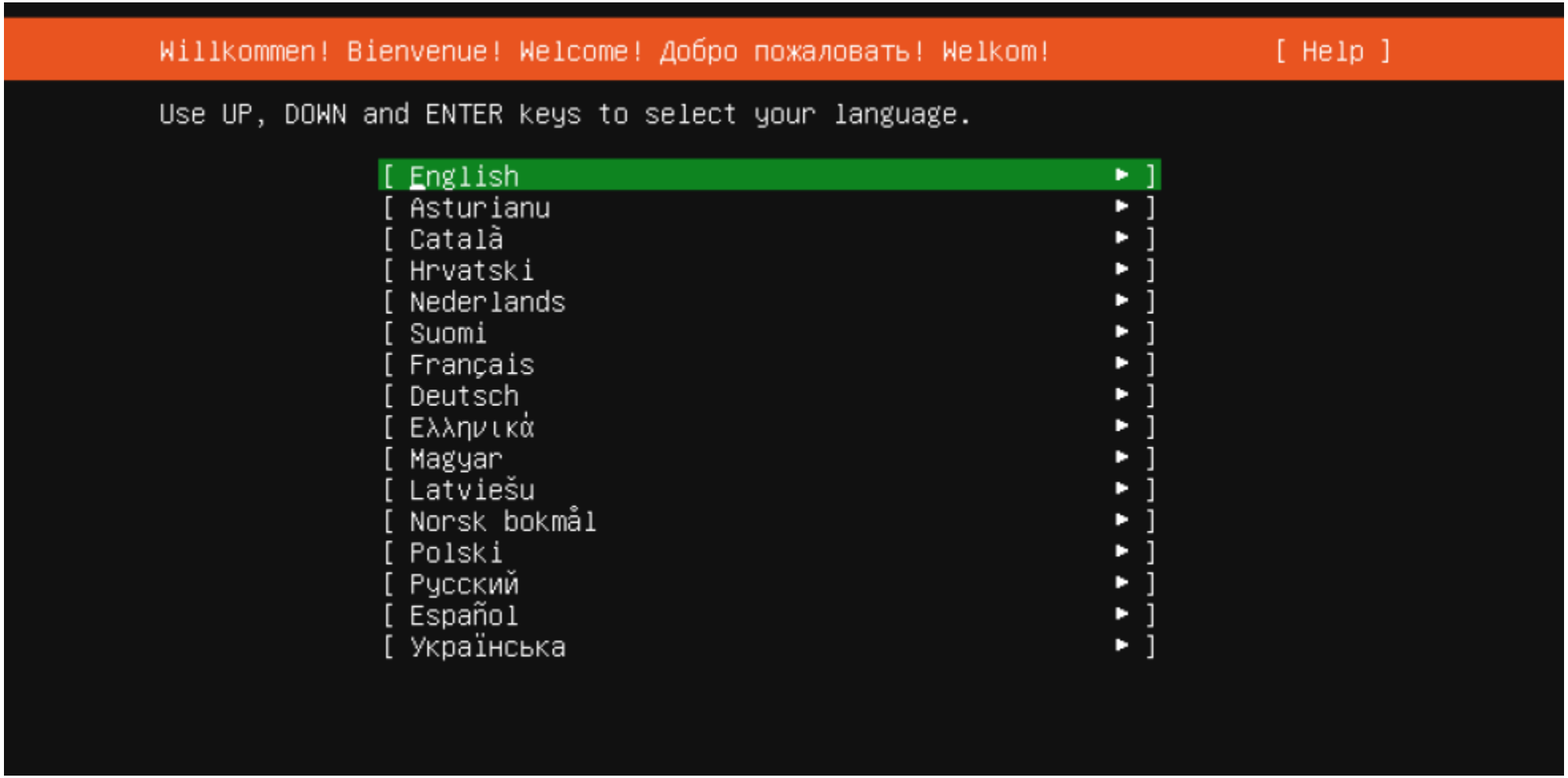
# Differences from Ubuntu Desktop

- There are a few differences between the Ubuntu Server Edition and the Ubuntu Desktop Edition.
- It should be noted that both editions use the same apt repositories, making it just as easy to install a server application on the Desktop Edition as it is on the Server Edition.
- The differences between the two editions are the lack of an X window environment in the Server Edition and the installation process.
- Ubuntu version 10.10 and prior, actually had different kernels for the server and desktop editions. Ubuntu no longer has separate -server and -generic kernel flavors.
- These have been merged into a single -generic kernel flavor to help reduce the maintenance burden over the life of the release.
- When running a 64-bit version of Ubuntu on 64-bit processors you are not limited by memory
- addressing space.
- To see all kernel configuration options you can look through /boot/config-4.14.0-server. Also, Linux Kernel in a Nutshell3 is a great resource on the options available.

# Installing Ubuntu Server

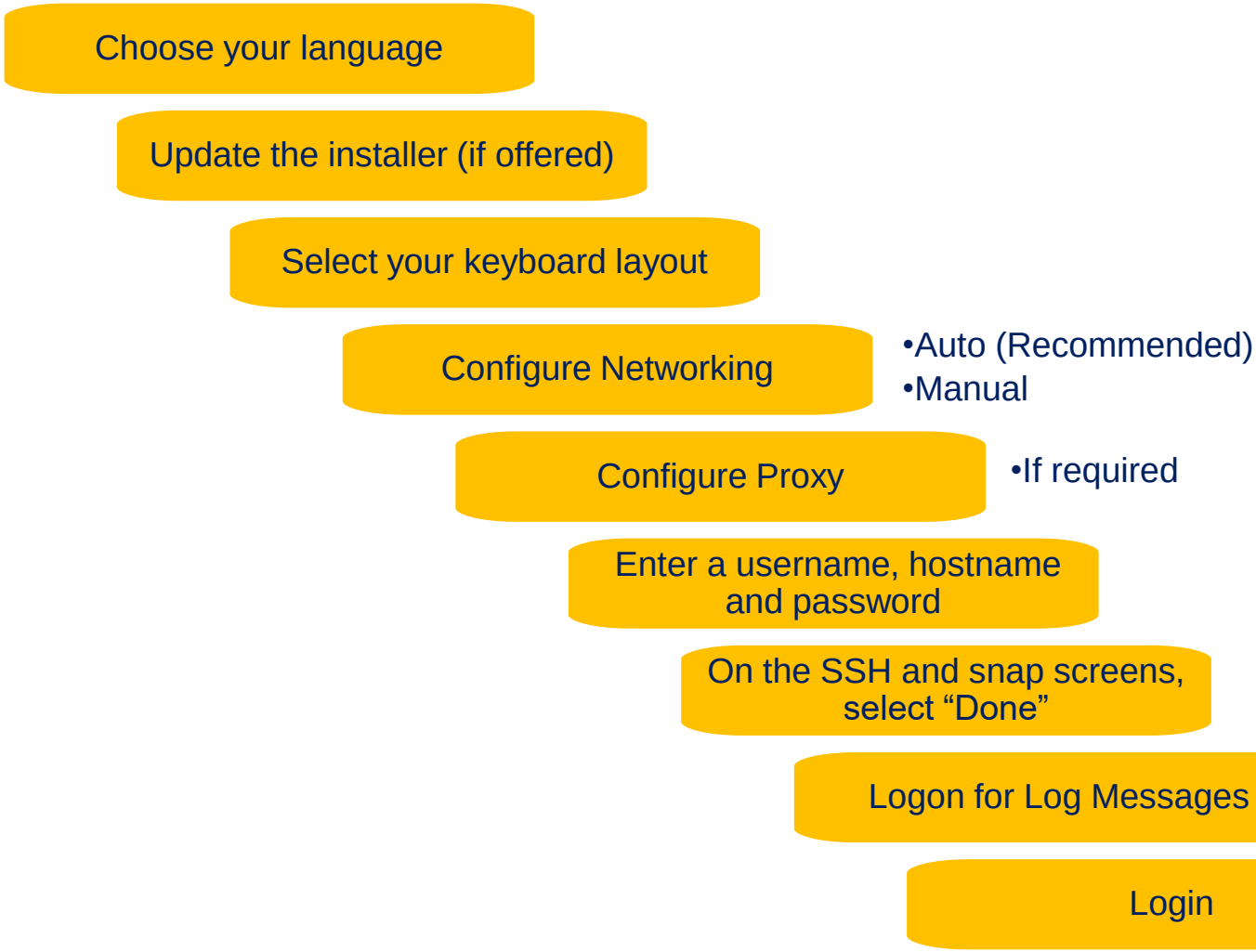
- Preparing install media
  - There are platform specific step-by-step examples for ppc64el installations.
  - For amd64, download the install image from https://releases.ubuntu.com/20.04/.
  - There are many ways to boot the installer but the simplest and most common way is to create a bootable USB stick to boot the system to be installed.
- Booting the installer
  - Plug the USB stick into the system to be installed and start it.
  - Most computers will automatically boot from USB or DVD, though in some cases this is disabled to improve boot times. If you don’t see the boot message and the “Welcome” screen which should appear after it, you will need to set your computer to boot from the install media.
  - There should be an on-screen message when the computer starts telling you what key to press for settings or a boot menu.
  - Depending on the manufacturer, this could be Escape, F2, F10 or F12. Simply restart your computer and hold down this key until the boot menu appears, then select the drive with the Ubuntu install media.

# Example of Installer



# Installation Steps

- The installer is designed to be easy to use and have sensible defaults so for a first install you can mostly just accept the defaults for the most straightforward install:



IT601 – System and Network Administration

# Software Package Management

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Software Package Management

- **Package management system is derived from the same system used by the Debian GNU/Linux Distribution.**
- **Package Files**
  - The package files contain all of the necessary files, meta-data, and instructions to implement a particular functionality or software application on your Ubuntu computer.
  - Debian package files typically have the extension '.deb', and usually exist in repositories which are collections of packages found on various media, such as CD-ROM discs, or online.
  - Packages are normally in a precompiled binary format; thus installation is quick, and requires no compiling of software.
- **Dependencies**
  - Many complex packages use dependencies.
  - Dependencies are additional packages required by the principal package in order to function properly.
    - For example, the speech synthesis package festival depends upon the package libasound2, which is a package supplying the ALSA sound library needed for audio playback. In order for festival to function, it and all of its dependencies must be installed.
  - The software management tools in Ubuntu will do this automatically.

# dpkg

- **dpkg is a package manager for Debian-based systems.**
- **It can install, remove, and build packages, but unlike other package management systems, it cannot automatically download and install packages or their dependencies.**
- **Using dpkg  to manage software**
  - List all packages installed on the system, from a terminal prompt type:

    *dpkg -l*
  - Depending on the amount of packages on your system, this can generate a large amount of output. Pipe the output through grep to see if a specific package is installed:

    *dpkg -l | grep apache2*
  - To list the files installed by a package, in this case the ufw package, enter:

    *dpkg -L ufw*
  - If you are not sure which package installed a file, dpkg -S may be able to tell you. For example:

    dpkg -S /etc/host.conf
    base-files: /etc/host.conf

# dpkg

- You can install a local .deb file by entering:

  sudo dpkg -i zip_3.0-4_i386.deb

- Uninstalling a package can be accomplished by:

  *sudo dpkg -r zip*

- For more dpkg options see the man page:

  *man dpkg*

# APT

- **The apt command is a powerful command-line tool, which works with Ubuntu's Advanced Packaging Tool (APT) performing such functions as**
  - installation of new software packages
  - upgrade of existing software packages
  - updating of the package list index
  - upgrading the entire Ubuntu system

- **Advantages**
  - Being a simple command-line tool, apt has numerous advantages over other package management tools available in Ubuntu for server administrators.
  - Ease of use over simple terminal connections (SSH)
  - The ability to be used in system administration scripts
  - Which can in turn be automated by the cron scheduling utility.

# Using APT

- **Installing a Package**

  *sudo apt install nmap*

- **Removing a Package**

  *sudo apt remove nmap*

  Also, adding the --purge option to apt remove will remove the package configuration files as well. This may or may not be the desired effect, so use with caution.

- **Updating the Package Index**

  The APT package index is essentially a database of available packages from the repositories defined in the /etc/apt/sources.list file and in the /etc/apt/sources.list.d directory. To update the local package index with the latest changes made in the repositories, type the following:

  *sudo apt update*

- **Upgrade Packages**

  Over time, updated versions of packages currently installed on your computer may become available from the package repositories (for example security updates). To upgrade your system, first update your package index as outlined above, and then type:

  *sudo apt upgrade*

- **Actions of the apt command, such as installation and removal of packages, are logged in the /var/log/dpkg.log log file.**

# Aptitude

- Launching Aptitude with no command-line options, will give you a menu-driven, text-based front-end to the Advanced Packaging Tool (APT) system.
- Many of the common package management functions, such as installation, removal, and upgrade, can be performed in Aptitude with single-key commands, which are typically lowercase letters.
- Aptitude is best suited for use in a non-graphical terminal environment to ensure proper functioning of the command keys.
- You may start the menu-driven interface of Aptitude as a normal user by typing the following command at a terminal prompt:

  *sudo aptitude*

- Actions
  - Install Packages
  - Remove Packages
  - Update Package Index
  - Upgrade Packages

# Aptitude

- The first column of information displayed in the package list in the top pane, when actually viewing packages lists the current state of the package, and uses the following key to describe the state of the package:
  - i: Installed package
  - c: Package not installed, but package configuration remains on system
  - p: Purged from system
  - v: Virtual package
  - B: Broken package
  - u: Unpacked files, but package not yet configured
  - C: Half-configured - Configuration failed and requires fix
  - H: Half-installed - Removal failed and requires fix
- To exit Aptitude, simply press the q key and confirm you wish to exit. Many other functions are available from the Aptitude menu by pressing the F10 key.
- Command Line Aptitude

  *sudo aptitude install nmap*

  *sudo aptitude remove nmap*

# Automatic Updates

- The unattended-upgrades package can be used to automatically install updated packages, and can be configured to update all packages or just install security updates.
- Install the package by entering the following in a terminal:

  *sudo apt install unattended-upgrades*

- To configure unattended-upgrades, edit /etc/apt/apt.conf.d/50unattended-upgrades and adjust the following to fit your needs:

```
Unattended-Upgrade::Allowed-Origins {
"${distro_id}:${distro_codename}";
"${distro_id}:${distro_codename}-security";
// "${distro_id}:${distro_codename}-updates";
// "${distro_id}:${distro_codename}-proposed";
// "${distro_id}:${distro_codename}-backports";
};
```

- Certain packages can also be blacklisted and therefore will not be automatically updated. To blacklist a package, add it to the list:

```
Unattended-Upgrade::Package-Blacklist {
// "vim";
// "libc6";
// "libc6-dev";
// "libc6-i686";
};
```

# Automatic Updates

- To enable automatic updates, edit /etc/apt/apt.conf.d/20auto-upgrades and set the appropriate apt configuration options:

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::AutocleanInterval "7";
APT::Periodic::Unattended-Upgrade "1";
```

- The above configuration updates the package list, downloads, and installs available upgrades every day. The local download archive is cleaned every week

- On servers upgraded to newer versions of Ubuntu, depending on your responses, the file listed above may not be there. In this case, creating a new file of this name should also work.

# Automatic Updates

- Notifications

  Configuring Unattended-Upgrade::Mail in /etc/apt/apt.conf.d/50unattended-upgrades will enable unattended-upgrades to email an administrator detailing any packages that need upgrading or have problems.

  Another useful package is apticron. apticron will configure a cron job to email an administrator information about any packages on the system that have updates available, as well as a summary of changes in each package.

  To install the apticron package, in a terminal enter:

  *sudo apt install apticron*

  Once the package is installed edit /etc/apticron/apticron.conf, to set the email address and other options:

  EMAIL="root@example.com"

# Configuration of APT

- Configuration of the Advanced Packaging Tool (APT) system repositories is stored in the /etc/apt/sources.list file and the /etc/apt/sources.list.d directory.

- You may edit the file to enable repositories or disable them. For example, to disable the requirement of inserting the Ubuntu CD-ROM whenever package operations occur, simply comment out the appropriate line for the CD-ROM, which appears at the top of the file:

```
# no more prompting for CD-ROM please
# deb cdrom:[Ubuntu 18.04 _Bionic Beaver_ - Release i386 (20111013.1)]/ bionic main restricted
```

- Extra Repositories
  - By default, the Universe and Multiverse repositories are enabled but if you would like to disable them edit /etc/apt/sources.list and comment the following lines:

```
deb http://archive.ubuntu.com/ubuntu bionic universe multiverse
deb-src http://archive.ubuntu.com/ubuntu bionic universe multiverse
deb http://us.archive.ubuntu.com/ubuntu/ bionic universe
deb-src http://us.archive.ubuntu.com/ubuntu/ bionic universe
deb http://us.archive.ubuntu.com/ubuntu/ bionic-updates universe
deb-src http://us.archive.ubuntu.com/ubuntu/ bionic-updates universe
deb http://us.archive.ubuntu.com/ubuntu/ bionic multiverse
```

```
deb-src http://us.archive.ubuntu.com/ubuntu/ bionic multiverse
deb http://us.archive.ubuntu.com/ubuntu/ bionic-updates multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ bionic-updates multiverse
deb http://security.ubuntu.com/ubuntu bionic-security universe
deb-src http://security.ubuntu.com/ubuntu bionic-security universe
deb http://security.ubuntu.com/ubuntu bionic-security multiverse
deb-src http://security.ubuntu.com/ubuntu bionic-security multiverse
```

IT601 – System and Network Administration

# Networking – Interfaces and Addressing

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Network Management

- Ubuntu ships with a number of graphical utilities to configure your network devices.
- The network management includes
  - Configuring Interfaces
  - IP Addressing
  - DNS
  - Bridging

# Ethernet Interfaces

- Ethernet interfaces are identified by the system using predictable network interface names.
- These names can appear as eno1 or enp0s25.
- In some cases an interface may still use the kernel eth# style of naming.
- Identify Ethernet Interfaces

  To identify all available Ethernet interfaces, you can use following commands as shown below.

  ip a

  quickly identify all available Ethernet interfaces

  sudo lshw -class network

  provides greater details around the hardware capabilities of specific adapters.

# Ethernet Interfaces

*ip a*

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever

2: enp0s25: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 00:16:3e:e2:52:42 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.102.66.200/24 brd 10.102.66.255 scope global dynamic eth0
       valid_lft 3257sec preferred_lft 3257sec
    inet6 fe80::216:3eff:fee2:5242/64 scope link
       valid_lft forever preferred_lft forever
```

# Ethernet Interfaces

*sudo lshw -class network*

```
*-network
    description: Ethernet interface
    product: MT26448 [ConnectX EN 10GigE, PCIe 2.0 5GT/s]
    vendor: Mellanox Technologies
    physical id: 0
    bus info: pci@0004:01:00.0
    logical name: eth4
    version: b0
    serial: e4:1d:2d:67:83:56
    slot: U78CB.001.WZS09KB-P1-C6-T1
    size: 10Gbit/s
    capacity: 10Gbit/s
    width: 64 bits
    clock: 33MHz
    capabilities: pm vpd msix pciexpress bus_master cap_list ethernet physical fibre 10000bt-fd
    configuration: autonegotiation=off broadcast=yes driver=mlx4_en driverversion=4.0-0
duplex=full firmware=2.9.1326 ip=192.168.1.1 latency=0 link=yes multicast=yes port=fibre
speed=10Gbit/s
    resources: iomemory:24000-23fff irq:481 memory:3fe200000000-3fe2000fffff
memory:240000000000-240007ffffff
```

# Ethernet Interface Logical Names

- Interface logical names can also be configured via a netplan configuration. If you would like control which interface receives a particular logical name use the match and set-name keys.

- The match key is used to find an adapter based on some criteria like MAC address, driver, etc. Then the set-name key can be used to change the device to the desired logial name.

```
network:
    version: 2
    renderer: networkd
    ethernets:
        eth_lan0:
            dhcp4: true
        match:
            macaddress: 00:11:22:33:44:55
    set-name: eth_lan0
```

# Ethernet Interface Settings

- **ethtool** is a program that can display and change Ethernet card settings such as
  - auto-negotiation,
  - port speed,
  - duplex mode,
  - and Wake-on-LAN.

- The following is an example of how to view supported features and configured settings of an Ethernet interface.

sudo ethtool eth4

```
Settings for eth4:
        Supported ports: [ FIBRE ]
        Supported link modes: 10000baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: No
        Supported FEC modes: Not reported
        Advertised link modes: 10000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: No
        Advertised FEC modes: Not reported
        Speed: 10000Mb/s
        Duplex: Full
        Port: FIBRE
        PHYAD: 0
        Transceiver: internal
        Auto-negotiation: off
        Supports Wake-on: d
        Wake-on: d
        Current message level: 0x00000014 (20)
                link ifdown
        Link detected: yes
```

# IP Addressing

- Temporary IP Address Assignment
- Dynamic IP Address Assignment (DHCP Client)
- Static IP Address Assignment
- Loopback Interface

# Temporary IP Address Assignment

- For temporary network configurations, you can use the ip command which is also found on most other GNU/Linux operating systems.
- The ip command allows you to configure settings which take effect immediately, however they are not persistent and will be lost after a reboot.
- To temporarily configure an IP address, you can use the ip command in the following manner. Modify the IP address and subnet mask to match your network requirements.

```
sudo ip addr add 10.102.66.200/24 dev enp0s25
```

- The ip can then be used to set the link up or down.

```
ip link set dev enp0s25 up
ip link set dev enp0s25 down
```

# Temporary IP Address Assignment

To verify the IP address configuration of enp0s25, you can use the ip command in the following manner.

ip address show dev enp0s25

```
10: enp0s25: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
 qlen 1000
    link/ether 00:16:3e:e2:52:42 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.102.66.200/24 brd 10.102.66.255 scope global dynamic eth0
       valid_lft 2857sec preferred_lft 2857sec
    inet6 fe80::216:3eff:fee2:5242/64 scope link
       valid_lft forever preferred_lft forever6
```

To configure a default gateway, you can use the ip command in the following manner. Modify the default gateway address to match your network requirements.

sudo ip route add default via 10.102.66.1

# Temporary IP Address Assignment

- To verify your default gateway configuration, you can use the ip command in the following manner.

```
ip route show
```

```
default via 10.102.66.1 dev eth0 proto dhcp src 10.102.66.200 metric 100
10.102.66.0/24 dev eth0 proto kernel scope link src 10.102.66.200
10.102.66.1 dev eth0 proto dhcp scope link src 10.102.66.200 metric 100
```

- If you require DNS for your temporary network configuration, you can add DNS server IP addresses in the file /etc/resolv.conf.
- In general, editing /etc/resolv.conf directly is not recommanded, but this is a temporary and non-persistent configuration.
- The example below shows how to enter two DNS servers to /etc/resolv.conf, which should be changed to servers appropriate for your network.

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

- If you no longer need this configuration and wish to purge all IP configuration from an interface, you can use the ip command with the flush option as shown below.

```
ip addr flush eth0
```

# Dynamic IP Address Assignment (DHCP Client)

- To configure your server to use DHCP for dynamic address assignment, create a netplan configuration in the file /etc/netplan/99_config.yaml.

- The example below assumes you are configuring your first Ethernet interface identified as enp3s0.

```
network:
version: 2
renderer: networkd
ethernets:
enp3s0:
dhcp4: true
```

The configuration can then be applied using the netplan command.

```
sudo netplan apply
```

# Dynamic IP Address Assignment (DHCP Client)

- To configure your system to use static address assignment, create a netplan configuration in the file /etc/netplan/99_config.yaml.

- The example below assumes you are configuring your first Ethernet interface identified as eth0. Change the addresses, gateway4, and nameservers values to meet the requirements of your network.

```
network:
version: 2
renderer: networkd
ethernets:
eth0:
addresses:
- 10.10.10.2/24
gateway4: 10.10.10.1
nameservers:
search: [mydomain, otherdomain]
addresses: [10.10.10.1, 1.1.1.1]
```

- The configuration can then be applied using the netplan command.

```
sudo netplan apply
```

# Loopback Interface

- The loopback interface is identified by the system as lo and has a default IP address of 127.0.0.1. It can be viewed using the ip command.

ip address show lo

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
```

IT601 – System and Network Administration

# Networking – Name Server and Bridging

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Name Resolution

- Name resolution as it relates to IP networking is the process of mapping IP addresses to hostnames, making it easier to identify resources on a network.

- Two cases shall be discussed,
  - Name resolution using DNS
  - Name resolution using  static hostname records

# Name resolution using DNS

- Generally, the file /etc/resolv.conf is  a static configuration file that rarely needed to be changed or automatically changed via DCHP client hooks.
- Systemd-resolved handles name server configuration, and it should be interacted with through the systemd-resolve command.
  - systemmd is replacement of sysV and LSB init, All have **PID=1**
  - systemctl: It Controls the systemd system and services.
  - journalctl: Used To manage journal, systemd's own logging system
  - hostnamectl: Can Control hostname.
  - localectl: Helps Configure system local and keyboard layout.
  - timedatectl: Used to Set time and date.
  - systemd-cgls : It Shows cgroup contents.
  - systemadm: It is a Front-end for systemctl command.

- Netplan configures systemd-resolved to generate a list of nameservers and domains to put in /etc/resolv.conf, which is a symlink

  /etc/resolv.conf -> ../run/systemd/resolve/stub-resolv.conf

# Name resolution using DNS

- To configure the resolver, add the IP addresses of the nameservers that are appropriate for your network to the netplan configuration file.

- It is also possible to add an optional DNS suffix search-lists to match your network domain names.
  The resulting file might look like the following:

```
network:
    version: 2
    renderer: networkd
    ethernets:
        enp0s25:
            addresses:
                - 192.168.0.100/24
        gateway4: 192.168.0.1
        nameservers:
            search: [mydomain, otherdomain]
            addresses: [1.1.1.1, 8.8.8.8, 4.4.4.4]
```

# Name resolution using DNS

- The search option can also be used with multiple domain names so that DNS queries will be appended in the order in which they are entered.
- For example, your network may have multiple sub-domains to search; a parent domain of example.com, and two sub-domains, sales.example.com and dev.example.com.
- If you have multiple domains you wish to search, your configuration might look like the following:

```
network:
    version: 2
    renderer: networkd
    ethernets:
        enp0s25:
            addresses:
                - 192.168.0.100/24
            gateway4: 192.168.0.1
            nameservers:
                search: [example.com, sales.example.com, dev.example.com]
                addresses: [1.1.1.1, 8.8.8.8, 4.4.4.4]
```

# Name resolution using DNS

- If you try to ping a host with the name of server1, your system will automatically query DNS for its Fully Qualified Domain Name (FQDN) in the following order:

  1. server1.example.com
  2. server2.sales.example.com
  3. server3.dev.example.com

- If no matches are found, the DNS server will provide a result of not found and the DNS query will fail.

# Name resolution using  static hostname records

- Static hostnames are locally defined hostname-to-IP mappings located in the file /etc/hosts.
  - Entries in the hosts file will have precedence over DNS by default.
  - When system tries to resolve a hostname and it matches an entry in /etc/hosts, it will not attempt to look up the record in DNS.
  - In some configurations, especially when Internet access is not required, servers that communicate with a limited number of resources can be conveniently set to use static hostnames instead of DNS.
- Example of a hosts file
  - Consider local servers have been identified by simple hostnames, aliases and their equivalent Fully Qualified Domain Names (FQDN's)

```
127.0.0.1       localhost
127.0.1.1       ubuntu-server
127.0.0.11      server1     server1.domain.com      vpn
127.0.0.12      server2     server2.domain.com      mail
127.0.0.13      server3     server3.domain.com      www
127.0.0.14      server4     server4.domain.com      ftp
```

# Name Service Switch Configuration

- The order in which your system selects a method of resolving hostnames to IP addresses is controlled by the Name Service Switch (NSS) configuration file /etc/nsswitch.conf.
- Typically static hostnames defined in the systems /etc/hosts file have precedence over names resolved from DNS.
- The following is an example of the line responsible for this order of hostname lookups in the file /etc/nsswitch.conf.

hosts: files mdns4_minimal [NOTFOUND=return] dns mdns4

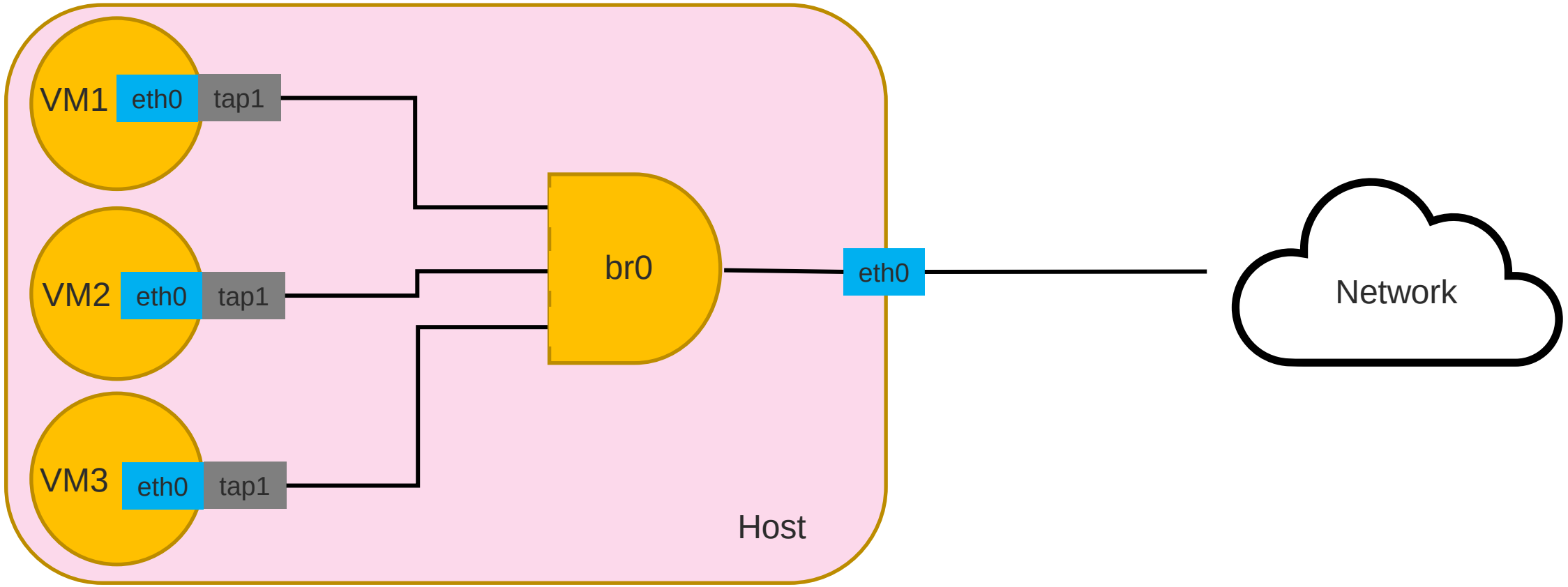
  - **files** first tries to resolve static hostnames located in /etc/hosts.
  - **mdns4_minimal** attempts to resolve the name using Multicast DNS.
  - **[NOTFOUND=return]** means that any response of not found by the preceding mdns4_minimal process should be treated as authoritative and that the system should not try to continue hunting for an answer.
  - **dns** represents a legacy unicast DNS query.
  - **mdns4** represents a Multicast DNS query.

- To modify the order of the above mentioned name resolution methods, you can simply change the hosts: string to the value of your choosing. For example, if you prefer to use legacy Unicast DNS versus Multicast DNS, you can change the string in /etc/nsswitch.conf as shown below.

hosts: files dns [NOTFOUND=return] mdns4_minimal mdns4

# Bridging

- Bridging multiple interfaces is a more advanced configuration but is very useful in multiple scenarios.
- One scenario is using bridge on a system with one interface to allow virtual machines direct access to the outside network.




- Another scenario is setting up a bridge with multiple network interfaces, then using a firewall to filter traffic between two network segments.

# Bridging

- Configure the bridge by editing your netplan configuration found in /etc/netplan/:

```
network:
    version: 2
    renderer: networkd
    ethernets:
        enp3s0:
            dhcp4: no
    bridges:
        br0:
            dhcp4: yes
            interfaces:
                - enp3s0
```

- Now apply the configuration to enable the bridge:

```
sudo netplan apply
```

- The new bridge interface should now be up and running. The brctl provides useful information about the state of the bridge, controls which interfaces are part of the bridge, etc.

IT601 – System and Network Administration

# Networking – Bridging (cont.)

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

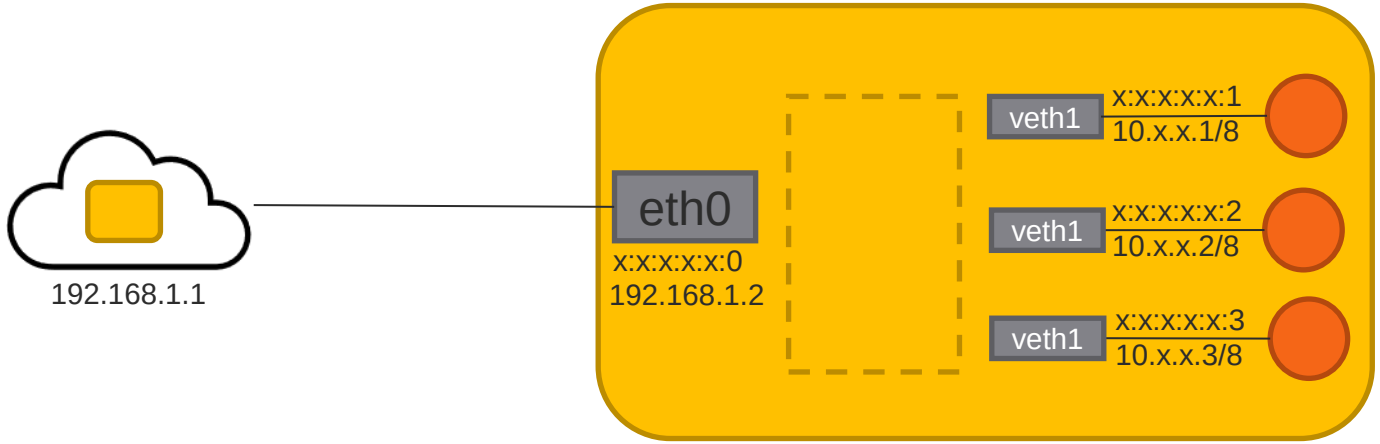
# Legacy Bridging

- The brctl utility  provide ubuntu ‘s legacy bridging that can  be installed using the following command.

  sudo apt install bridge-utils

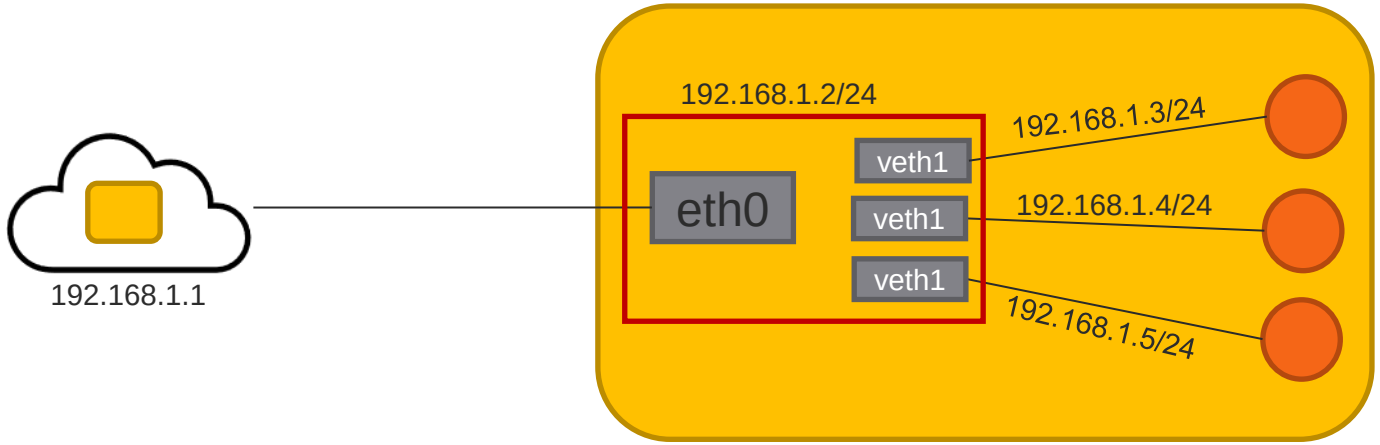
- creating a new bridge device, we’ll create one called br-sna1

  brctl addbr br-sna1
  ip link set br-sn1 up




- A net device can be added to  bridge using following command

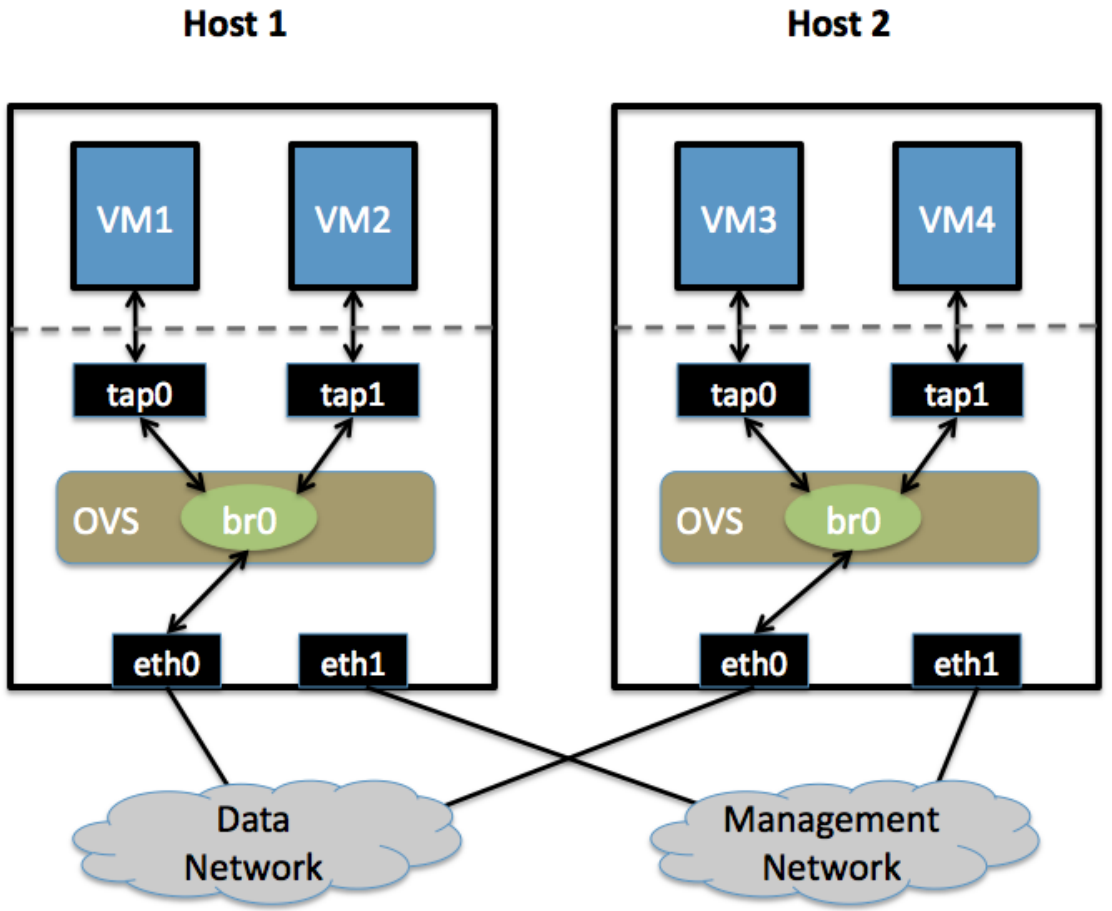
  brctl addif br-sna1 NetDeviceName




- See man brctl for more information

# Open vSwitch Bridge devices

- Open vSwitch (OVS) is an open source, production-quality, multilayer-virtual switch.
- Open vSwitch is designed for massive network automation through programmatic extension, but still with the support for standard protocols and management interfaces.
  - LACP , Link Aggregation Control Protocol
  - NetFlow
  - sFlow
  - IPFIX , Internet Protocol Flow Information Export
  - RSPAN , Remote SPAN
  - CLI
  - 802.1ag
- It is designed to support distribution across multiple physical servers similar to VMware's vNetwork distributed vswitch or Cisco's Nexus 1000V.
- OVS can be installed as below
  - sudo apt update
  - sudo apt install openvswitch-switch
- Service is started automatically after installation:
  - systemctl status openvswitch-switch.service
  - ovs-vsctl show

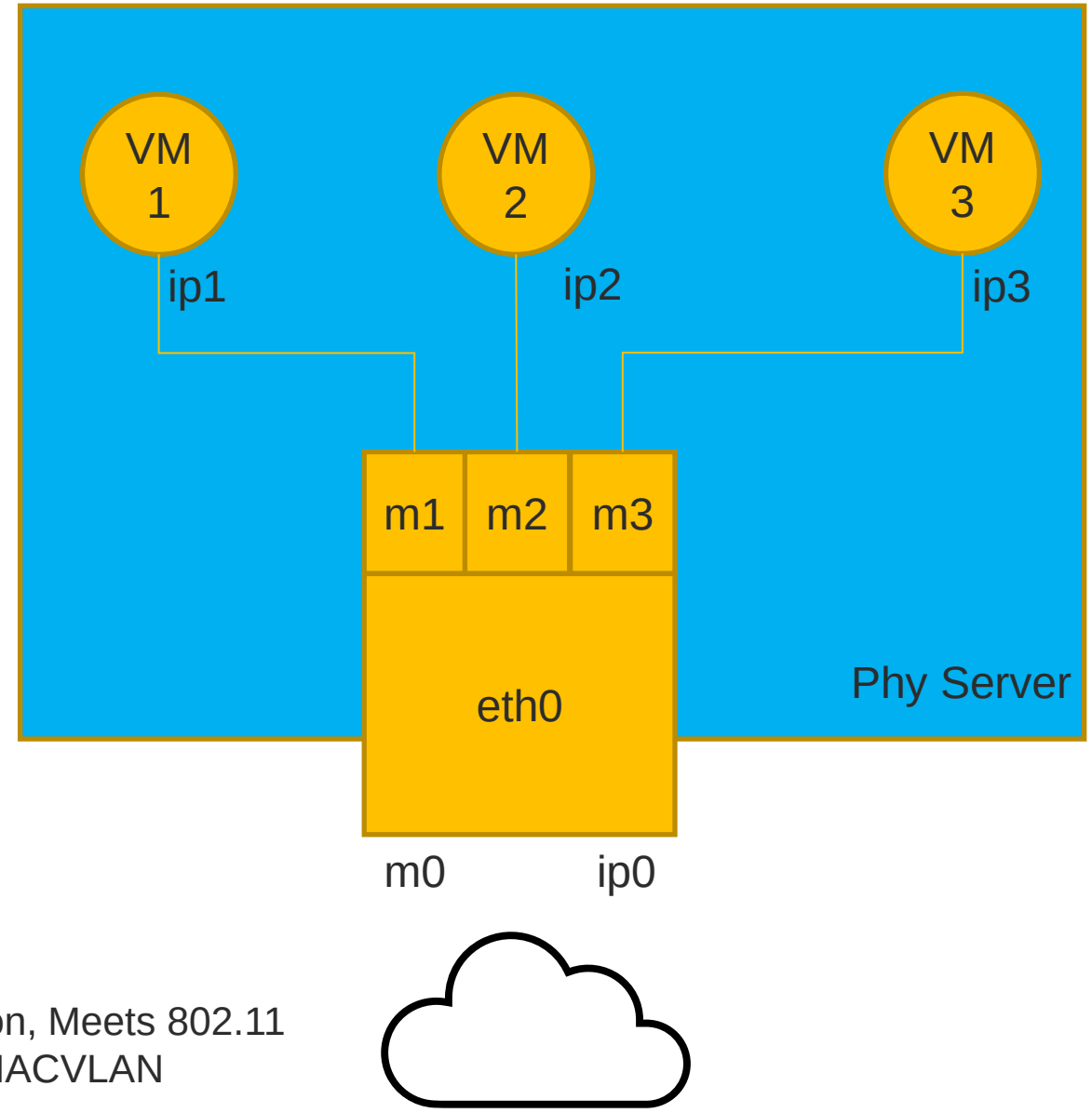


# MACVLAN

- MacVLAN helps the user to configure sub-interfaces of a parent physical Ethernet interface with its own unique MAC address and as a result with its own IP address.

- Applications, VMs and containers can now be grouped to a specific sub-interface, in order to connect directly to the physical network using their own MAC and IP addresses.

- Drawbacks
  - limitation to the number of different MAC addresses allowed on the physical port.
  - NICs have a limitation on the number of MAC addresses they support natively.
  - IEEE 802.11 protocol specifications, multiple MAC addresses on a single client are not allowed.




- Common DHCP Server, Low CPU, Normal Network Utilization, Meets 802.11 standards and Easy to Set-Up are some salient feature of MACVLAN

IT601 – System and Network Administration

# Networking – Name Server and Bridging

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Name Resolution

- Name resolution as it relates to IP networking is the process of mapping IP addresses to hostnames, making it easier to identify resources on a network.

- Two cases shall be discussed,
  - Name resolution using DNS
  - Name resolution using  static hostname records

# Name resolution using DNS

- Generally, the file /etc/resolv.conf is  a static configuration file that rarely needed to be changed or automatically changed via DCHP client hooks.
- Systemd-resolved handles name server configuration, and it should be interacted with through the systemd-resolve command.
  - systemmd is replacement of sysV and LSB init, All have **PID=1**
  - systemctl: It Controls the systemd system and services.
  - journalctl: Used To manage journal, systemd's own logging system
  - hostnamectl: Can Control hostname.
  - localectl: Helps Configure system local and keyboard layout.
  - timedatectl: Used to Set time and date.
  - systemd-cgls : It Shows cgroup contents.
  - systemadm: It is a Front-end for systemctl command.

- Netplan configures systemd-resolved to generate a list of nameservers and domains to put in /etc/resolv.conf, which is a symlink

  /etc/resolv.conf -> ../run/systemd/resolve/stub-resolv.conf

# Name resolution using DNS

- To configure the resolver, add the IP addresses of the nameservers that are appropriate for your network to the netplan configuration file.

- It is also possible to add an optional DNS suffix search-lists to match your network domain names.
  The resulting file might look like the following:

```
network:
    version: 2
    renderer: networkd
    ethernets:
        enp0s25:
            addresses:
                - 192.168.0.100/24
        gateway4: 192.168.0.1
        nameservers:
            search: [mydomain, otherdomain]
            addresses: [1.1.1.1, 8.8.8.8, 4.4.4.4]
```

# Name resolution using DNS

- The search option can also be used with multiple domain names so that DNS queries will be appended in the order in which they are entered.
- For example, your network may have multiple sub-domains to search; a parent domain of example.com, and two sub-domains, sales.example.com and dev.example.com.
- If you have multiple domains you wish to search, your configuration might look like the following:

```
network:
    version: 2
    renderer: networkd
    ethernets:
        enp0s25:
            addresses:
                - 192.168.0.100/24
            gateway4: 192.168.0.1
            nameservers:
                search: [example.com, sales.example.com, dev.example.com]
                addresses: [1.1.1.1, 8.8.8.8, 4.4.4.4]
```

# Name resolution using DNS

- If you try to ping a host with the name of server1, your system will automatically query DNS for its Fully Qualified Domain Name (FQDN) in the following order:

  1. server1.example.com
  2. server2.sales.example.com
  3. server3.dev.example.com

- If no matches are found, the DNS server will provide a result of not found and the DNS query will fail.

# Name resolution using  static hostname records

- Static hostnames are locally defined hostname-to-IP mappings located in the file /etc/hosts.
  - Entries in the hosts file will have precedence over DNS by default.
  - When system tries to resolve a hostname and it matches an entry in /etc/hosts, it will not attempt to look up the record in DNS.
  - In some configurations, especially when Internet access is not required, servers that communicate with a limited number of resources can be conveniently set to use static hostnames instead of DNS.
- Example of a hosts file
  - Consider local servers have been identified by simple hostnames, aliases and their equivalent Fully Qualified Domain Names (FQDN's)

```
127.0.0.1       localhost
127.0.1.1       ubuntu-server
127.0.0.11      server1     server1.domain.com      vpn
127.0.0.12      server2     server2.domain.com      mail
127.0.0.13      server3     server3.domain.com      www
127.0.0.14      server4     server4.domain.com      ftp
```

# Name Service Switch Configuration

- The order in which your system selects a method of resolving hostnames to IP addresses is controlled by the Name Service Switch (NSS) configuration file /etc/nsswitch.conf.
- Typically static hostnames defined in the systems /etc/hosts file have precedence over names resolved from DNS.
- The following is an example of the line responsible for this order of hostname lookups in the file /etc/nsswitch.conf.

hosts: files mdns4_minimal [NOTFOUND=return] dns mdns4

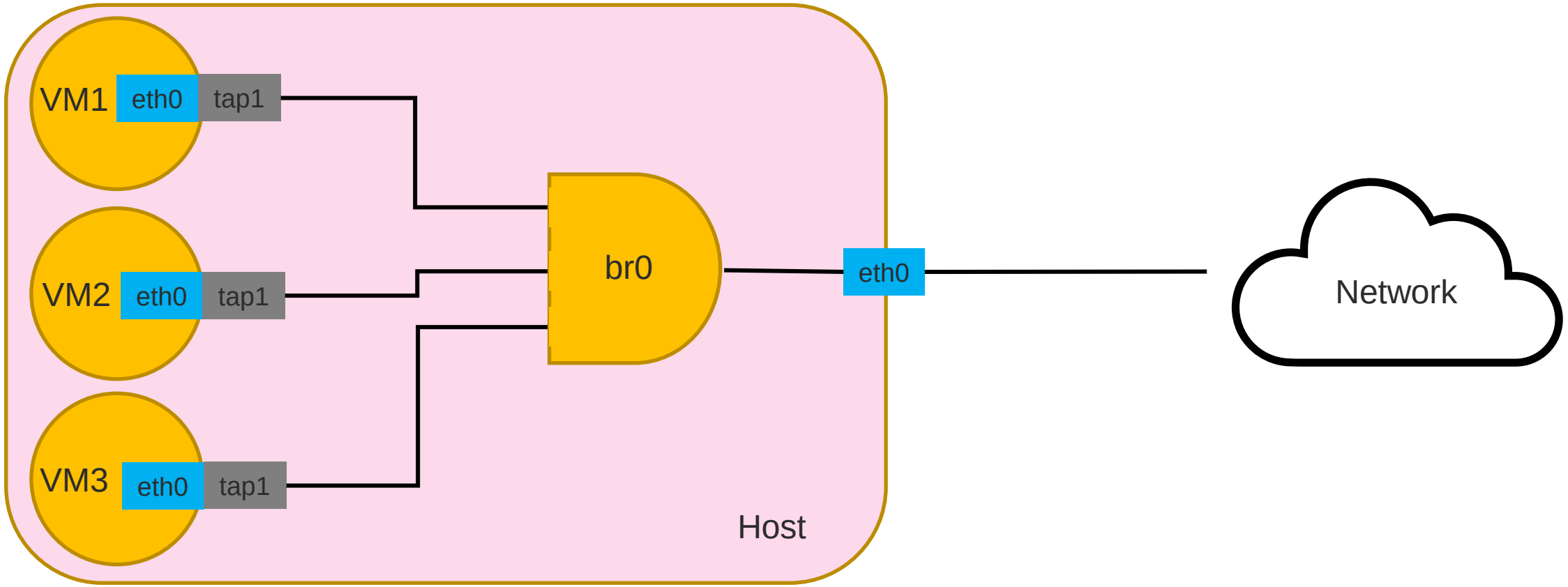
  - **files** first tries to resolve static hostnames located in /etc/hosts.
  - **mdns4_minimal** attempts to resolve the name using Multicast DNS.
  - **[NOTFOUND=return]** means that any response of not found by the preceding mdns4_minimal process should be treated as authoritative and that the system should not try to continue hunting for an answer.
  - **dns** represents a legacy unicast DNS query.
  - **mdns4** represents a Multicast DNS query.

- To modify the order of the above mentioned name resolution methods, you can simply change the hosts: string to the value of your choosing. For example, if you prefer to use legacy Unicast DNS versus Multicast DNS, you can change the string in /etc/nsswitch.conf as shown below.

hosts: files dns [NOTFOUND=return] mdns4_minimal mdns4

# Bridging

- Bridging multiple interfaces is a more advanced configuration but is very useful in multiple scenarios.
- One scenario is using bridge on a system with one interface to allow virtual machines direct access to the outside network.




- Another scenario is setting up a bridge with multiple network interfaces, then using a firewall to filter traffic between two network segments.

# Bridging

- Configure the bridge by editing your netplan configuration found in /etc/netplan/:

```
network:
    version: 2
    renderer: networkd
    ethernets:
        enp3s0:
            dhcp4: no
    bridges:
        br0:
            dhcp4: yes
            interfaces:
                - enp3s0
```

- Now apply the configuration to enable the bridge:

```
sudo netplan apply
```

- The new bridge interface should now be up and running. The brctl provides useful information about the state of the bridge, controls which interfaces are part of the bridge, etc.

IT601 – System and Network Administration

# User Management

Arif Husen

Department of Computer Science and Information Technology,

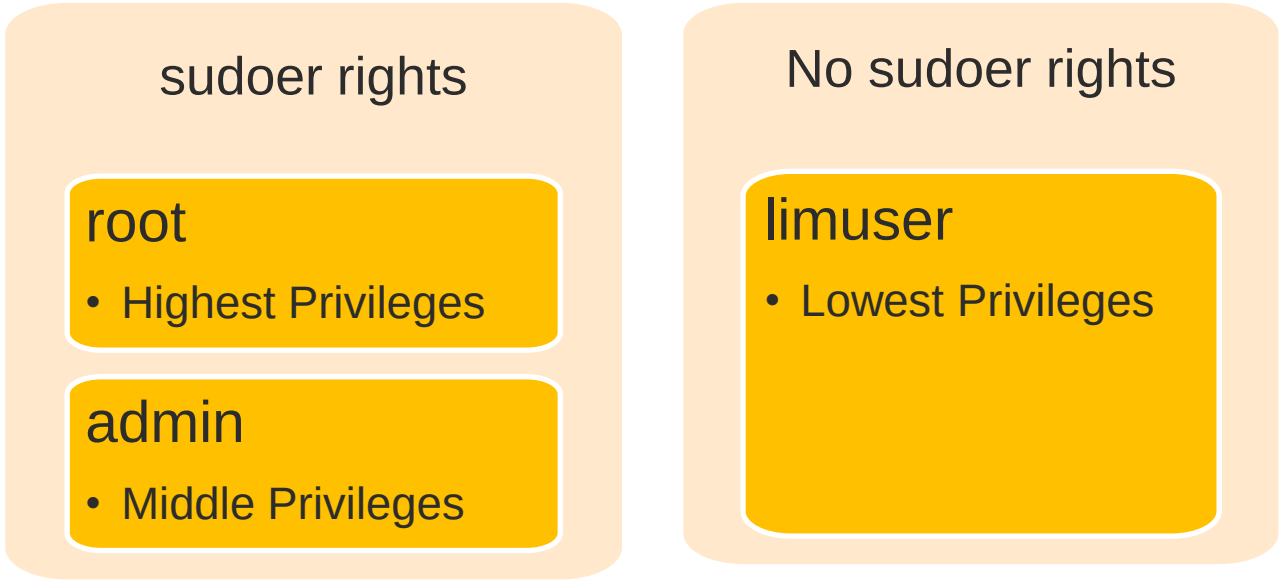
Virtual University of Pakistan

# User Management

- User management is a critical part of maintaining a secure system. Ineffective user and privilege management often lead many systems into being compromised.
- it is important  to understand how you can protect your server through simple and effective user account management techniques.

- Root User

  - Root is the superuser account in Unix and Linux. It is a user account for administrative purposes, and typically has the highest access rights on the system.
  - Usually, the root user account is called root . However, in Unix and Linux, any account with user id 0 is a root account, regardless of the name.



# Root user in Ubuntu

- In ubuntu server, administrative root account  is disabled by default.
  - This does not mean that the root account has been deleted or that it may not be accessed. It merely has been given a password which matches no possible encrypted value, therefore may not log in directly by itself
  - Instead, users are encouraged to make use of a tool by the name of sudo to carry out system administrative duties.
- sudo allows an authorized user to temporarily elevate their privileges using their own password instead of having to know the password belonging to the root account.
  - This simple yet effective methodology provides accountability for all user actions, and gives the administrator granular control over which actions a user can perform with said privileges.
- By default, the initial user created by the installer is a member of the group "sudo" which is added to the file /etc/sudoers as an authorized sudo user.
- To give any other account full root access through sudo, simply add them to the sudo group.

# Enabling/disabling root

- If for some reason you wish to enable the root account, simply give it a password:

  sudo passwd

- sudo will prompt you for your password, and then ask you to supply a new password for root.

```
[sudo] password for username: (enter your own password)
Enter new UNIX password: (enter a new password for root)
Retype new UNIX password: (repeat new password for root)
passwd: password updated successfully
```

- To disable the root account password, use the following passwd syntax:

  sudo passwd -l root

- However, to disable the root account itself, use the following command:

  usermod --expiredate 1

- You should read more on sudo by reading the man page:

  man sudo

# User Management Operations

The process for managing local users and groups is straightforward and differs very little from most other GNU/Linux operating systems. Ubuntu and other Debian based distributions encourage the use of the "adduser" package for account management.

**Add/Remove User**

sudo adduser username

- follow the prompts to give the account a password and identifiable characteristics

sudo deluser username

- delete a user account and its primary group

**Lock/Unlock User**

sudo passwd -l username

sudo passwd -u username

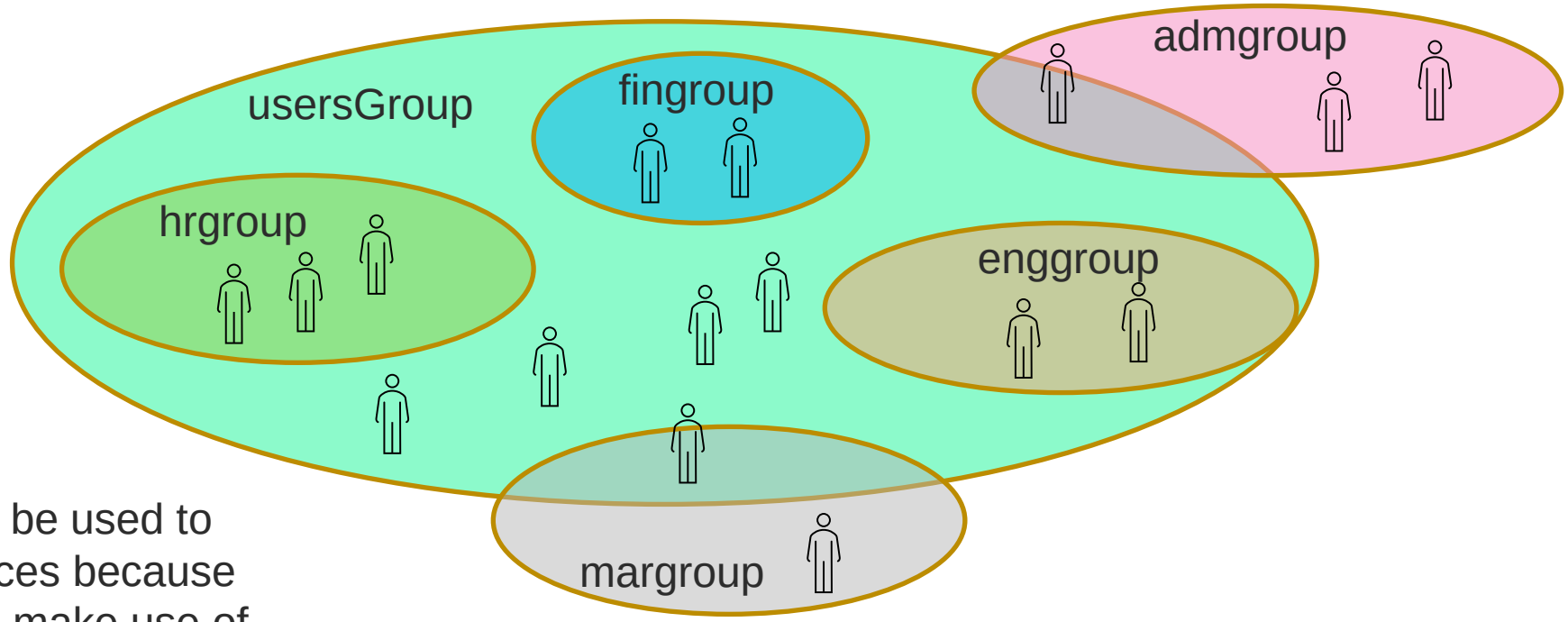
- Deleting an account does not remove their respective home folder. It is up to you whether or not you wish to delete the folder manually or keep it according to your desired retention policies.

Remember, any user added later on with the same UID/GID as the previous owner will now have access to this folder if you have not taken the necessary precautions.

# User Group Management

- A user can be assigned to group(s)  based on their  department or activities.

- A group allows a user special access to system resources, such as files, directories, or processes (programs) that are running on the system.




- This group membership can also be used to prevent access to system resources because several security features in Linux make use of groups to impose security restrictions.

# User Group Management

- Every user is a member of at least one group. This first group is called the user's primary group. Any additional groups a user is a member of are called the user's secondary groups.
- Group membership can be displayed by executing either the id or groups command:

```
student@onecoursesource:~$ id
uid=1002(student) gid=1002(student)
groups=1002(student),60(games),1001(ocs)

student@onecoursesource:~$ groups
student games ocs
```

- Both the id and groups commands display information about the current user by default. Both commands also accept an argument of another user account name:

```
student@onecoursesource:~$ id root
uid=0(root) gid=0(root) groups=0(root)
student@onecoursesource:~$ groups root
root : root
```

- The most important difference between primary and secondary group membership relates to when a user creates a new file. Each file is owned by a user ID and a group ID.
- When a user creates a file, the user's primary group membership is used for the group ownership of the file:

# Group information

- Group information is stored in several files:
  - The /etc/passwd file contains user account information, including the primary group membership for each user.
  - The /etc/group file stores information about each group, including the group name, group ID (GID) and secondary user membership.

    ```
    student@onecoursesource:~$ grep student /etc/passwd
    student:x:1002:1002::/home/student:
    ```

  - The /etc/gshadow file stores additional information for the group, including group administrators and the group password.

    ```
    student@onecoursesource:~$ head /etc/group
    root:x:0:
    daemon:x:1:
    bin:x:2:
    sys:x:3:
    adm:x:4:syslog,bo
    tty:x:5:
    disk:x:6:
    lp:x:7:
    mail:x:8:
    news:x:9:
    ```

# Special Groups

- A typical Linux system will have many default group accounts. These default group accounts typically have GID values under 1000, making it easy for an administrator to recognize these as special accounts.
- Additionally, if you add new software to the system, more groups may be added as software vendors make use of both user and group accounts to provide controlled access to files that are part of the software.
- Administrators who are focused on security should be aware of these special group accounts because these accounts can provide either security features or pose security threats.

| Group | Description |
|---|---|
| root | This group account is reserved for the system administrator. Do not add a regular user to this group because it will provide the regular user with elevated access to system files. |
| adm | Members of this group typically have access to files related to system monitoring (such as log files). Being able to see the contents of these files can provide more information about the system than a regular user would typically have. |
| lp | This is one of many groups (including tty, mail, and cdrom) used by the operating system to provide access to specific files. Typically, regular users are not added to these groups because they are used by background processes called daemons. |
| sudo | This group is used in conjunction with the sudo command. |
| staff | A default group that was traditionally used on Unix systems but is rarely used in modern Linux distributions. |
| users | A default group that is rarely used in modern Linux distributions. |
| operators | A group that was traditionally used on Unix systems for users who required elevated privileges for specific system tasks. This group is rarely used in modern Linux distributions. |

# Adding Removing Groups

- To add or delete a personalized group, use the following syntax, respectively:

```
sudo addgroup groupname
sudo delgroup groupname
```

- To add a user to a group, use the following syntax:

```
sudo adduser username groupname
```

IT601 – System and Network Administration

# User Level Security

Arif Husen

Department of Computer Science and Information Technology,

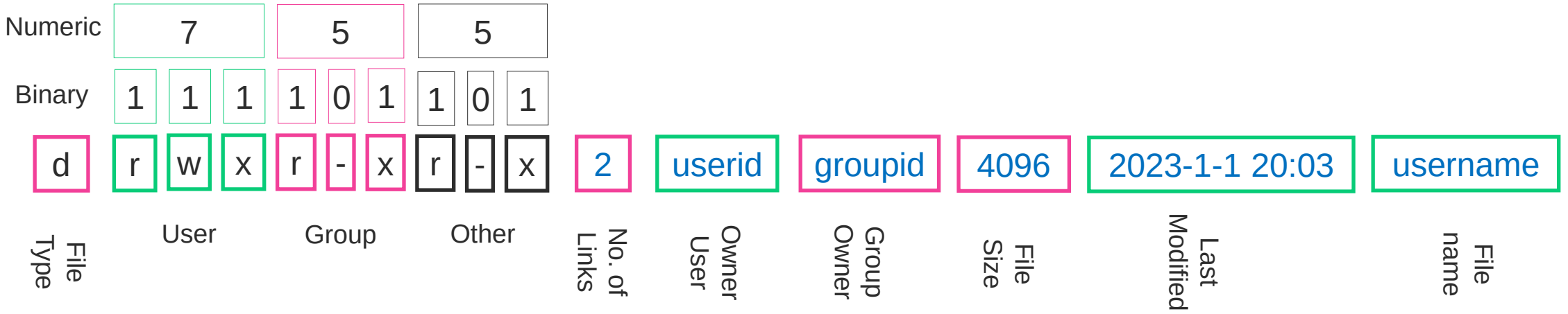
Virtual University of Pakistan

# User Profile Security

- When a new user is created, the *adduser* utility creates a new home directory named /home/username.
  - The default profile is modeled according to contents of /etc/skel, which includes all profile basics.
  - For multiuser environment, close attention is required to the user home directory permissions to ensure confidentiality.
- By default, user home directories in Ubuntu are created with world read/execute permissions.
  - This means that all users can browse and access the contents of other user's home directories.
- To verify your current user home directory permissions, use the following syntax:

```
ls -ld /home/username
```

```
drwxr-xr-x 2 username username 4096 2007-10-02 20:03 username
```



# User Profile Security

- To remove the world readable-permissions, following command can be used.

  sudo chmod 0750 /home/username

- The efficient approach is to modify the adduser global default permissions when creating user home folders. Simply edit the file /etc/adduser.conf and modify the DIR_MODE variable to something appropriate, so that all new home directories will receive the correct permissions.

```
DIR_MODE=0750
```

ls -ld /home/username

drwxr-x--- 2 username username 4096 2007-10-02 20:03 username

# Password Policy

- A strong password policy is one of the most important aspects of your security posture. Many successful security breaches involve simple brute force and dictionary attacks against weak passwords.

- To offer any form of remote access involving your local password system, make sure you adequately address

  - Minimum password complexity requirements
  - Maximum password lifetimes
  - Frequent audits of your authentication systems

# Password Expiry

- When creating user accounts, you should make it a policy to have a minimum and maximum password age forcing users to change their passwords when they expire.
- To easily view the current status of a user account, use the following syntax:

  sudo chage -l username

```
Last password change : Jan 20, 2015
Password expires : never
Password inactive : never
Account expires : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
```

- To set any of these values, simply use the following syntax, and follow the interactive prompts:

  sudo chage username

- Example : Change the explicit expiration date (-E) to 01/31/2015, minimum password age (-m) of 5 days, maximum password age (-M) of 90 days, inactivity period (-I) of 30 days after password expiration, and a warning time period (-W) of 14 days before password expiration:

  sudo chage -E 01/31/2015 -m 5 -M 90 -I 30 -W 14 username

# Other Considerations

- Many applications use alternate authentication mechanisms that can be easily. It is important to understand and control how users authenticate and gain access to services and applications on your server.
- SSH Access by Disabled Users
  - Simply disabling/locking a user account will not prevent a user from logging into your server remotely if they have previously set up RSA public key authentication.
  - They will still be able to gain shell access to the server, without the need for any password.
    - Remember to check the users home directory for files that will allow for this type of authenticated SSH access, e.g. /home/username/.ssh/authorized_keys.
    - Remove or rename the directory .ssh/ in the user's home folder to prevent further SSH authentication capabilities.
    - Be sure to check for any established SSH connections by the disabled user, as it is possible they may have existing inbound or outbound connections. Kill any that are found.

      who | grep username (to get the pts/# terminal)
      sudo pkill -f pts/#
  - Restrict SSH access to only required user accounts. You may create a group called "sshlogin" and add the group name as the value associated with the AllowGroups variable located in the file /etc/ssh/sshd_config.

    AllowGroups sshlogin
- External User Database Authentication

# IT601 – System and Network Administration

## < End>

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

IT601 – System and Network Administration

# Remote Administration

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# OpenSSH

- This topic introduces a powerful collection of tools for the remote control of, and transfer of data between, networked computers called OpenSSH.


- OpenSSH is a freely available version of the Secure Shell (SSH) protocol family of tools for remotely controlling, or transferring files between, computers.

- Traditional tools used to accomplish these functions, such as telnet or rcp, are insecure and transmit the user's password in cleartext when used.

- OpenSSH provides a server daemon and client tools to facilitate secure, encrypted remote control and file transfer operations, effectively replacing the legacy tools.

# OpenSSH

- The OpenSSH server component, sshd, listens continuously for client connections from any of the client tools.
- When a connection request occurs, sshd sets up the correct connection depending on the type of client tool connecting.
  - if the remote computer is connecting with the ssh client application, the OpenSSH server sets up a remote control session after authentication.
  - If a remote user connects to an OpenSSH server with scp, the OpenSSH server daemon initiates a secure copy of files between the server and client after authentication.
- OpenSSH can use many authentication methods, including plain password, public key, and Kerberos tickets.

# Install OpenSSH

- Installation of the OpenSSH client and server applications is simple.
  - To install the OpenSSH client applications on your Ubuntu system, use this command at a terminal prompt:

    sudo apt install openssh-client
  - To install the OpenSSH server application, and related support files, use this command at a terminal prompt:

    sudo apt install openssh-server

- Configuring the OpenSSH
  - You may configure the default behavior of the OpenSSH server application, sshd, by editing the file /etc/ssh/sshd_config.
  - For information about the configuration directives used in this file, you may view the appropriate manual page with the following command, issued at a terminal prompt:

    man sshd_config

# Configuring the OpenSSH

- There are many directives in the sshd configuration file controlling such things as communication settings, and authentication modes.
- Example configuration : Various  directives that can be changed by editing the /etc/ssh/sshd_config file.

| | | |
|---|---|---|
| 1 | Copy the /etc/ssh/sshd_config file and protect it from writing with the following commands, issued at a terminal prompt: | sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.original<br>sudo chmod a-w /etc/ssh/sshd_config.original |
| 2 | To set your OpenSSH to listen on TCP port 2222 instead of the default TCP port 22, change the Port directive as such: | Port 2222 |
| 3 | To have sshd allow public key-based login credentials, simply add or modify the line: | PubkeyAuthentication yes |
| 4 | To make your OpenSSH server display the contents of the /etc/issue.net file as a pre-login banner, simply add or modify the line In the /etc/ssh/sshd_config file. | Banner /etc/issue.net |
| 5 | After making changes to the /etc/ssh/sshd_config file, save the file, and restart the sshd server application to effect the changes using the following command at a terminal prompt: | sudo systemctl restart sshd.service |

# SSH Keys

- SSH keys allow authentication between two hosts without the need of a password. SSH key authentication uses two keys, a private key and a public key.
  - To generate the keys, from a terminal prompt enter:

    ssh-keygen -t rsa

    This will generate the keys using the RSA Algorithm. During the process you will be prompted for a password. Simply hit Enter when prompted to create the key.

  - By default the public key is saved in the file ~/.ssh/id_rsa.pub, while ~/.ssh/id_rsa is the private key. Now copy the id_rsa.pub file to the remote host and append it to ~/.ssh/authorized_keys by entering:

    ssh-copy-id username@remotehost

  - Finally, double check the permissions on the authorized_keys file, only the authenticated user should have read and write permissions. If the permissions are not correct change them by:

    chmod 600 .ssh/authorized_keys

  - You should now be able to SSH to the host without being prompted for a password.

# Puppet

- Puppet is a cross platform framework enabling system administrators to perform common tasks using code.
- The code can do a variety of tasks from installing new software, to checking file permissions, or updating user accounts.
- Puppet is great not only during the initial installation of a system, but also throughout the system's entire life cycle. In most circumstances puppet will be used in a client/server configuration.
- Puppet uses a client-server approach and consists of the following systems:
  - The Puppet Master is a server with the Puppet Master daemon that manages crucial system information for all nodes using manifests.
  - The Puppet Agents are nodes with Puppet installed on them with the Puppet Agent daemon running.

# Puppet

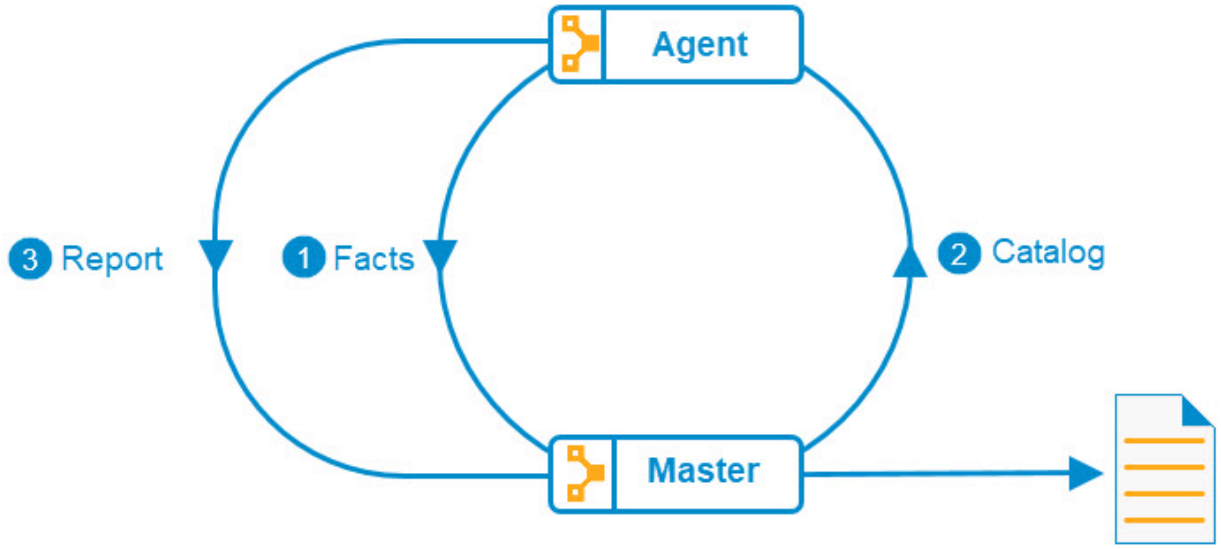
- Puppet utilizes a client/server architecture consisting of the Puppet Master and Puppet Agents. Puppet Agents uses pull mode to poll the master and retrieve node-specific and site-specific configuration info.

- The topology goes through the following steps:

  1 - A node running a Puppet Agent daemon gathers all the information (facts) about itself, and the agent sends the facts to the Puppet Master.

  2 - The Puppet Master uses the data to create a catalog on how the node should be configured and sends it back to the Puppet Agent.

  3 - The Puppet Agent configures itself based on the catalog and reports back to the Puppet Master.



# Installing and Configuring Puppet

- Prior to configuring puppet you may want to add a DNS CNAME record for puppet.example.com, where example.com is your domain.
- By default Puppet clients check DNS for puppet.example.com as the puppet server name, or Puppet Master.
- If you do not wish to use DNS, you can add entries to the server and client /etc/hosts file. For example, in the Puppet server's /etc/hosts file add:

```
127.0.0.1 localhost.localdomain localhost puppet
192.168.1.17 puppetclient.example.com puppetclient
```

- On each Puppet client, add an entry for the server:

```
192.168.1.16 puppetmaster.example.com puppetmaster puppet
```

- To install Puppet, in a terminal on the server enter:

```
sudo apt install puppetmaster
```

- On the client machine, or machines, enter:

```
sudo apt install puppet
```

# Installing and Configuring Puppet

- Create a folder path for the apache2 class:

  sudo mkdir -p /etc/puppet/modules/apache2/manifests

- Now setup some resources for apache2. Create a file /etc/puppet/modules/apache2/manifests/init.pp containing the following:

```
class apache2 {
    package { 'apache2':
        ensure => installed,
    }
    service { 'apache2':
        ensure => true,
        enable => true,
        require => Package['apache2'],
    }
}
```

- Next, create a node file /etc/puppet/code/environments/production/manifests/site.pp with:

```
node 'puppetclient.example.com' {
    include apache2
}
```

- The final step for this simple Puppet server is to restart the daemon:

  sudo systemctl restart puppetmaster.service

# Installing and Configuring Puppet

- Now everything is configured on the Puppet server, it is time to configure the client.
  - First, configure the Puppet agent daemon to start. Edit /etc/default/puppet, changing START to yes:

    START=yes
  - Then start the service:

    sudo systemctl start puppet.service
  - View the client cert fingerprint

    sudo puppet agent --fingerprint
  - Back on the Puppet server, view pending certificate signing requests:

    sudo puppet cert list
  - On the Puppet server, verify the fingerprint of the client and sign puppetclient's cert:

    sudo puppet cert sign puppetclient.example.com
  - On the Puppet client, run the puppet agent manually in the foreground. This step isn't strictly speaking necessary, but it is the best way to test and debug the puppet service.

    sudo puppet agent --test
  - Check /var/log/syslog on both hosts for any errors with the configuration. If all goes well the apache2 package and it's dependencies will be installed on the Puppet client.

# Zentyal

- Zentyal is a Linux small business server that can be configured as a gateway, infrastructure manager, unified threat manager, office server, unified communication server or a combination of them.

## Integrated

- All network services managed by Zentyal are tightly integrated, automating most tasks.
- This saves time and helps to avoid errors in network configuration and administration.

## Opensource

- Zentyal is open source, released under the GNU General Public License (GPL) and runs on top of Ubuntu GNU/Linux.
- Zentyal consists of a series of packages (usually one for each module) that provide a web interface to configure the different servers or services.
- Zentyal publishes one major stable release once a year based on the latest Ubuntu LTS release.

## Configuration

- The configuration is stored on a key-value Redis database, but users, groups, and domains-related configuration is on OpenLDAP.
- When you configure any of the available parameters through the web interface, final configuration files are overwritten using the configuration templates provided by the modules.

## Advantage

- The main advantage of using Zentyal is a unified, graphical user interface to configure all network services and high, out-of-the-box integration between them.

# Installing and Configuring Zantyal

- Create a new user to access the Zentyal web interface, run:

  sudo adduser username sudo

- Add the Zentyal repository to your repository list:

  sudo add-apt-repository "deb http://archive.zentyal.org/zentyal 3.5 main extra"

- Import the public keys from Zentyal:

  sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 10E239FF
  wget -q http://keys.zentyal.org/zentyal-4.2-archive.asc -O- | sudo apt-key add -

- Update your packages and install Zentyal:

  sudo apt update
  sudo apt install zentyal

- During installation you will be asked to set a root MySQL password and confirm port 443.

# Installing and Configuring Zantyal

- Any system account belonging to the sudo group is allowed to log into the Zentyal web interface. The user created while installing Ubuntu Server will belong to the sudo group by default.

- To access the Zentyal web interface, point a browser to https://localhost/ or to the IP address of your remote server.

- As Zentyal creates its own self-signed SSL certificate, you will have to accept a security exception on your browser. Log in with the same username and password used to log in to your server.

- Once logged in you will see an overview of your server. Individual modules, such as Antivirus or Firewall, can be installed by simply clicking them and then clicking Install. Selecting server roles like Gateway or Infrastructure can be used to install multiple modules at once.

# IT601 – System and Network Administration

# Logging

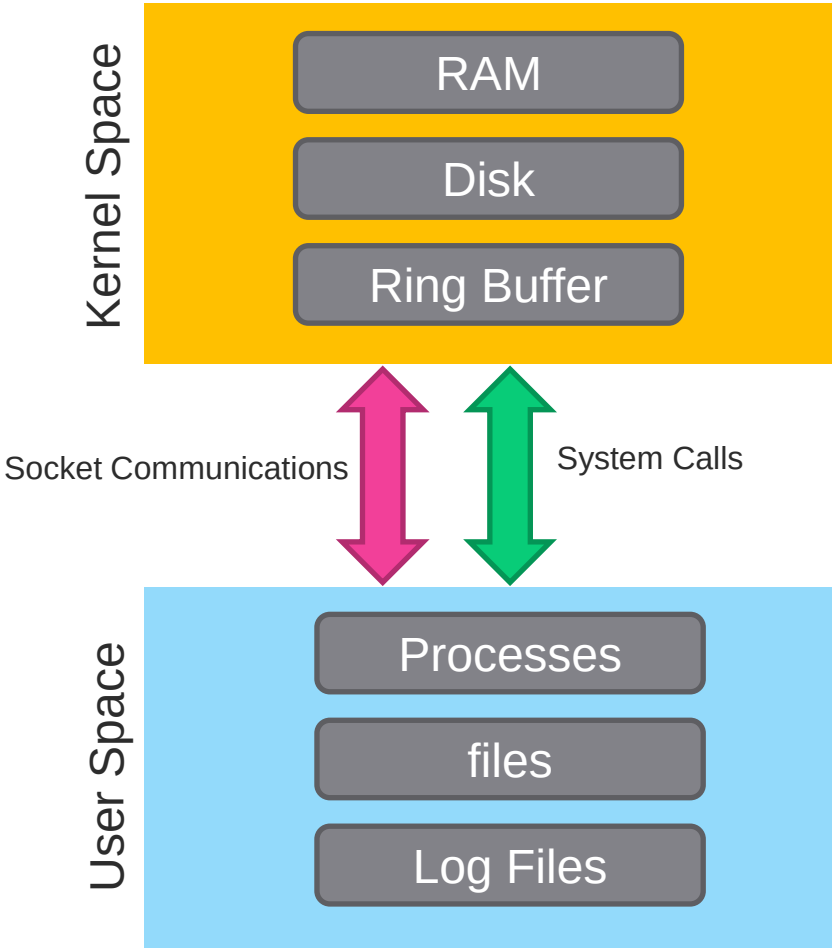
Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Logging

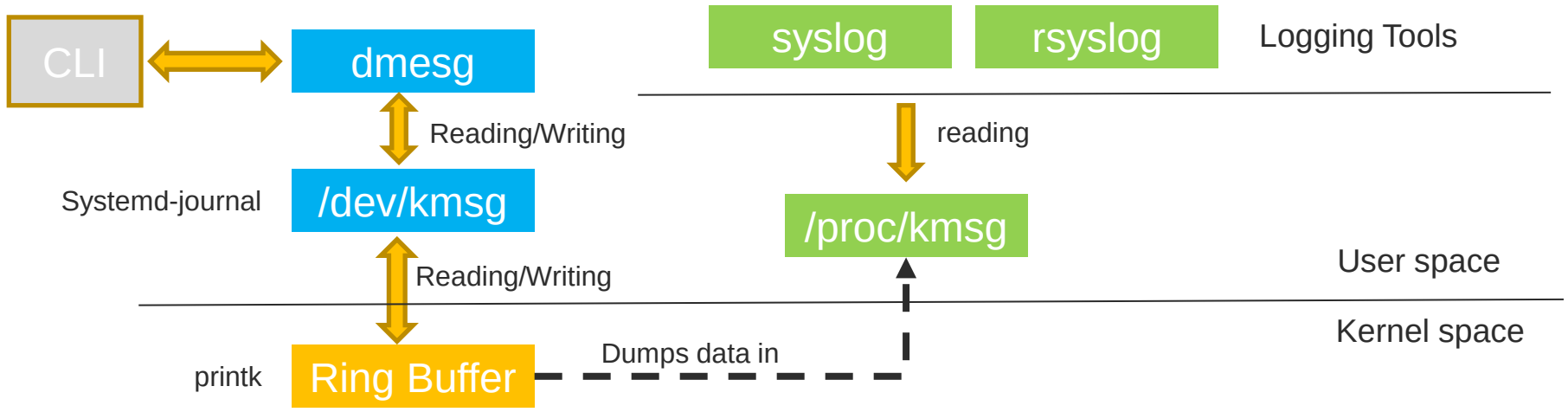
- Logging refers to record keeping of information about events that occur in a computer system, such as problems, errors or just information on current operations.
  - Different types of events may occur in the operating system or in other software.
- These log messages can then be used to monitor and understand the operation of the system, to debug problems, or during an audit.
- Logging is particularly important in multi-user software, to have a central overview of the operation of the system.
- On Linux, you have two types of logging mechanisms :
  - Kernel logging: related to errors, warning or information entries that your kernel may write
  - User logging: linked to the user space, those log entries are related to processes or services that may run on the host machine.



# Kernel Logging

- On the kernel space, logging is done via the Kernel Ring Buffer. **The ring buffer is a circular buffer that is the first datastructure storing log messages when the system boots up.**
  - When starting Linux machine, if log messages are displayed on the screen, those messages are stored in the kernel ring buffer.
  - The Kernel logging is started before user logging
  - The kernel ring buffer, pretty much like any other log files on your system can be inspected.
  - In order to open Kernel-related logs on your system, you have to use the “dmesg” command.

  Example of events : Errors in mounting a disk, Driver Loading etc



# Log files locations

- There are many different log files that all serve different purposes. When trying to find a log about something, you should start by identifying the most relevant file.

- **System logs**

| Auth log | Daemon Log | Debug log | Kernel log | System log |
|---|---|---|---|---|
| /var/log/auth.log | /var/log/daemon.log | /var/log/debug | /var/log/kern.log | /var/log/syslog |

- **Application logs**

| Apache logs | X11 server logs | Print System Logs | Rootkit Hunter Log | SMB Logs |
|---|---|---|---|---|
| /var/log/apache2/ | /var/log/Xorg.0.log | /var/log/cups/error_log | /var/log/rkhunter.log | /var/log/samba |

- **Non-human-readable logs**

| Login failures log | Last logins log | Login records log |
|---|---|---|
| /var/log/faillog | /var/log/lastlog | /var/log/lastlog |

# Viewing and monitoring log files

- The most basic way to view files from the command line is using the cat command. You simply pass in the filename, and it outputs the entire contents of the file: cat file.txt.
- Viewing the start or end of a file
  - It is generally required to quickly view the first or last n number of lines of a file.
  - the head and tail commands come in handy.
  - These commands work much like cat, although you can specify how many lines from the start/end of the file you want to view.
  - To view the first 15 lines of a file, run head -n 15 file.txt, and to view the last 15, run tail -n 15 file.txt.
  - Due to the nature of log files being appended to at the bottom, the tail command will generally be more useful.
- Monitoring files
  - To monitor a log file, you may pass the -f flag to tail. It will keep running, printing new additions to the file, until you stop it (Ctrl + C). For example: tail -f file.txt.
- Searching files
  - One way that we looked at to search files is to open the file in less and press /.
  - A faster way to do this is to use the grep command.
  - We specify what we want to search for in double quotes, along with the filename, and grep will print all the lines containing that search term in the file. For example, to search for lines containing “test” in file.txt, you would run grep "test" file.txt.
  - If the result of a grep search is too long, you may pipe it to less, allowing you to scroll and search through it: grep "test" file.txt | less.

# System Logging Daemon (syslogd)

- The system logging daemon syslogd, also known as sysklogd, awaits logging messages from numerous sources and routes the messages to the appropriate file or network destination.
  - Messages logged to syslogd usually contain common elements like system hostnames and time-stamps in addition to the specific log information.

- Configuration of syslogd
  - The syslogd daemon's configuration file is /etc/syslog.conf.
  - Each entry in this file consists of two fields, the selector and the action.
  - The selector field specifies a facility to be logged, such as for example the auth facility which deals with authorization, and a priority level to log such information at, such as info, or warning.
  - The action field consists of a target for the log information, such as a standard log file (i.e. /var/log/syslog), or the hostname of a remote computer to send the log information to.

# Log Rotation

- When viewing directory listings in /var/log or any of its subdirectories, you may encounter log files with names such as daemon.log.0, daemon.log.1.gz, and so on.
  - What are these log files? They are 'rotated' log files. That is, they have automatically been renamed after a predefined time-frame, and a new original log started. After even more time the log files are compressed with the gzip utility as in the case of the example daemon.log.1.gz.
- The purpose of log rotation is to archive and compress old logs so that they consume less disk space, but are still available for inspection as needed.
- Typically, logrotate is called from the system-wide cron script /etc/cron.daily/logrotate, and further defined by the configuration file /etc/logrotate.conf. Individual configuration files can be added into /etc/logrotate.d

# Log Rotation

- Log files that have zeroes appended at the end are rotated files. That means log file names have automatically been changed within the system.
- logrotate handles systems that create significant amounts of log files. The command is used by the cron scheduler and reads the logrotate configuration file /etc/logrotate.conf. It's also used to read files in the logrotate configuration directory.

```
var/log/log name here].log {
Missingok
Notifempty
Compress
Size 20k
Daily
Create 0600 root root
}
```

- The commands perform the actions as follows:

  - **missingok** – Tells logrotate not to output an error if a log file is missing
  - **notifempty** – Does not rotate the log file if it is empty. It reduces the size of the log file with gzip
  - **size** – Ensures that the log file does not exceed the specified dimension and rotates it otherwise
  - **daily** – Rotates the log files on a daily schedule. This can also be done on a weekly or monthly schedule
  - **create** – Instantiates a log file where the owner and group are a root user

# Logging related commands

1) dmesg . the dmesg kernel ring buffer utility
2) faillog, the faillog command (and also the faillog configuration file via man 5 faillog)
3) grep , the grep pattern searching utility
4) head , the head utility
5) klogd, the kernel log daemon (klogd)
6) last , the last command which shows last logged in users
7) less , the less paging utility
8) logger , the logger command-line interface to syslog utility
9) logrotate , the the logrotate utility
10) savelog , the savelog log file saving utility
11) syslogd , the system log daemon (syslogd)
12) syslog.conf , the syslogd configuration file
13) Tail , the tail utility

# IT601 – System and Network Administration

## < End>

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

IT601 – System and Network Administration

# Shell Scripts

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Writing and Editing Files

- Vim is an acronym for Vi IMproved. It is a free and open-source cross-platform text editor. It was first released by Bram Moolenaar in 1991 for UNIX variants.

- Vim is based on the original Vi editor, which was created by Bill Joy in 1976. In the 90's, it started becoming clear that Vi was lacking in some features when compared with the Emacs editor.

- VIM includes all the missing features of VI.

- VIM is generally preinstalled with many linux distributions, if not  it can be installed as below

```
sudo apt-get update
sudo apt-get install vim
```

# Vim Modes

- Everything in Vim is considered a mode. You can achieve whatever you want if you understand modes in Vim.
- There are many modes in Vim. But, we'll be looking at the 4 most important modes.

| Command Mode | Insert Mode | Command-Line Mode | Visual Mode |
|---|---|---|---|
| ▪ Default mode, also called Normal mode. | ▪ to edit the contents of the file. | ▪ To execute commands | ▪ to visually select some text and run commands over that section of code |
| ▪ To switch from one mode to another, you have to come to Command Mode | ▪ You can switch to insert mode by pressing i from command mode. | ▪ the commands in this mode are prefixed with a colon (:) | ▪ switch to this mode by pressing v from the command mode. |
| ▪ The commands that you run without any prefix (colon) indicate that you're running the command in command mode. | ▪ You can use the Esc key to switch back to command mode. | ▪ Switch to this mode by pressing : (colon) in command mode | |

# VIM Commands

## Insert mode commands

- a Append text following current cursor position
- A Append text to the end of current line
- i Insert text before the current cursor position
- I Insert text at the beginning of the cursor line
- o Open up a new line following the current line and add text there
- O Open up a new line in front of the current line and add text there

## Command mode commands

- Ctrl + e - 1 line up
- Ctrl + d - 1/2 page up
- Ctrl + f - 1 page up
- Ctrl + y - 1 line down
- Ctrl + u - 1/2 page down
- Ctrl + b - 1 page down
- % - use with '{','}','(',')' to jump with the matching one.
- 0 - first column of the line
- $ - jump to the last character of the line

# VIM Commands

- **Editing Commands**
  - d …delete the characters from the cursor position up the position given by the next command
  - c …change the character from the cursor position up to the position indicated by the next command.
  - y …copy the characters from the current cursor position up to the position indicated by the next command.
  - p …paste previous deleted or yanked (copied) text after the current cursor position.

Note: Doubling d, c or y operates on the whole line, for example yy copies the whole line.

- **Undo and Redo**
  - u - you can undo almost anything using u in the command mode.
  - Ctrl+r - undo is undoable using Ctrl-r.
- **Searching and Replacing**
  - :s/old/new/gc
  - :s/old/new/g
- **Save the file**
  - :wq , Save file and exit
  - :q! , Exit file without saving the changes

# First Script

- Create a new script file with name : myfirstScript.sh

  Vi myfirstScript.sh

- Write the following content

```
#!/bin/sh

# Author : IT601
# Copyright (c) Virtual University of Pakistan

echo “Hello Virtual University Student, What is your student iD”

read VUID

echo “WELCOME !, $VUID"
```

- Make the script executable

  Chmod 777 myfirstScript.sh

- Run script

  ./myfirstScript.sh

# Variables

- **Variable Names**

  The name of a variable can contain only letters (a to z or A to Z), numbers ( 0 to 9) or the underscore character ( _). By convention, Unix shell variables will have their names in UPPERCASE.

- **Defining Variables**

  VAR_NAME=variable_value

- **Accessing Values**

  echo $VAR_NAME

- **Read-only Variables**

  readonly VAR_NAME

- **Unsetting Variables**

  unset VAR_NAME

- **Variable Types**

  - Local Variables
  - Environment Variables
  - Shell Variables

# Special variables

- $0 - The filename of the current script.
- $n - These variables correspond to the arguments with which a script was invoked. Here n is a positive decimal number corresponding to the position of an argument.
- $# - The number of arguments supplied to a script.
- $* - All the arguments are double quoted. If a script receives two arguments, $* is equivalent to $1 $2.
- $@ - All arguments are individually double quoted. If script receives two arguments, $@ is equivalent to $1 $2.
- $? - The exit status of the last command executed.
- $$ - The process number of the current shell. This is the process ID under which they are executing.
- $! - The process number of the last background command.

# Defining Array Values

- **Basic Syntax**

  array_name[index]=value

- **For the ksh shell, here is the syntax of array initialization**

  set -A array_name value1 value2 ... valuen

- **For the bash shell, here is the syntax of array initialization**

  array_name=(value1 ... valuen)

- **Accessing Array Values**

  ${array_name[index]}

# Operators

## Arithmetic Operators

| | | |
|---|---|---|
| + (Addition) | Adds values on either side of the operator | `expr $a + $b` will give 30 |
| - (Subtraction) | Subtracts right hand operand from left hand operand | `expr $a - $b` will give -10 |
| * (Multiplication) | Multiplies values on either side of the operator | `expr $a \* $b` will give 200 |
| / (Division) | Divides left hand operand by right hand operand | `expr $b / $a` will give 2 |
| % (Modulus) | Divides left hand operand by right hand operand and returns remainder | `expr $b % $a` will give 0 |
| = (Assignment) | Assigns right operand in left operand | a = $b would assign value of b into a |
| == (Equality) | Compares two numbers, if both are same then returns true. | [ $a == $b ] would return false. |
| != (Not Equality) | Compares two numbers, if both are different then returns true. | [ $a != $b ] would return true. |

## Relational Operators

| | | |
|---|---|---|
| **-eq** | Checks if the value of two operands are equal or not; if yes, then the condition becomes true. | [ $a -eq $b ] is not true. |
| **-ne** | Checks if the value of two operands are equal or not; if values are not equal, then the condition becomes true. | [ $a -ne $b ] is true. |
| **-gt** | Checks if the value of left operand is greater than the value of right operand; if yes, then the condition becomes true. | [ $a -gt $b ] is not true. |
| **-lt** | Checks if the value of left operand is less than the value of right operand; if yes, then the condition becomes true. | [ $a -lt $b ] is true. |
| **-ge** | Checks if the value of left operand is greater than or equal to the value of right operand; if yes, then the condition becomes true. | [ $a -ge $b ] is not true. |
| **-le** | Checks if the value of left operand is less than or equal to the value of right operand; if yes, then the condition becomes true. | [ $a -le $b ] is true. |

# Operators

## Boolean Operators

| | | |
|---|---|---|
| **!** | This is logical negation. This inverts a true condition into false and vice versa. | [ ! false ] is true. |
| **-o** | This is logical OR. If one of the operands is true, then the condition becomes true. | [ $a -lt 20 -o $b -gt 100 ] is true. |
| **-a** | This is logical AND. If both the operands are true, then the condition becomes true otherwise false. | [ $a -lt 20 -a $b -gt 100 ] is false. |

## String Operators

| | | |
|---|---|---|
| **=** | Checks if the value of two operands are equal or not; if yes, then the condition becomes true. | [ $a = $b ] is not true. |
| **!=** | Checks if the value of two operands are equal or not; if values are not equal then the condition becomes true. | [ $a != $b ] is true. |
| **-z** | Checks if the given string operand size is zero; if it is zero length, then it returns true. | [ -z $a ] is not true. |
| **-n** | Checks if the given string operand size is non-zero; if it is nonzero length, then it returns true. | [ -n $a ] is not false. |
| **str** | Checks if str is not the empty string; if it is empty, then it returns false. | [ $a ] is not false. |

# Operators

- File Test Operators

| | | |
|---|---|---|
| **-b file** | Checks if file is a block special file; if yes, then the condition becomes true. | [ -b $file ] is false. |
| **-c file** | Checks if file is a character special file; if yes, then the condition becomes true. | [ -c $file ] is false. |
| **-d file** | Checks if file is a directory; if yes, then the condition becomes true. | [ -d $file ] is not true. |
| **-f file** | Checks if file is an ordinary file as opposed to a directory or special file; if yes, then the condition becomes true. | [ -f $file ] is true. |
| **-g file** | Checks if file has its set group ID (SGID) bit set; if yes, then the condition becomes true. | [ -g $file ] is false. |
| **-k file** | Checks if file has its sticky bit set; if yes, then the condition becomes true. | [ -k $file ] is false. |
| **-p file** | Checks if file is a named pipe; if yes, then the condition becomes true. | [ -p $file ] is false. |
| **-t file** | Checks if file descriptor is open and associated with a terminal; if yes, then the condition becomes true. | [ -t $file ] is false. |
| **-u file** | Checks if file has its Set User ID (SUID) bit set; if yes, then the condition becomes true. | [ -u $file ] is false. |
| **-r file** | Checks if file is readable; if yes, then the condition becomes true. | [ -r $file ] is true. |
| **-w file** | Checks if file is writable; if yes, then the condition becomes true. | [ -w $file ] is true. |
| **-x file** | Checks if file is executable; if yes, then the condition becomes true. | [ -x $file ] is true. |
| **-s file** | Checks if file has size greater than 0; if yes, then condition becomes true. | [ -s $file ] is true. |
| **-e file** | Checks if file exists; is true even if file is a directory but exists. | [ -e $file ] is true. |

# Control Statements

- The if...else statements

if...fi statement

```
if [ expression ]
then
   Statement(s) if true
fi
```

```
a=10
b=20
if [ $a == $b ]
then
   echo "a is equal to b"
fi
```

if...else...fi statement

```
if [ expression ]
then
   Statement(s) if true
else
   Statement(s) if not
true
fi
```

```
a=10
b=20
if [ $a == $b ]
then
   echo "a is equal to b"
else
   echo "a is not equal to b"
fi
```

if...elif...else...fi statement

```
if [ expression 1 ]
then
   Statement(s) to be executed if expression 1 is true
elif [ expression 2 ]
then
   Statement(s) to be executed if expression 2 is true
elif [ expression 3 ]
then
   Statement(s) to be executed if expression 3 is true
else
   Statement(s) to be executed if no expression is true
fi
```

```
a=10
b=20
if [ $a == $b ]
then
   echo "a is equal to b"
elif [ $a -gt $b ]
then
   echo "a is greater than b"
elif [ $a -lt $b ]
then
   echo "a is less than b"
else
   echo "None of the condition met"
fi
```

# Control Statements

## The case...esac Statement

```
case word in
   pattern1)
      Statement(s) to be executed if pattern1 matches
      ;;
   pattern2)
      Statement(s) to be executed if pattern2 matches
      ;;
   pattern3)
      Statement(s) to be executed if pattern3 matches
      ;;
   *)
     Default condition to be executed
     ;;
esac
```

```
FRUIT="kiwi"

case "$FRUIT" in
   "apple") echo "Apple pie is quite tasty."
   ;;
   "banana") echo "I like banana nut bread."
   ;;
   "kiwi") echo "New Zealand is famous for kiwi."
   ;;
esac
```

# Loops

## ➢ The while loop

```
while command
do
   Statement(s) to be executed if command
is true
done
```

```
a=0
while [ $a -lt 10 ]
do
   echo $a
   a=`expr $a + 1`
done
```

## ➢ The for loop

```
for var in word1 word2 ... wordN
do
   Statement(s) to be executed
for every word.
done
```

```
for var in 0 1 2 3 4 5 6 7 8 9
do
   echo $var
done
```

## ➢ The until loop

```
until command
do
   Statement(s) to be executed
until command is true
done
```

```
a=0

until [ ! $a -lt 10 ]
do
   echo $a
   a=`expr $a + 1`
done
```

## ➢ The select loop

```
select var in word1 word2 ... wordN
do
   Statement(s) to be executed for
every word.
done
```

```
select K in tea cofee water juice appe all none
do
   case $K in
      tea|cofee|water|all)
         echo "Go to canteen"
         ;;
      juice|appe)
         echo "Available at home"
      ;;
      none)
         break
      ;;
      *) echo "ERROR: Invalid selection"
      ;;
   esac
done
```

# Nesting while Loops

```
while command1 ; # this is loop1, the outer loop
do
   Statement(s) to be executed if command1 is true

   while command2 ; # this is loop2, the inner loop
   do
      Statement(s) to be executed if command2 is true
   done

   Statement(s) to be executed if command1 is true
done
```

```
a=0
while [ "$a" -lt 10 ]    # this is loop1
do
   b="$a"
   while [ "$b" -ge 0 ]  # this is loop2
   do
      echo -n "$b "
      b=`expr $b - 1`
   done
   echo
   a=`expr $a + 1`
done
```

- The break Statement
- The continue statement
- Substitution

```
a=10
echo -e "Value of a is $a \n"
```

- Escape Sequences

| | |
|---|---|
| **** | backslash |
| **\a** | alert (BEL) |
| **\b** | backspace |
| **\c** | suppress trailing newline |
| **\f** | form feed |
| **\n** | new line |
| **\r** | carriage return |
| **\t** | horizontal tab |
| **\v** | vertical tab |

# Creating Functions

## Syntax

```
function_name () {
   list of commands
}
```

## Simple Function

```
# Define your function here
Hello () {
   echo "Hello World"
}
```

```
# Invoke your function
Hello
```

## Passing Parameters

```
# Define your function here
Hello () {
   echo "Hello World $1 $2"
}
```

```
# Invoke your function
Hello test1 test 2
```

## Returning Data

```
# Define your function here
Hello () {
   echo "Hello World $1 $2"
   return 10
}
```

```
# Invoke your function
Hello Zara Ali

# Capture value returnd by last command
ret=$?

echo "Return value is $ret"
```

# IT601 – System and Network Administration

## < End>

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

IT601 – System and Network Administration

# Servers

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Introdcution

A group of computer machines in an organization  generall referd to as server.

It is used to provide different types of services.

- Application services
- Web services
- Back-office processing
- Databases
- Batch computation
- Etc.

# Servers vs Services

**A *server* is a computer machine consisting of.**

- **Hardware**

CPU


Memory

Storage

Power Supply


Network Interface


Motherboard


Peripheral Devices


Rack

- **Operating System**

Linux Server

Windows Server

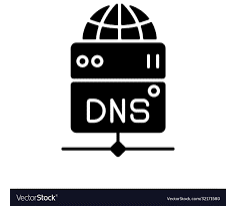
UNIX

- **Software packages**

MySQL

DHCP SERVER

ORACLE DATABASE



# How are Servers different?

- 1000s of clients depend on server.
- Requires high reliability.
- Requires tighter security.
- Often expected to last longer.
- Investment amortized over many clients, longer lifetime.

IT601 – System and Network Administration

# Server CPU

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# CPUs

**There are two main types of server processors:**

**X86.**

- X86 processors are the most common type of processor found in servers.
- They are made by companies such as Intel and AMD.
- X86 processors are designed for general-purpose computing.
- They can be used for a variety of tasks, including web hosting, database management, and file sharing.


**CPU**

**RISC**

- RISC processors are designed for specific tasks.
- They are often used in high-performance servers.
- RISC processors are made by companies such as IBM and Oracle.

# Selection of CPU

- The type of server processor you need depends on the type of server you are using.
- If you are using a general-purpose server, an x86 processor is likely the best choice.
- If you are using a high-performance server, a RISC processor may be the better choice.

## Clock speed

- keep up with the demands of modern businesses
- ablity to process large amounts of data quickly and efficiently

## Cores

- number of cores in a processor can have a big impact on its performance.
- More cores means that the processor can handle more tasks at the same time.
- in the market today have up to 32 cores and more

## Memory support

- support large amounts of memory
- need to store and process large amounts of data

# Selection of CPU

## Expand-ability

- Need to be expandable so that businesses can add more features as their needs change.
- Built-in features such as security or management tools
- Expansion slots so that businesses can add more features as they need them

## Efficient Data Management

- Data management is a key concern
- Must be able to efficiently handle large amounts of data
  - keep the server running smoothly, even when under heavy load

## Cost and Power Consumption

- Most important factors to consider is cost
  - Initial Cost
  - Operational Cost
- energy-efficient
- Right balance between power consumption and performance to minimize your carbon footprint

# Selection of CPU

| Budget | Workload |
| --- | --- |
| • key factors to consider is your budget | • The server load |

Source : https://dtc1.com/guide-for-server-processors/

# SERVER CPU Choices

## 1- Intel Xeon

(a) - Multicore with two threads per core, 1.8 to 3.3 Ghz, 8 cores
(b) - upto 18 MB L3 Cache

## 2 - AMD Opteron

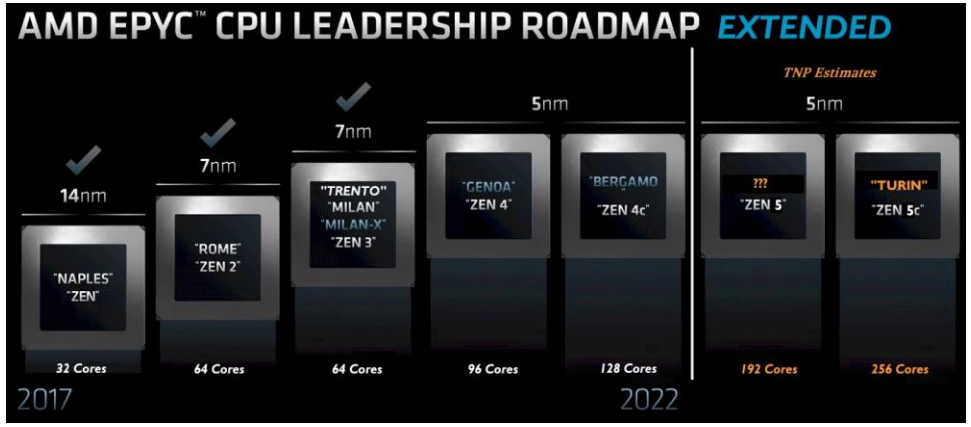
(a) - 4, 6, 8, or 12 cores @ 1.4 to 3.2 GHz
(b) - Up to 12 MB L3 cache

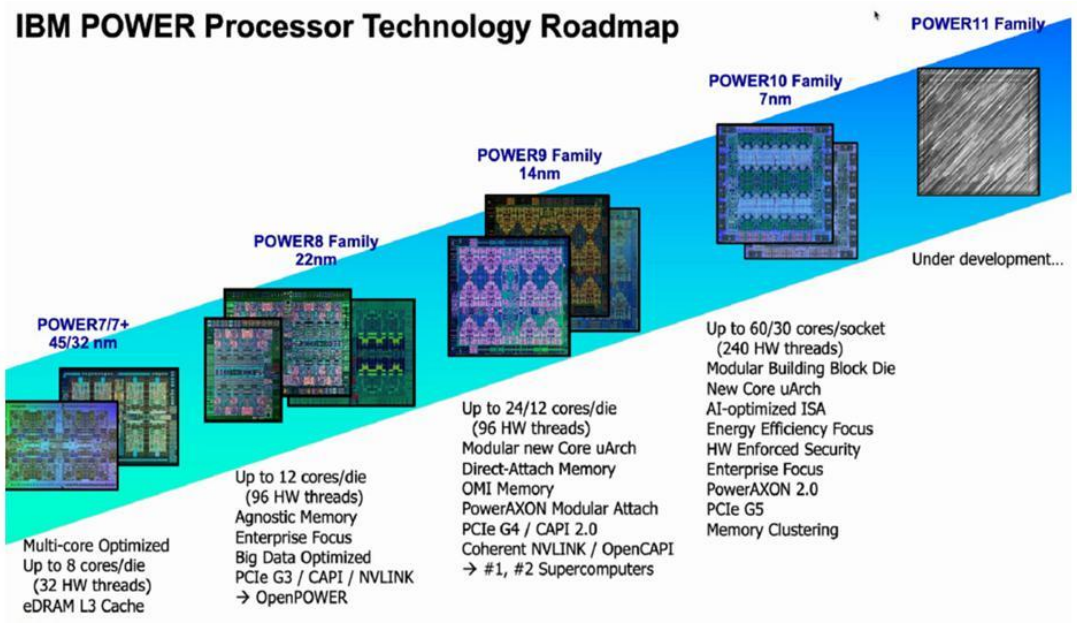
## 3 - IBM Power 7/8/9/10...

(a) - 4, 6, or 8 cores with 4 threads each @ 3.0 to 4.25 GHz
(b) - 4 MB L3 cache per core (up to 32MB for 8-core)

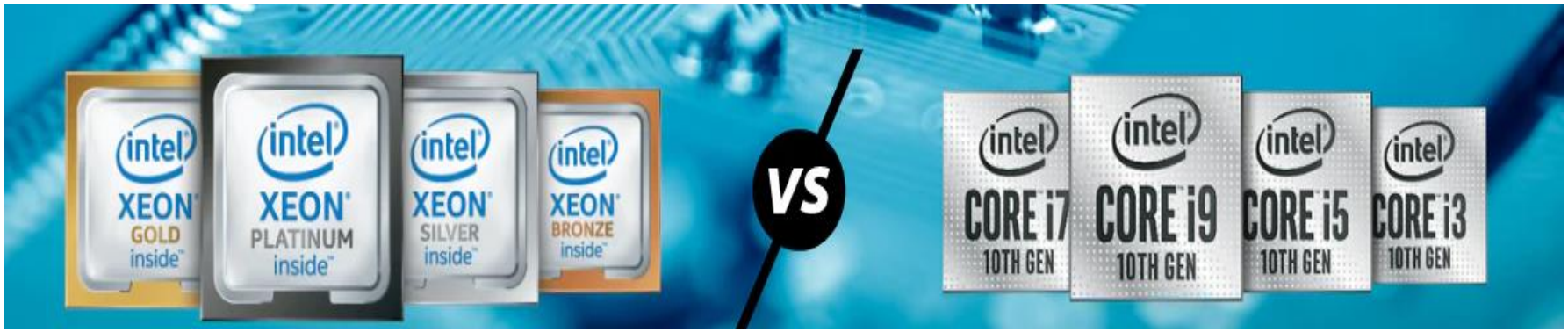
## 4 - Sun Niagara 3

(a) - 16 cores with 8 threads each @ 1.67 GHz
(b) - 6 MB L2 cache



# Xeon vs Pentium/Core CPUs




Xeon based on Pentium/Core with changes that vary by model:

- Allows more CPUs
- Has more cores
- Better hyper-threading
- Faster/larger CPU caches
- Faster/larger RAM support

# Intel Processor Families

**1) Intel® Xeon® Scalable Processors**
4th Gen Intel® Xeon® Scalable processors have the most built-in accelerators of any CPU on the market to improve performance in AI, analytics, networking, storage, and HPC.

**1) Intel® Xeon® Max Series**
Maximize bandwidth with the Intel® Xeon® CPU Max Series, the first and only x86-based processor with high-bandwidth memory (HBM).

**1) Intel® Xeon® W Processor**
Designed for creative professionals, delivering the performance you need for VFX, 3D rendering, and 3D CAD on a workstation.

**1) Intel® Xeon® D Processor**
When space and power are at a premium, these innovative system-on-a-chip processors bring workload optimized performance.

**1) Intel® Xeon® E Processor**
Essential, business-ready performance, expandability and reliability for entry server solutions.

IT601 – System and Network Administration

# Server Memory

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Memory

**Server memories are concerned witn**

- RAM
- CPU Cache

**Two Major aspects of server memory are**

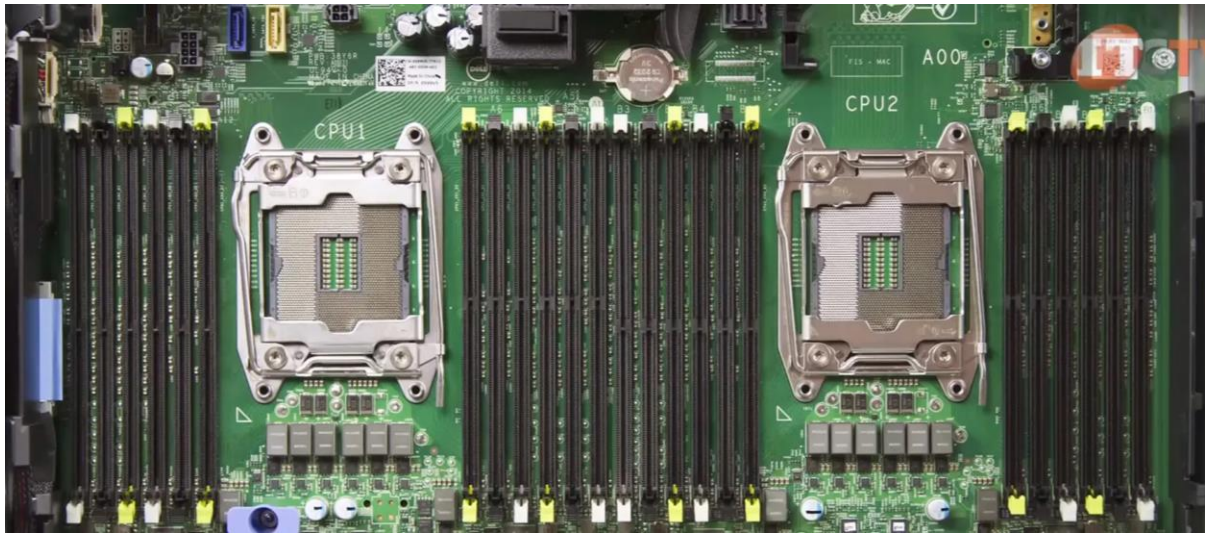
- Large Capacity
- Higher speed

**x86 supports up to 64GB with PAE.**

**x86-64 supports 1 PB (1024 TB)**

**Servers need faster RAM than desktops.**

- Higher memory speeds.
- Multiple DIMMs accessed in parallel.
- Larger CPU caches.





# Types of Memory

**FBDIMM**
– Fully buffered

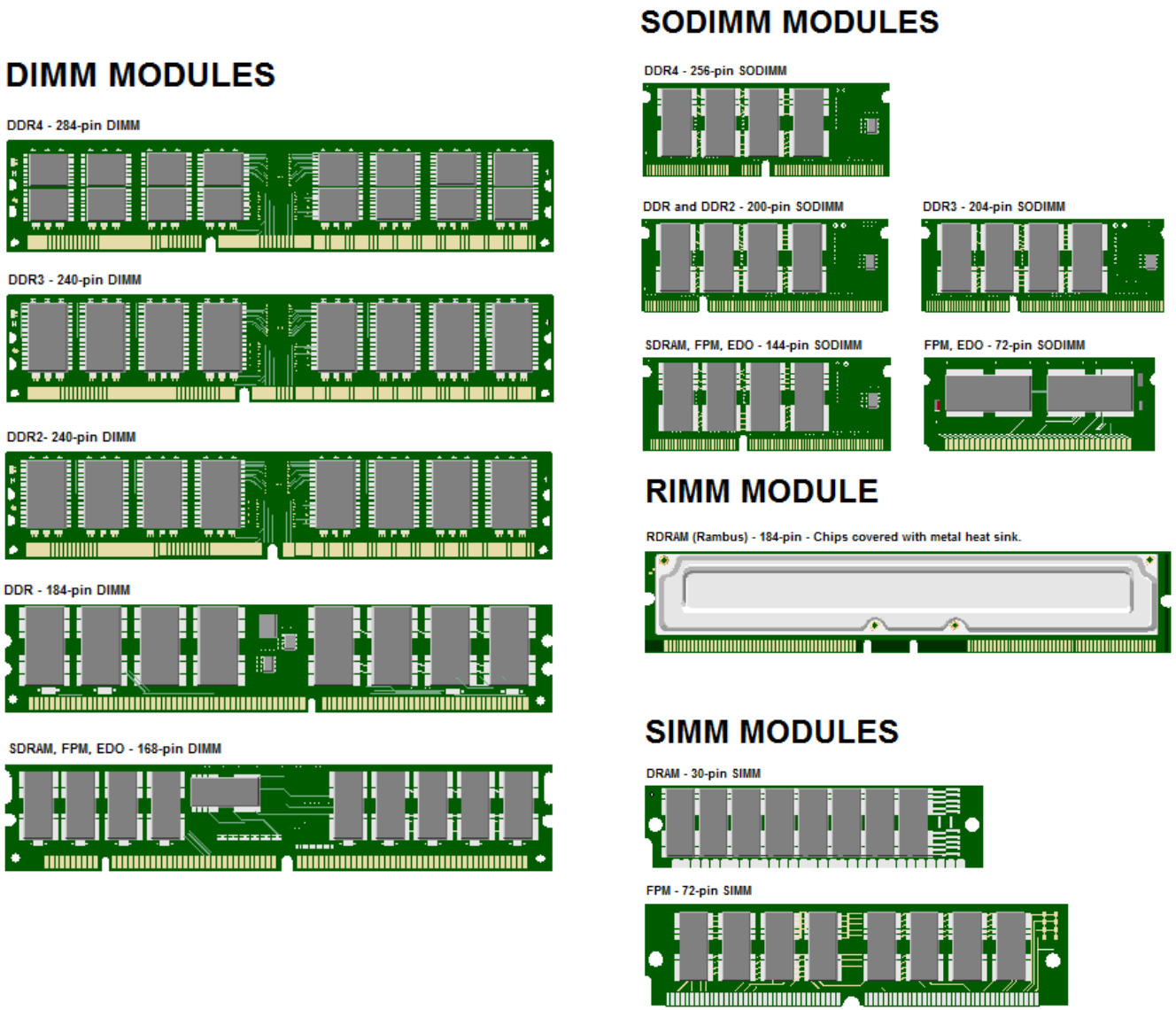
**UDIMM**
- Unbuffered dual in-line memory module

**RDIMM**
- Registered dual in-line memory module

**LRDIMM**
- Load Reduced dual in-line memory module

**SODIMM**
- Used with laptops

**MicroDIMM**




*Source : https://www.pcmag.com/encyclopedia/term/memory-module*

# Transfer Speed of RAM

**SDRAM**
- Synchronous Dynamic Random Access Memory

**DDR**
- Double Data Rate SDRAM

**DDR2**
– 2 x times the DDR

**DDR3**
low power, Twice clock multiplier with a four times clock multiplier , No F/B Compatibility, 2 x DDR2 TT

**DDR4**
- low power higher module density and lower voltage requirements, coupled with higher data rate transfer speeds

| DDR SDRAM Standard | Internal rate (MHz) | Bus clock (MHz) | Prefetch | Data rate (MT/s) | Transfer rate (GB/s) | Voltage (V) |
|---|---|---|---|---|---|---|
| SDRAM | 100-166 | 100-166 | 1n | 100-166 | 0.8-1.3 | 3.3 |
| DDR | 133-200 | 133-200 | 2n | 266-400 | 2.1-3.2 | 2.5/2.6 |
| DDR2 | 133-200 | 266-400 | 4n | 533-800 | 4.2-6.4 | 1.8 |
| DDR3 | 133-200 | 533-800 | 8n | 1066-1600 | 8.5-14.9 | 1.35/1.5 |
| DDR4 | 133-200 | 1066-1600 | 8n | 2133-3200 | 17-21.3 | 1.2 |

IT601 – System and Network Administration

# Cache Memory

Arif Husen

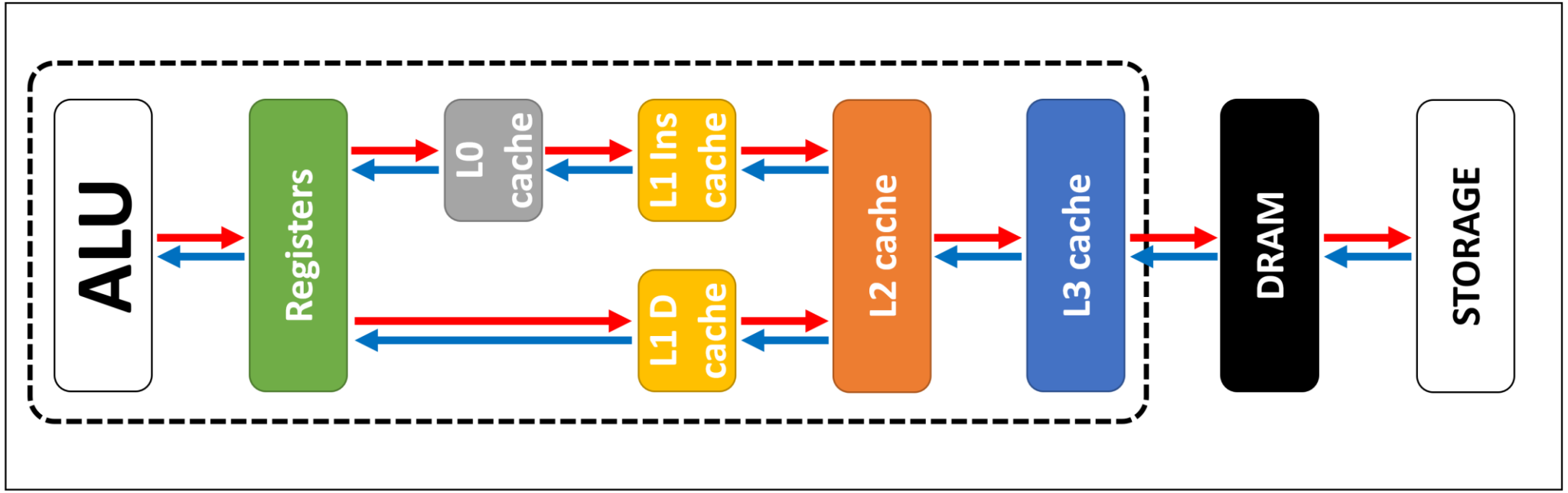
Department of Computer Science and Information Technology,

Virtual University of Pakistan

# The L1, L2, L3 Cache

Cache memory is high speed  memories placed between RAM  and CPU.

RAM -> L3 -> L2 – L1->  Registers - > CPU




Source : https://www.techspot.com/article/2066-cpu-l1-l2-l3-cache/

IT601 – System and Network Administration

# System Bus

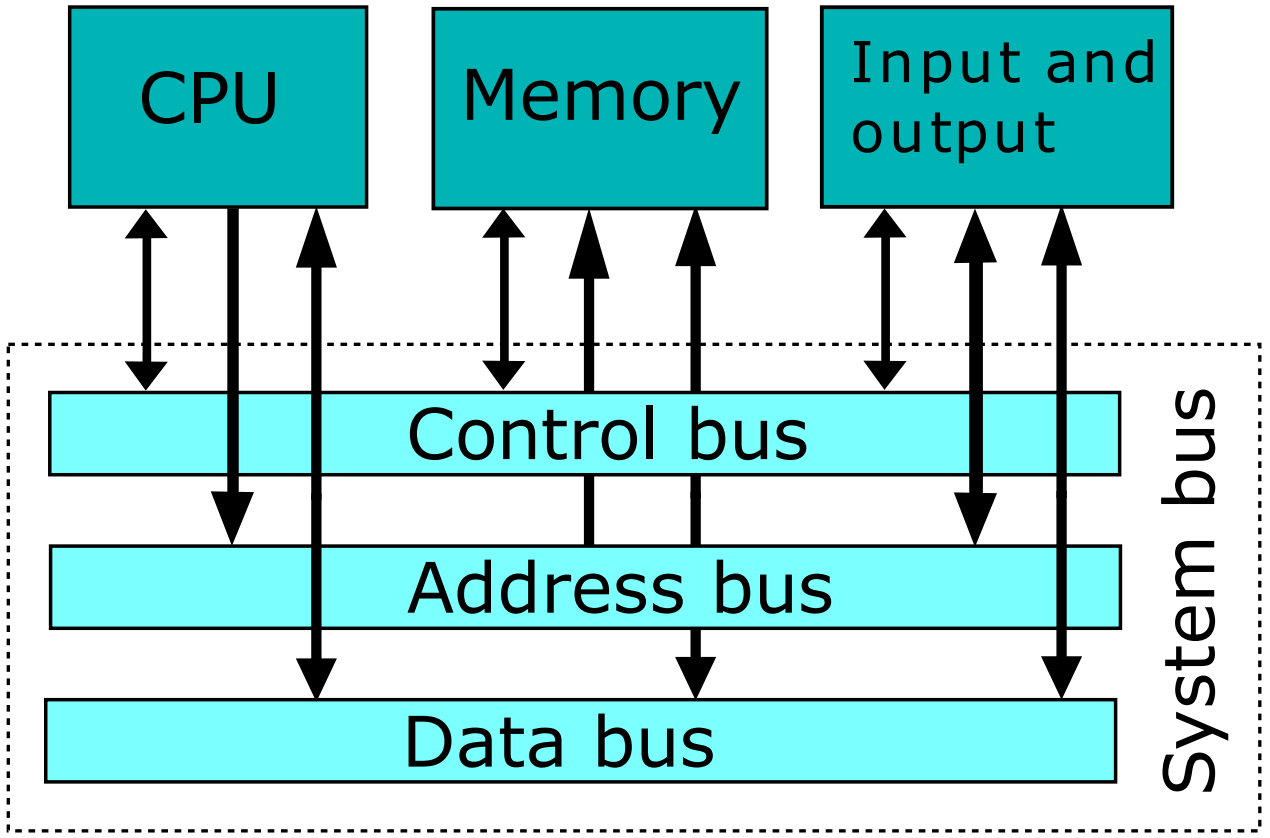
Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# System Buses

- **Data sharing**
- **Addressing**
- **Power**
- **Timing**



# Expansion Bus Types

common expansion bus types are

- **ISA**- Industry Standard Architecture
  - designed for use in the original IBM PC
  - 8 bit / 16 bit
  - The ISA bus ran at a clock speed of 4.77 MHz and improved version 8MHz.
  - 16-bit version of the ISA bus is sometimes known as the AT bus (AT-Advanced Technology).
- **MCA -** Micro Channel Architecture
  - IBM developed this bus as a replacement for ISA when they designed the PS/2 PC in 1987
  - speed of 10MHz and supported either 16-bit or 32-bit data
- **EISA - Extended Industry Standard Architecture**
  - as an alternative to MCA
  - use a 32-bit data path and provided 32 address lines, giving access to 4GB of memory
  - ran at 8MHz in order for it to be compatible with ISA

# Expansion Bus Types

common expansion bus types are

- **VESA -** Video Electronics Standards Association
  - invented to help standardize PCs video specifications
  - 32-bit data path and ran at 25 or 33 MHZ
  - VL-Bus was superseded by PC
- **PCMCIA -** Personal Computer Memory Card Industry Association (Also called PC bus)
- **AGP -** Accelerated Graphics Port
- **SCSI -** Small Computer Systems Interface
- **Universal Serial Bus (USB)**

# Expansion Bus Types

Servers need high I/O throughput.

- Fast peripherals: SCSI-3, Gigabit ethernet
- Often use multiple and/or faster buses.

PCI

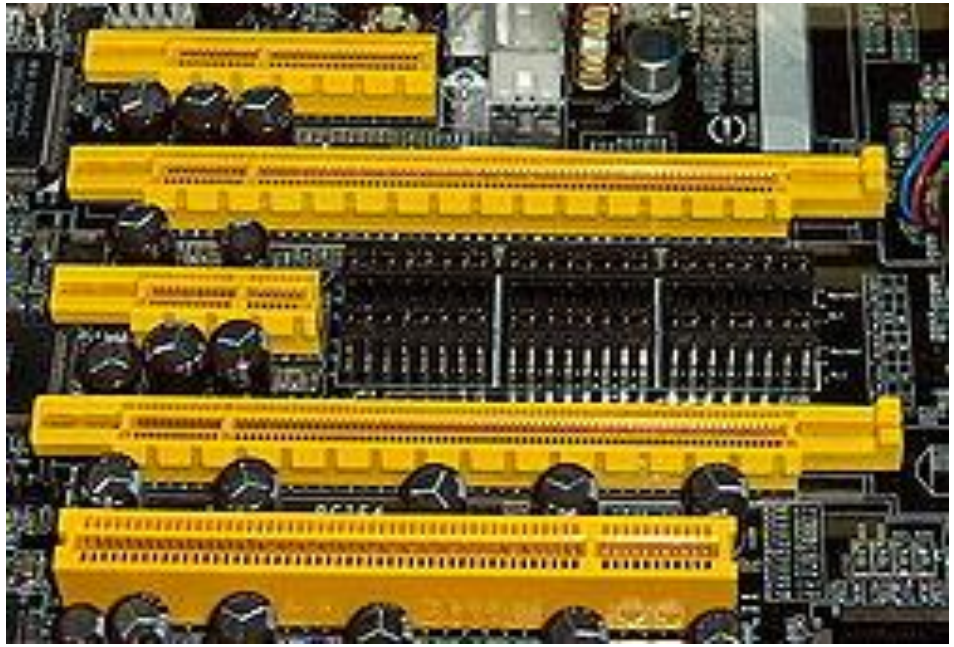
- Desktop: 32-bit 33 MHz, 133 MB/s
- Server: 64-bit 66 MHz, 533 MB/s

PCI-X (backward compatible)

- v1.0: 64-bit 133 MHz, 1.06 GB/s
- v2.0: 64-bit 533 MHz, 4.3 GB/s

PCI Express (PCIe)

- Serial architecture, v3.0 up to 16 GB/s



IT601 – System and Network Administration

# Power Supply

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Server Power Supply

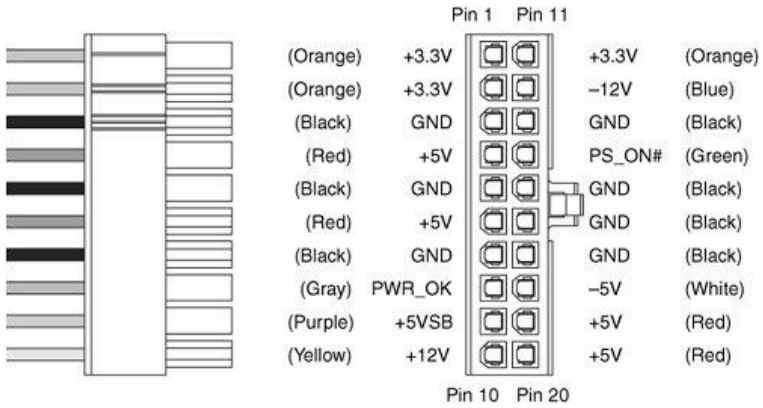
- Servers based on the ATX or microATX form factors generally use an ATX power supply

- Pedestal servers based on one of the Server System Infrastructure (SSI) form factors generally use an EPS12V power supply

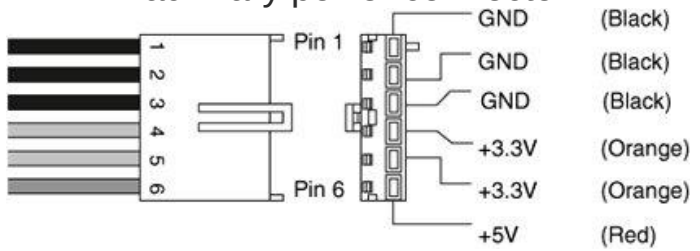
# ATX Power Supply Standards

- ATX power supplies were originally designed for use in desktop computers.
- Widely used in entry-level tower and slimline servers.
- ATX power supply standards include the following:
  - ATX version 2.03 Older entry-level tower servers
  - ATX12V More recent entry-level tower servers
  - ATX1U 1U slimline servers
  - ATX2U 2U slimline servers

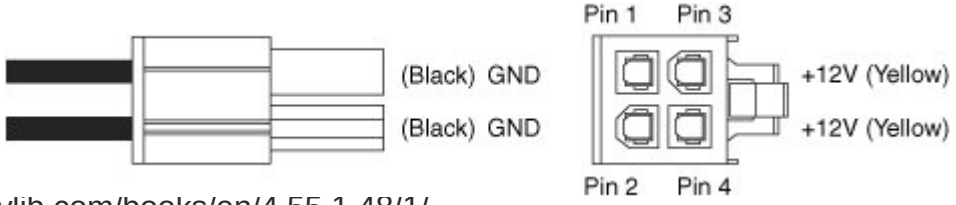
The ATX version 2.03 power supply connector




ATX auxiliary power connector.




ATX12V power connector

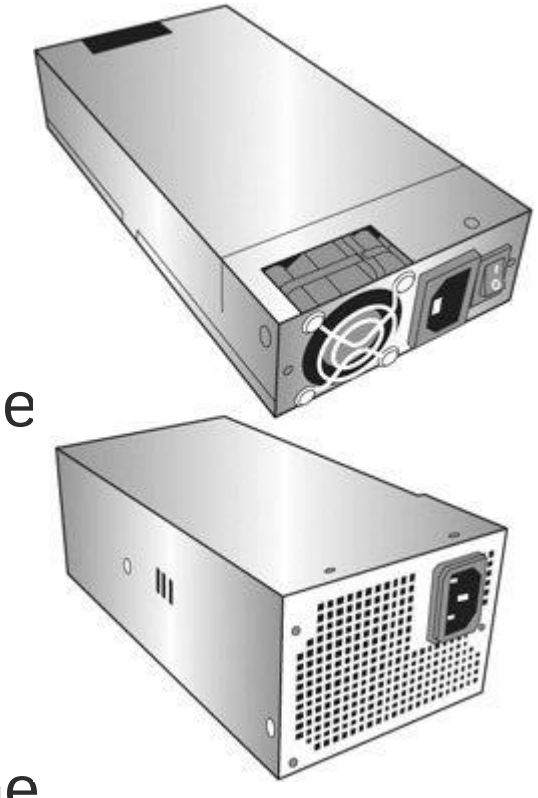



https://flylib.com/books/en/4.55.1.48/1/

# ATX1U/2U Rack-Mounted Power Supplies

- Rack-mounted servers use a variety of power supply standards.

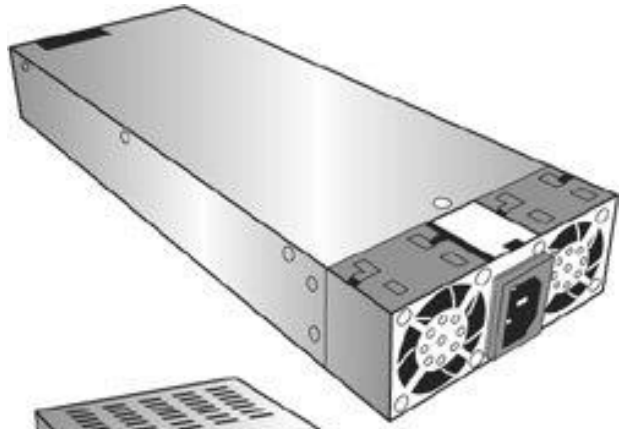
- Most 1U and 2U servers use the power supply standards developed by the SSI Forum, such as the ATX1U and ATX2U power supplies
  - The ATX1U and ATX2U power supply standards use the same 20-pin ATX power supply, floppy, and hard disk power connectors used by the ATX 2.03 and ATX12V v1.x power supply standards.
  - Some 200w and larger ATX1U power supplies also feature the 4-pin ATX12V power supply connector
  - ATX2U power supplies feature the 6-pin auxiliary power supply connector



# SSI Power Supplies

The SSI Forum has developed a series of power supply and connector form factors designed for use in various types of servers.

- Pedestal-mounted servers
  - EPS12V
  - ERP12V
- 1U rack-mounted servers
  - EPS1U
- 2U rack-mounted servers
  - EPS2U
  - ERP2U



EPS1U

EPS2U

IT601 – System and Network Administration

# Storage

Arif Husen

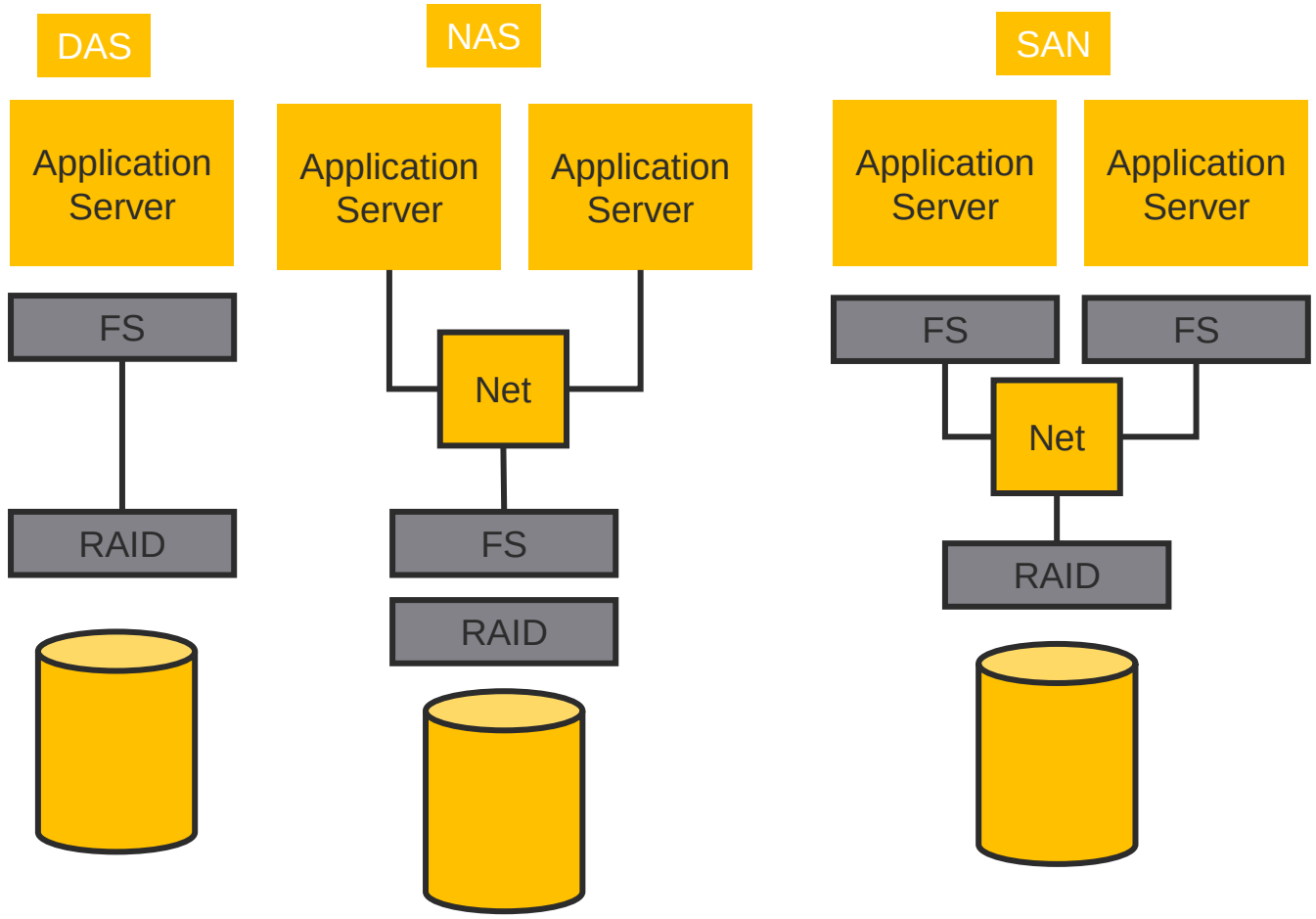
Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Storage Types

Four Types of Storage Solutions are :

- Direct-attached storage (DAS)
  - Conencted with SAS, SATA or
  - PCIe
- Network-attached storage (NAS)
- Storage area network (SAN)
- Cloud storage
  - Requires Internet



# Storage Types

- **Direct-attached storage**
  - HDD
    - Applications : Suitable for  Write Heavy Applications, Backups
    - Advantages : Cheap, High Capacity, Unlimited number of writes , Low Cost
    - Disadvantages : Frgile, Breaks down over time, Slow
  - SSD
    - Applications : General storage , Boot device
    - Advantages : Relatively cheap , Fast enough for most , Enough capacity for most
    - Disadvantages : Limited number of writes , May not be suitable for enterprise use
  - NVME SSD
    - Applications : High-performance computing , Boot drives , Databases
    - Advantages : Currently the fastest persistent server storage type on the market
    - Disadvantages : Can be very expensive , Limited number of writes

# Storage Types

- **Direct-attached storage**
  - HDD
    - Applications : Suitable for  Write Heavy Applications, Backups
    - Advantages : Cheap, High Capacity, Unlimited number of writes , Low Cost
    - Disadvantages : Frgile, Breaks down over time, Slow
  - SSD
    - Applications : General storage , Boot device
    - Advantages : Relatively cheap , Fast enough for most , Enough capacity for most
    - Disadvantages : Limited number of writes , May not be suitable for enterprise use
  - NVME SSD
    - Applications : High-performance computing , Boot drives , Databases
    - Advantages : Currently the fastest persistent server storage type on the market
    - Disadvantages : Can be very expensive , Limited number of writes

# Storage Types

- **Network-attached storage**
  - Applications : Provides access to the same data on multiple systems
  - Advantages : Can be very fast and very high capacity, Can mix SSDs and hard drives to make a faster solution than hard drives alone, RAID storage always
  - Disadvantages : Requires a separate computer, Expensive
- **Storage area network**
  - Applications : Sophisticated databases , Virtualization deployments , Large virtual desktop infrastructures (VDIs) , Enterprise resource management workloads
  - Advantages : Resilient, Removes single points of failure , Easily withstands failure of multiple components/devices
  - Disadvantages : Expensive , Complex

# Storage Types

- **Cloud Storage**
  - Applications : Off-site backups , Provides access to data from multiple locations
  - Advantages :Peace of mind
  - Disadvantages : Requires Internet connection , Usage rates may impact upload/download capacity

# SAS , SATA SSDs and NVME

➢ **Hard Disks:**

- Three RPM speeds: 7,200 RPM, 10,000 RPM, and 15,000 RPM
- Capacity varies, starting at about 300GB and going up to over 10TB
- SATA 6, SAS 6, and SAS 12 are typical connection speeds

➢ **SSDs**

- NAND flash memory cells and are always faster than hard drives, primarily because they do not have to “seek” the data on the disk, and latency is low
- SAS or SATA connector
- SATA 6, SAS 6, and SAS 12, SAS 24
- High Cost
- SSDs wear out as they are written to

# SAS , SATA SSDs and NVME

- **Hard Disks:**

- SAS and SATA bus interfaces were originally designed for slow, mechanical hard drives

- NVMe storage can connect via a wide range of connector types from M.2 to U.2 to U.3 to newer standards such as E1.S, also known as “ruler” SSDs

- connect over the PCIe bus, typically an x4 connection, though x2 and x8 are also possible

- The maximum speeds for Gen 3 PCIe and Gen 4 PCIe are 3.5GB/s and 7 GB/s, respectively

# RAID and its Levels

- RAID (redundant array of independent disks)
- A way of storing the same data in different places on multiple hard disks or solid-state drives (SSDs) to protect data in the case of a drive failure.
- There are different RAID levels, however, and not all have the goal of providing redundancy.

RAID 0

RAID 1

Raid 5

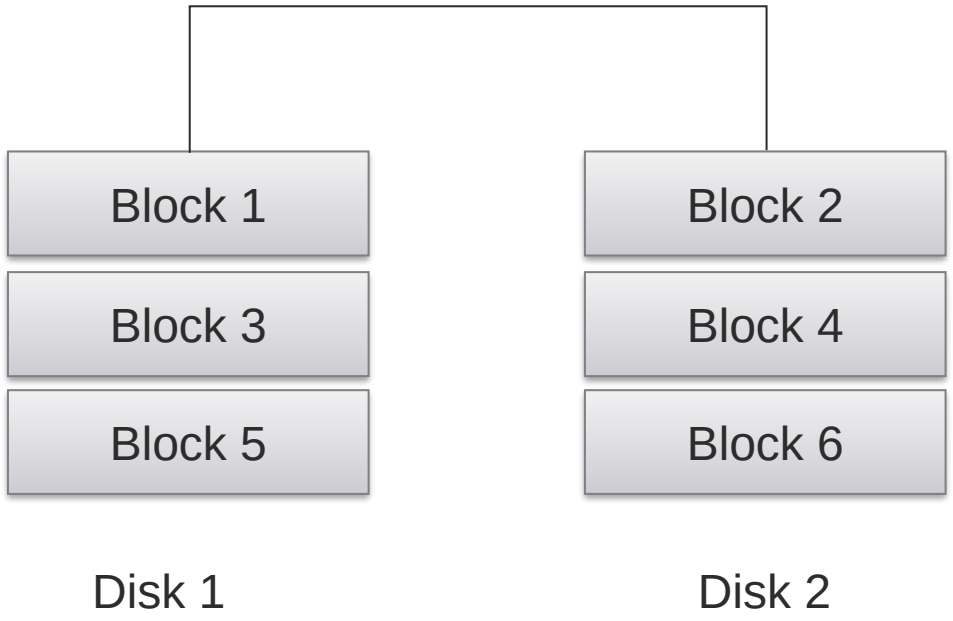
RAID 6

RAID 10 (RAID 1 + 0)

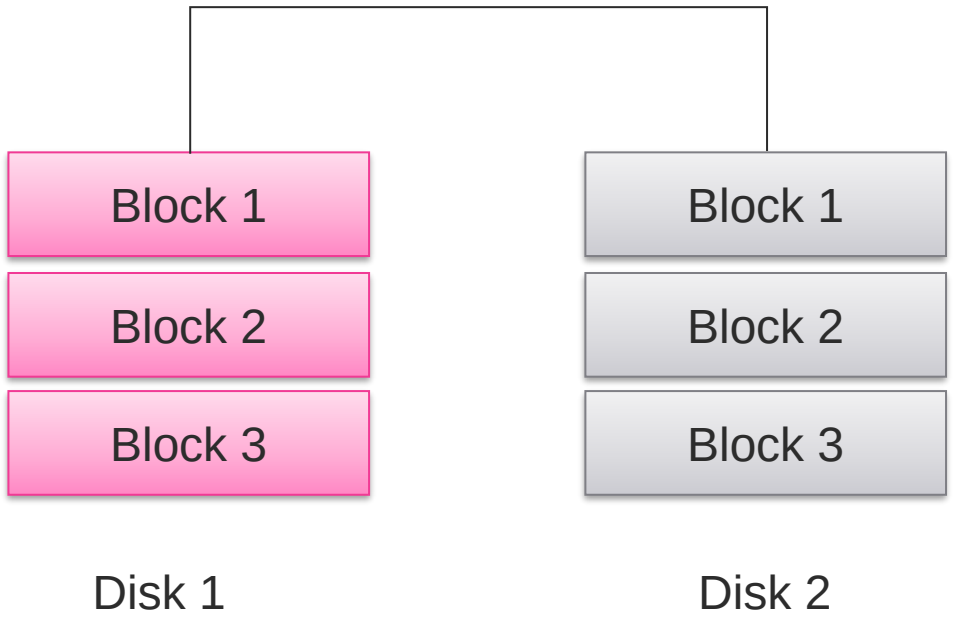
RAID 50 (RAID 5 + 0)

RAID 60 (RAID 6 + 0)

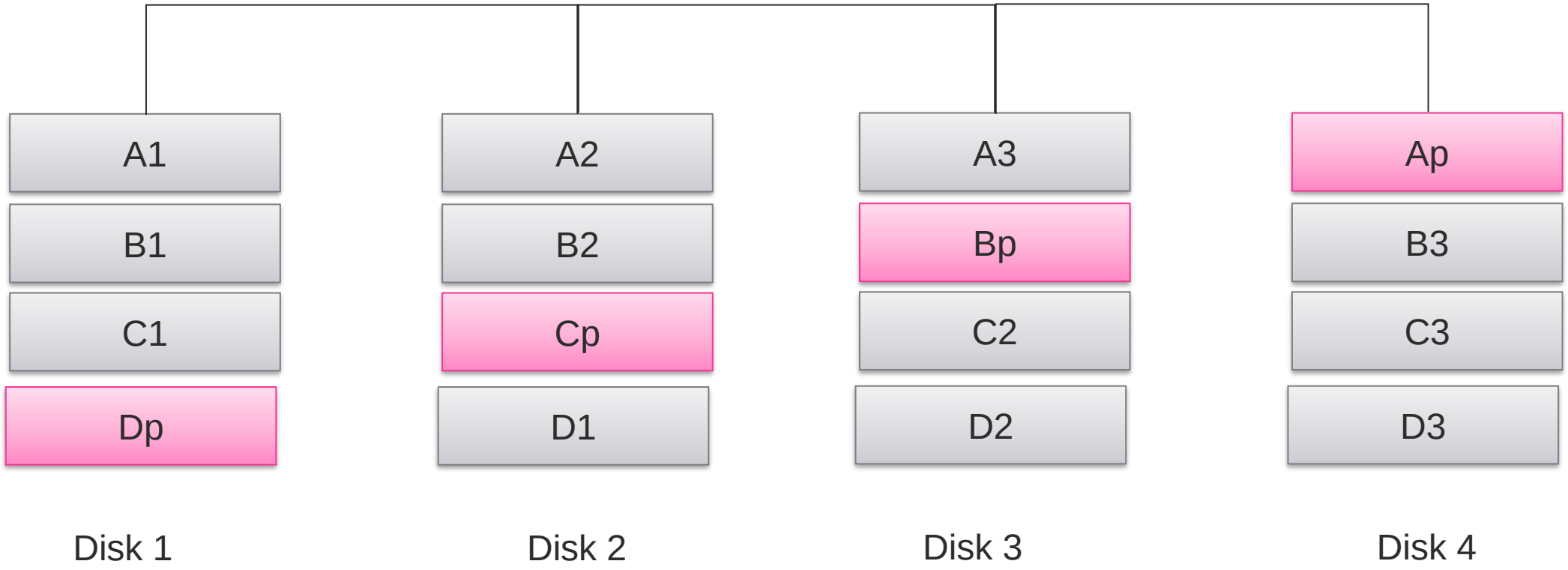
# RAID 0



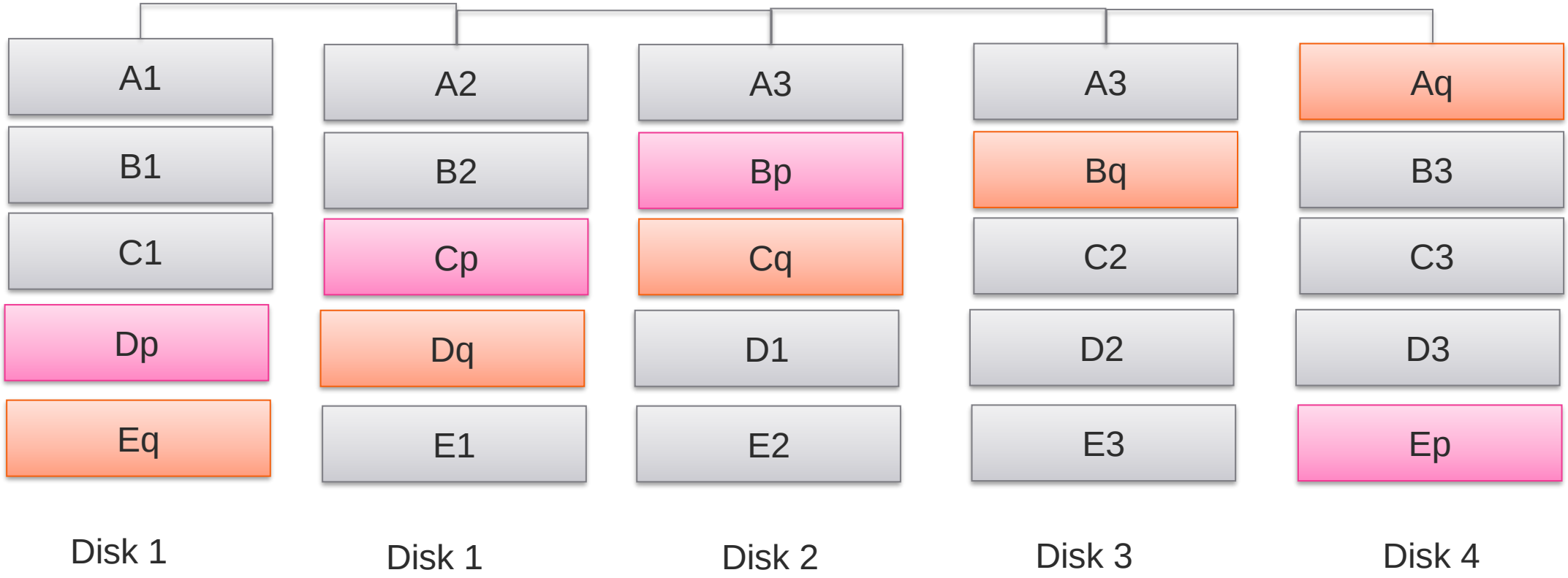
# RAID 1



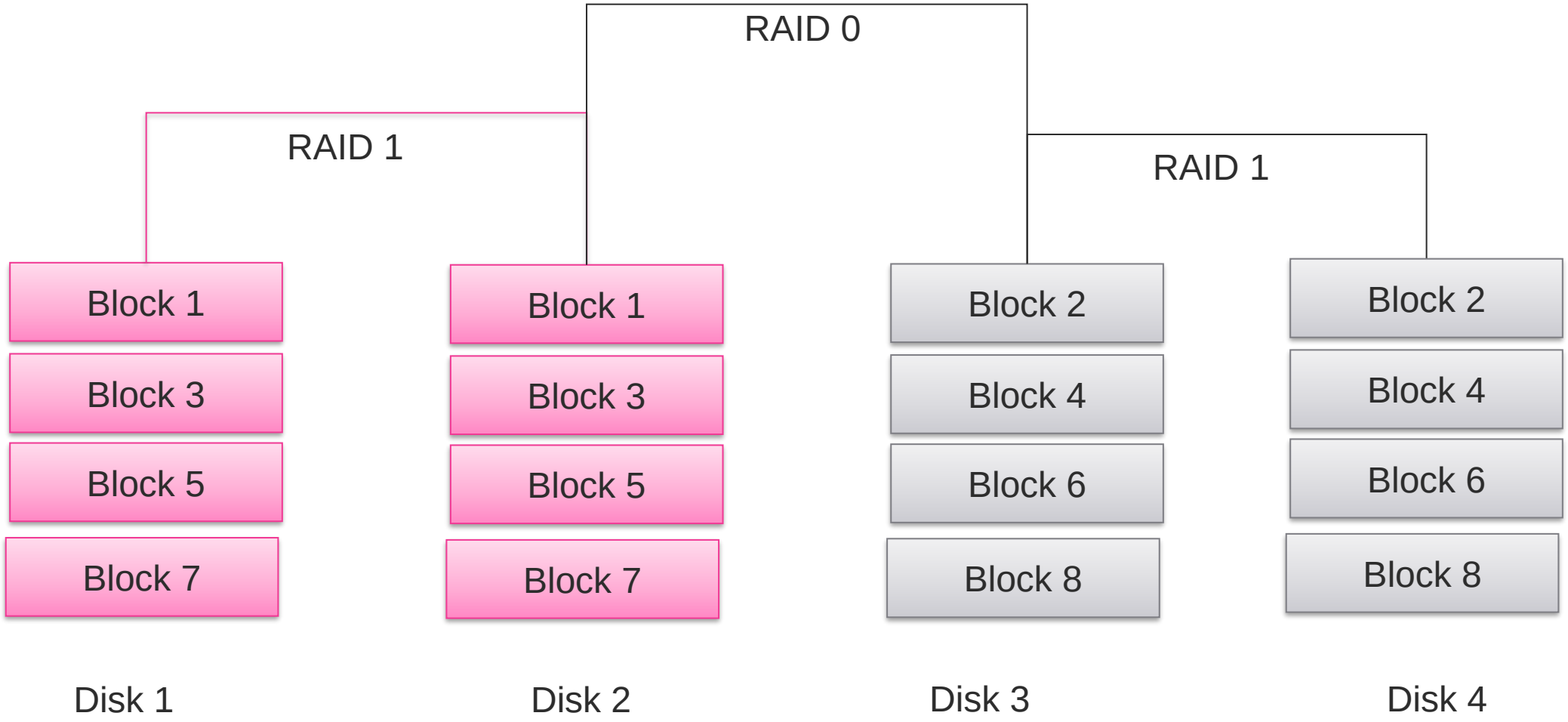
# RAID 5


A1
B1
C1
Dp
A2
B2
Cp
D1
A3
Bp
C2
D2
Ap
B3
C3
D3
Disk 1
Disk 2
Disk 3
Disk 4

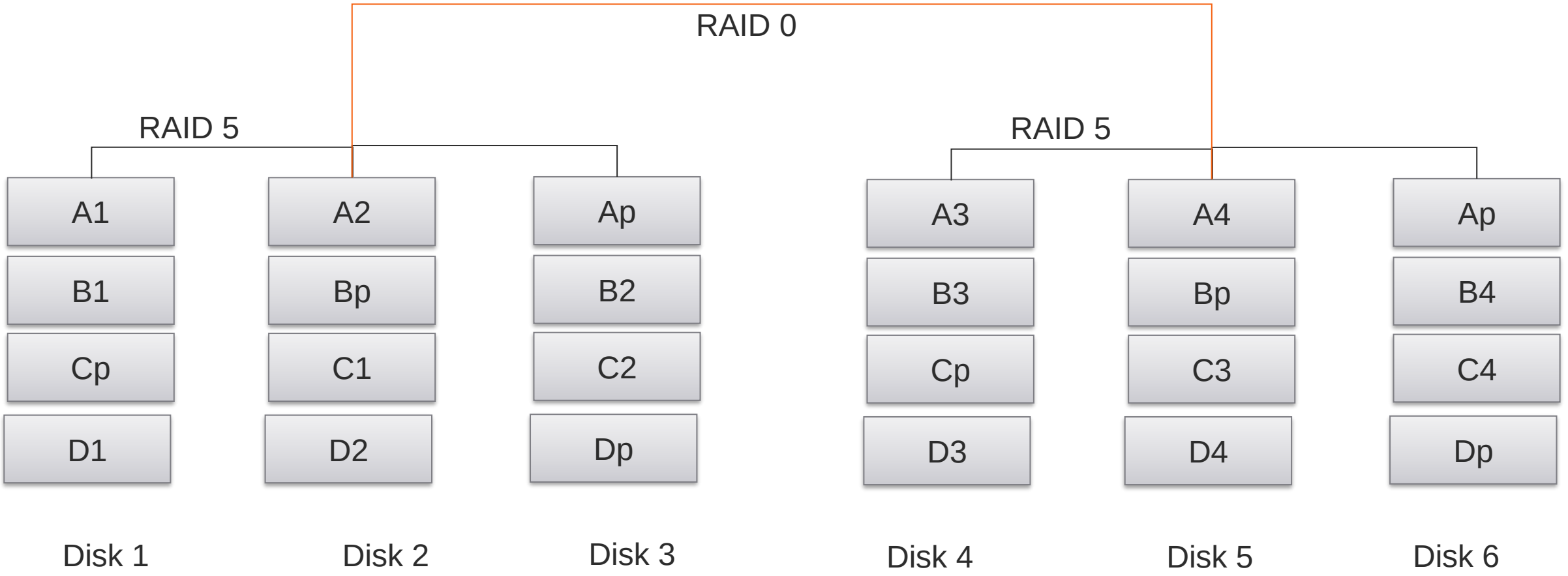
# RAID 6


A1
B1
C1
Dp
Eq
A2
B2
Cp
Dq
E1
A3
Bp
Cq
D1
E2
A3
Bq
C2
D2
E3
Aq
B3
C3
D3
Ep
Disk 1
Disk 1
Disk 2
Disk 3
Disk 4

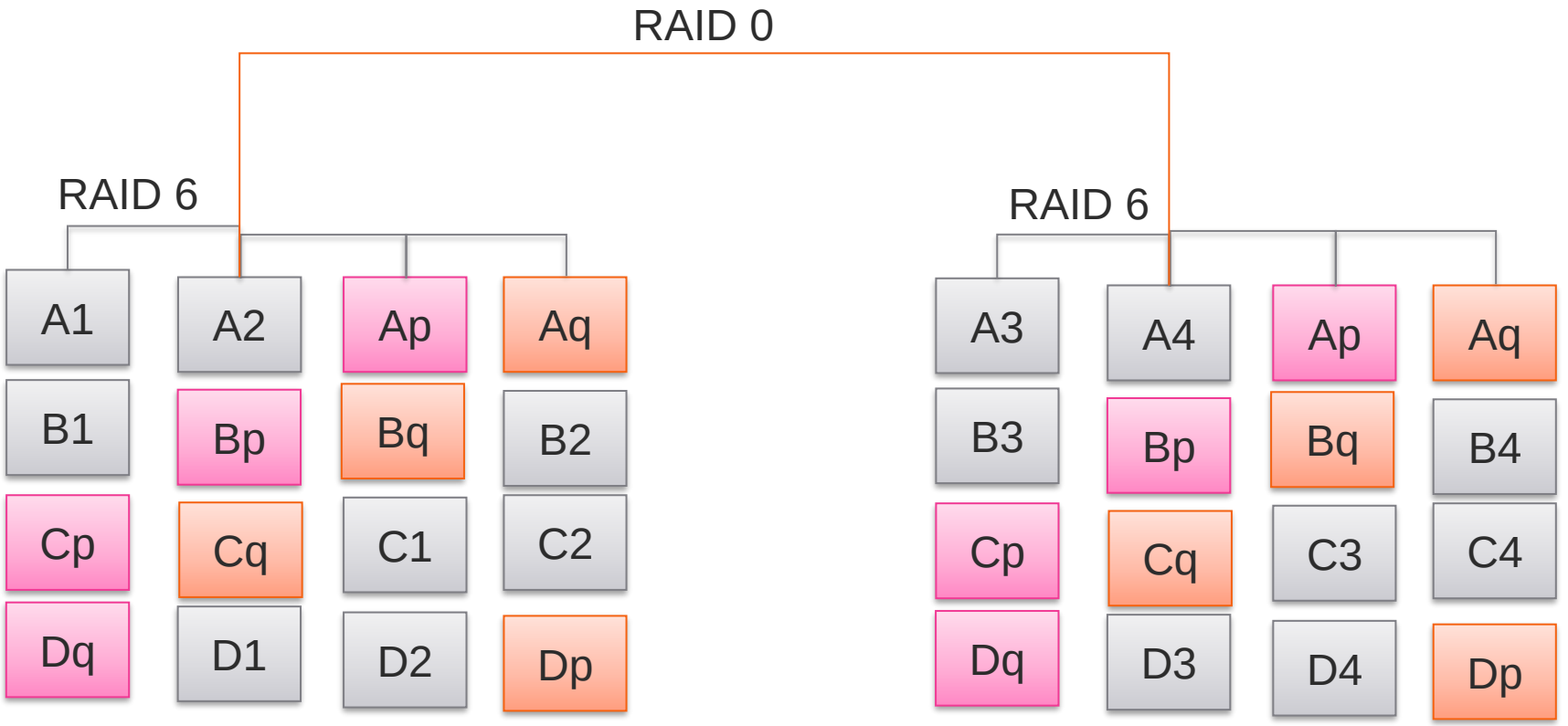
# RAID 10


RAID 0
RAID 1
RAID 1
Block 1
Block 3
Block 5
Block 7
Block 1
Block 3
Block 5
Block 7
Block 2
Block 4
Block 6
Block 8
Block 2
Block 4
Block 6
Block 8
Disk 1
Disk 2
Disk 3
Disk 4

# RAID 50


RAID 0
RAID 5
RAID 5
A1
B1
Cp
D1
A2
Bp
C1
D2
Ap
B2
C2
Dp
A3
B3
Cp
D3
A4
Bp
C3
D4
Ap
B4
C4
Dp
Disk 1
Disk 2
Disk 3
Disk 4
Disk 5
Disk 6

# RAID 60


RAID 0
RAID 6
RAID 6
A1
A2
Ap
Aq
B1
Bp
Bq
B2
Cp
Cq
C1
C2
Dq
D1
D2
Dp
A3
A4
Ap
Aq
B3
Bp
Bq
B4
Cp
Cq
C3
C4
Dq
D3
D4
Dp

IT601 – System and Network Administration

# Server Racks

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Server Racks

- Racks organize IT equipment into standardized assemblies that make efficient use of space and other resources
- At the most basic level, a rack consists of two or four vertical mounting rails and the supporting framework required to keep the rails in place.
- The rails and framework are typically made of steel or aluminum to support hundreds or even thousands of pounds of equipment.
- The width of the rails, the horizontal and vertical spacing of the mounting holes, the size of the equipment cabinets and other measurements are standardized
- **Rack Standardization**
  - Most IT equipment is nominally 19 inches wide (including mounting hardware) and follows a standard set by the Electronics Industry Alliance (EIA) and now maintained by the Electronic Components Industry Association (ECIA).
  - The current 19-inch rack standard is called EIA-310-E, which is essentially equivalent to IEC-60297-3-100 or DIN 41494 in other regions. (There's also a standard for 23-inch wide telecom equipment.

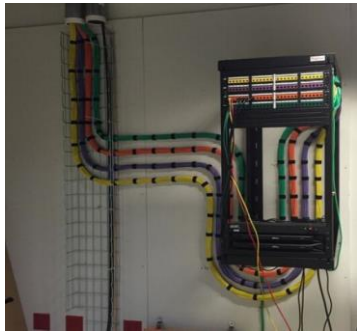
## Rack Units

- Although 19-inch racks are always the same nominal width, the height and depth vary.
- The depth of the rack rails is usually adjustable to some degree.
- The height of the rack is divided into standardized segments called rack units. Each rack unit is 1.75 inches high, and the height of a rack or an equipment cabinet is expressed as the number of rack units followed by the letter “U”. For example, a 42U rack contains 42 rack units.
- That does not mean the rack is exactly 42 x 1.75 inches high because racks usually include at least a little extra space at the top and bottom that isn’t usable rack space.
- It does mean that the rack will accommodate any combination of standard rack equipment up to 42U—whether it’s 42 x 1U switches, 14 x 3U servers or 21 x 1U switches with 7 x 3U servers.
- Remember that the rack also has to be deep enough for the equipment and rated to support the combined weight of all the equipment.

# Server Racks

## Rack Types

- Open frame racks are just that—open frames with mounting rails, but without sides or doors.
- Rack enclosures have removable front and rear doors, removable side panels and four adjustable vertical mounting rails (posts).
- Wall-mount racks are designed to be attached to the wall, saving floor space and fitting in areas where other racks can’t. They can be open frame racks or enclosed cabinets.



# Server Racks

- **Basic rack options**
  - Doors
  - Side Panels
  - Roof
  - Casters and Levelers
  - Locks
  - Hinged Wall Bracket
  - Mounting Holes
  - Color
  - Toolless Mounting

- **Factors influencing Rack Choice**
  - Airflow and cooling
  - Equipment width
  - Security options

# Server Racks

## Rack enclosure and Cooling

- Basic Airflow
- Side Panels
- Airflow Management
- Cable Management
- Thermal Ducts
- Active Heat Removal
- Close-Coupled Cooling

## Other Considerations

- Power Distribution
- Battery Backup
- Device Management
- Patch Panels
- Environmental Monitoring
- Security
- Shock Pallet
- Knockdown
- Stability
- Environmental Protection
- Seismic Protection

IT601 – System and Network Administration

# Other Server Selection Factors

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Buying Servers - additional features for server hardware

- Extensibility
- More CPU performance
- High-performance I/ORack mountable
- Upgrade options
- No side-access needs
- High-availability options
- Maintenance contracts
- Management options

IT601 – System and Network Administration

# Server Architecture/Approaches

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Server Architecture / Approaches

**Primary Approaches**

- **All eggs in one basket:**
One machine used for many purposes

- **Beautiful snowflakes:**
Many machines, each uniquely configured

- **Buy in bulk, allocate fractions:**
Large machines partitioned into many smaller virtual machines using virtualization or containers

**Other Approaches**

- **Grid computing:**
Many machines managed one as unit

- **Blade servers:**
A hardware architecture that places many machines in one chassis

- **Cloud-based compute services:**
Renting use of someone else's servers

- **Software as a service (SaaS):**
Web-hosted applications

- **Server appliances:**
Purpose-built devices, each providing a different service

# IT601 – System and Network Administration

# All in One

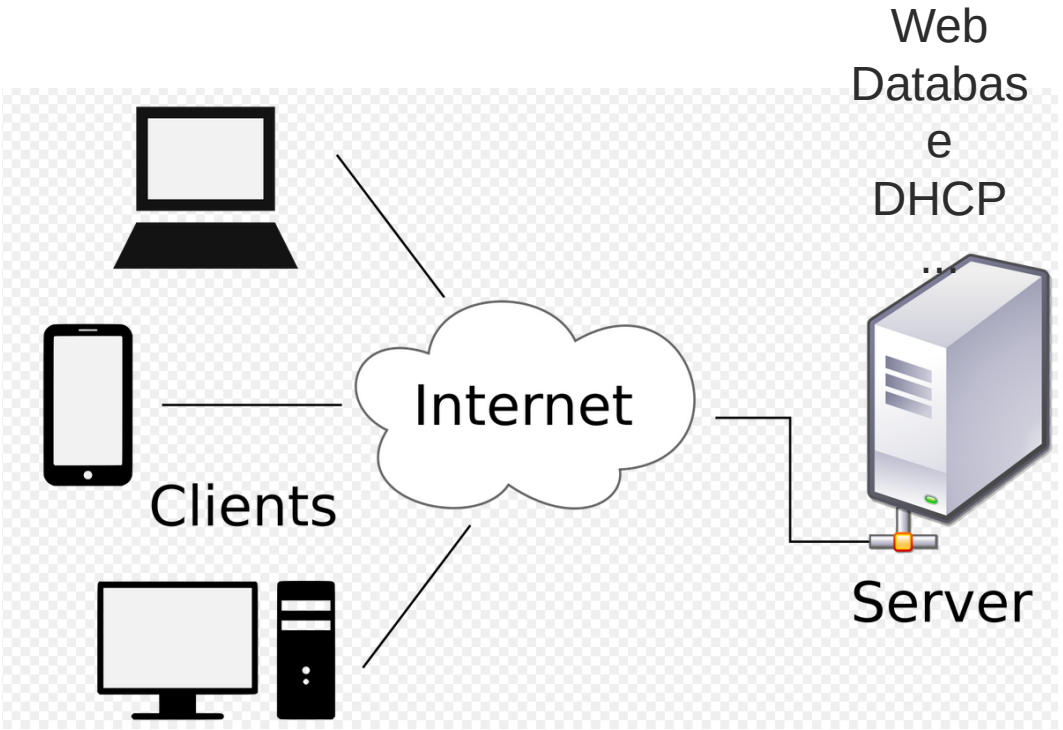
Arif Husen

**Department of Computer Science and Information Technology,**

**Virtual University of Pakistan**

# All in One / All eggs in one basket

- One Main Server
  - All services like Database, Web, DNS, HTTP,Proxy all runs on same
  - High End Hardware
  - Virtual Machines vs. Containers
  - Complete Failure
  - Complex process



IT601 – System and Network Administration

# Snowflake

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Snowflake Architecture

- A better strategy is to use a separate machine for each service.
- purchase servers as they are needed, ordering the exact model and configuration that is right for the application
- Each machine is sized for the desired application:
  - RAM, disk, number and speeds of NICs, and enough extra capacity, or expansion slots, for projected growth during the expected life of the machine.
  - Vendors can compete to provide their best machine that meets these specifications.
- The benefit of this strategy is that the machine is the l meets the requirements.
  - The downside is that the result is a fleet of unique machines.
  - Each is a beautiful, special little snowflake

# Snowflake Architecture

- While snowflakes are beautiful, nobody enjoys a blizzard.  Each new system adds administrative overhead proportionally.

- For example,
  - it would be a considerable burden if each new server required learning an entirely new RAID storage subsystem.
  - Each one would require learning how to configure it, replace disks, upgrade the firmware, and so on.
  - If, instead, the IT organization standardized on a particular RAID product, each new machine would simply benefit from what was learned earlier.

# Snowflake Architecture -> Asset Tracking

- **When managing many unique machines it becomes increasingly important to maintain an inventory of machines**
- **The inventory should document**
  - Technical information such as the operating system
  - Hardware parameters such as amount of RAM, type of CPU, and so on.
  - It should also track the machine owner (either a person or a department),
  - the primary contact (useful when the machine goes haywire),
  - the services being used.
- **Use automated means**
  - Makes it less likely the information will become outdated.
  - If automated collection is not possible, an automated annual review, requiring affirmations from the various contacts, can detect obsolete information that needs to be updated.
- **Inventory applications**

# Snowflake Architecture -> Reducing Variations

- **Always be on the lookout for opportunities to reduce the number of variations in platforms or technologies being supported.**
- **Discourage gratuitous variations by taking advantage of the fact that people lean toward defaults.**

- **Make right easy:**
  - Make sure that the lazy path is the path you want people to take.
  - Select a default hardware vendor, model, and operating system and make it super easy to order.
  - Provide automated OS installation, configuration, and updates.
  - Provide a wiki with information about recommended models and options, sales contact information, and assistance.

# Snowflake Architecture -> Global Optimization

- **While it sounds efficient to customize each machine to the exact needs of the service it provides, the result tends to be an unmanageable mess.**

- **Classic example of a local optimization that results in a global deoptimization.**
  - For example, if all server hardware is from one vendor, adding a single machine from a different vendor requires learning a new firmware patching system, investing in a new set of spare parts, learning a new customer support procedure, and so on.

  - The added work for the IT team may not outweigh the benefits gained from using the new vendor.

IT601 – System and Network Administration

# Buy in bulk, allocate fractions

Arif Husen

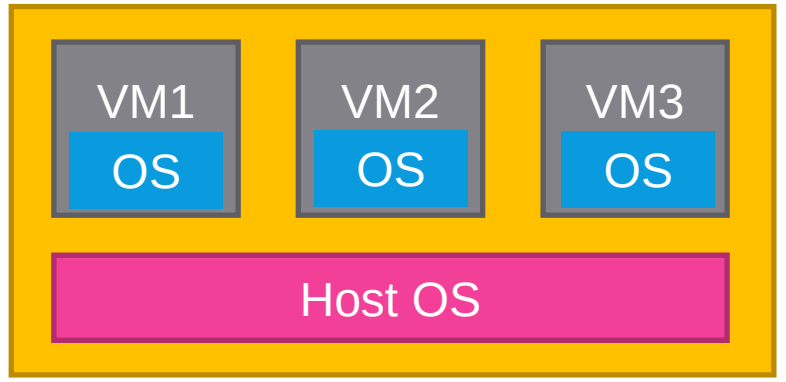
Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Buy in bulk, allocate fractions

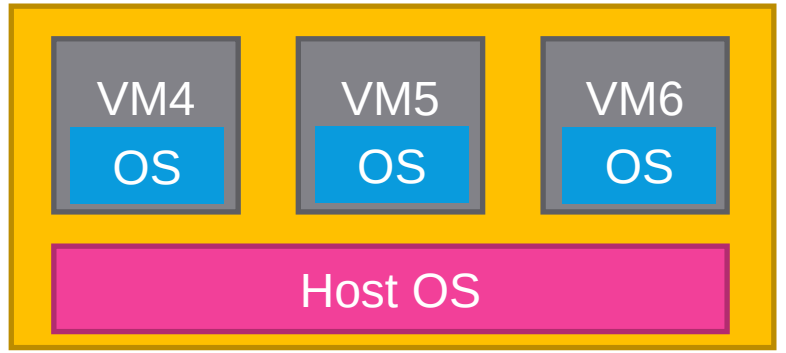
- **The next strategy is to buy computing resources in bulk and allocate fractions of it as needed**

- **One way to do this is through virtualization.**
  - That is, an organization purchases large physical servers and divides them up for use by customers by creating individual virtual machines (VMs).
  - A virtualization cluster can grow by adding more physical hardware as more capacity is needed.



Physical Server 1

- **Buying an individual machine has a large overhead.**
  - It must be specified, ordered, approved, received, rack mounted, and prepared for use.
  - It can take weeks. Virtual machines, by comparison, can be created in minutes.
  - A virtualization cluster can be controlled via a portal or API calls, so automating processes is easy.



Physical Server 2

# Buy in bulk, allocate fractions

- **VMs can also be resized.**
  - You can add RAM, vCPUs, and disk space to a VM via an API call instead of a visit to the datacenter.
  - If customers request more memory and you add it using a management app on your iPhone while sitting on the beach, they will think you are doing some kind of magic.
- **Virtualization improves computing efficiency.**
  - Physical machines today are so powerful that applications often do not need the full resources of a single machine.
  - The excess capacity is called stranded capacity because it is unusable in its current form.
  - Sharing a large physical machine's power among many smaller virtual machines helps reduce stranded capacity, without getting into the "all eggs in one basket" trap.

# Buy in bulk, allocate fractions – Benefits of Isolation

- Stranded capacity could also be mitigated by running multiple services on the same machine.
- However, virtualization provides better isolation than simple multitasking. The benefits of isolation include

  - **Independence**
    - Each VM can run a different operating system.
    - On a single physical host there could be a mix of VMs running a variety of Microsoft Windows releases, Linux releases, and so on.

  - **Resource isolation**
    - The disk and RAM allocated to a VM are committed to that VM and not shared.
    - Processes running on one VM can’t access the resources of another VM.
    - In fact, programs running on a VM have little or no awareness that they are running on VMs, sharing a larger physical machine.

# Buy in bulk, allocate fractions – Stranded Capacity Mitigation

- **Granular security**
  - A person with root access on one VM does not automatically have privileged access on another VM. Suppose you had five services, each run by a different team.
  - If each service was on its own VM, each team could have administrator or root access for its VM without affecting the security of the other VMs.
  - If all five services were running on one machine, anyone needing root or administrator access would have privileged access for all five services.
- **Reduced dependency hell**
  - Each machine has its own operating system and system libraries, so they can be upgraded independently.

# Buy in bulk, allocate fractions – VM Management

- Like other strategies, keeping a good inventory of VMs is important.
- The cluster management software will keep an inventory of which VMs exist, but you need to maintain an inventory of who owns each VM and its purpose.
- Some clusters are tightly controlled, only permitting the IT team to create VMs with the care and planning reminiscent of the laborious process previously used for physical servers.
- Other clusters are general-purpose compute farms providing the ability for customers to request new machines on demand.
  - provide a self-service way for customers to create new machines
  - The process can be fully automated via the API to avoid delays VM creation
  - In addition to creating VMs, users should be able to reboot and delete their own VMs

# Buy in bulk, allocate fractions – VM Management

- There should be limits in place so that customers can't overload the system by creating too many VMs.
- Typically limits are based on existing resources, daily limits, or per-department allocations.
- Users can become confused if you permit them to select any amount of disk space and RAM.
- They often do not know what is required or reasonable.
  - One strategy is to simply offer reasonable defaults for each OS type.
  - Another strategy is to offer a few options: small, medium, large, and custom.
  - As in the other strategies, it is important to limit the amount of variation.

# Buy in bulk, allocate fractions – Live Migration

Most virtual machine cluster management systems permit live migration of VMs, which means a VM can be moved from one physical host to another while it is running.
Aside from a brief performance reduction during the transition, the users of the VM do not even know they're being moved.

**Benefits of Live Migration**

- Live migration makes management easier.
- It can be used to rebalance a cluster, moving VMs off overloaded physical machines to others that are less loaded.
- It also lets you work around hardware problems.
- If a physical machine is having a hardware problem, its VMs can be evacuated to another physical machine.
- The owners of the VM can be blissfully unaware of the problem. They simply benefit from the excellent uptime.

➢ **Shared Storage**

- The architecture of a typical virtualization cluster includes many physical machines that share a SAN for storage of the VM's disks.
- By having the storage external to any particular machine, the VMs can be easily migrated between physical machines.

# Buy in bulk, allocate fractions – Live Migration

Shared storage is depicted in Figure on next slide. The VMs run on the VM servers and the disk volumes they use are stored on the SAN. The VM servers have little disk space of their own.

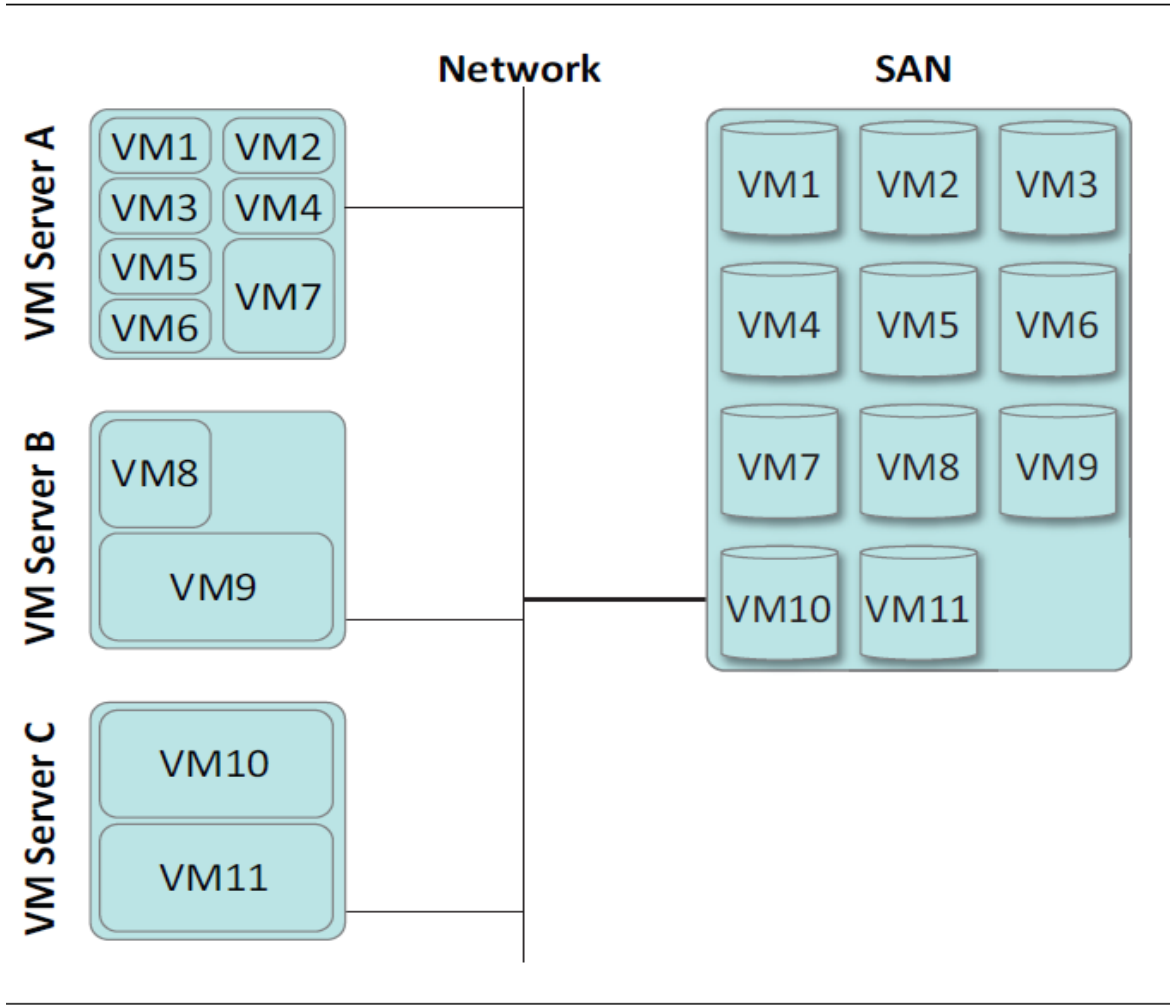



Figure 13.1: VM servers with shared storage

# Buy in bulk, allocate fractions – Live Migration

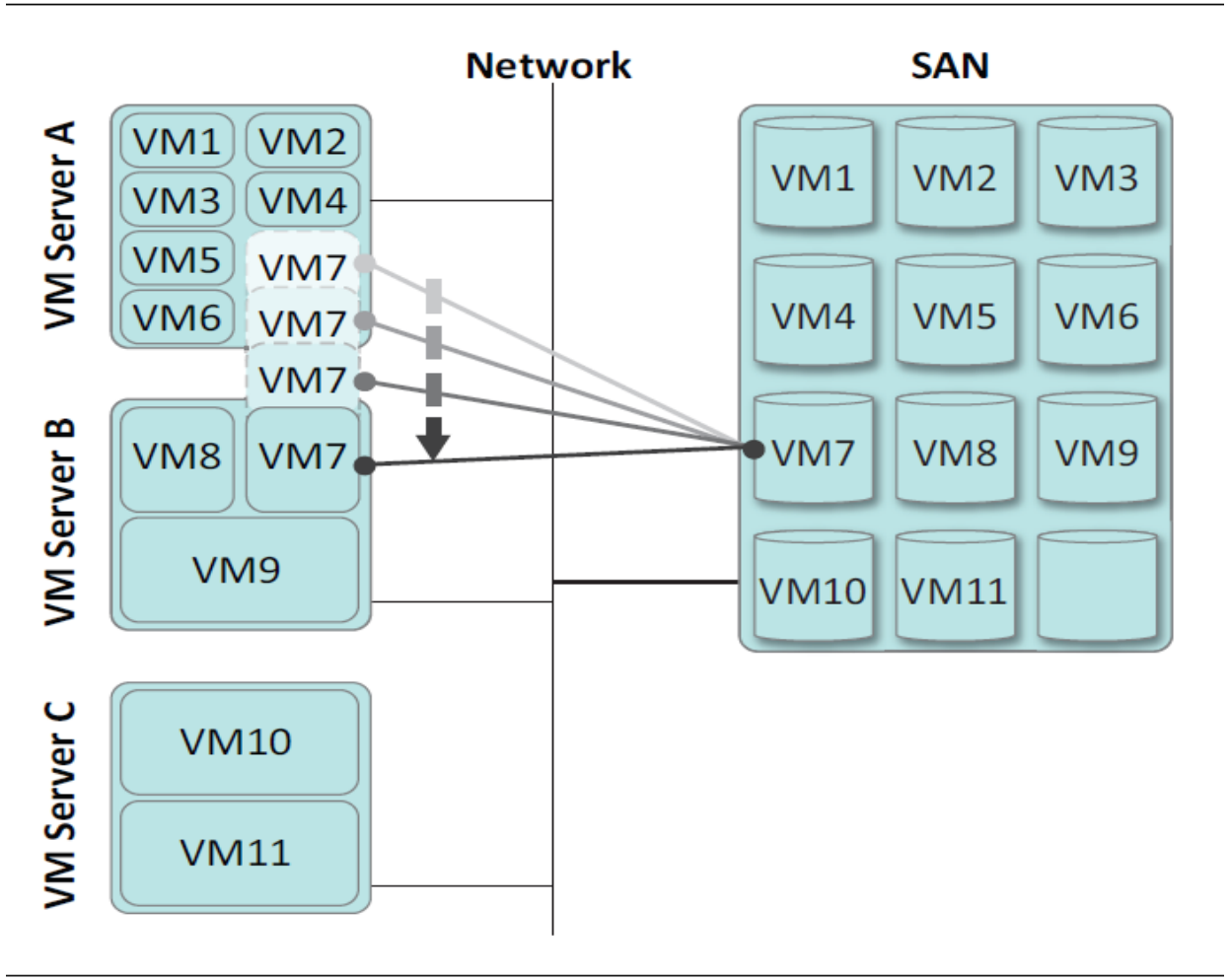



Figure 13.2: Migrating VM7 between servers

IT601 – System and Network Administration

# Buy in bulk, allocate fractions Other Aspects

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Buy in bulk, allocate fractions – VM Packing

- VMs can reduce the amount of stranded compute capacity, they do not eliminate it.
- VMs cannot span physical machines.
  - As a consequence, we often get into situations where the remaining RAM on a physical machine is not enough for a new VM.
- The best way to avoid this is to create VMs that are standard sizes that pack nicely
- Example
  - Define the small configuration
  - Define the medium configuration
  - Large configuration to fit two VMs

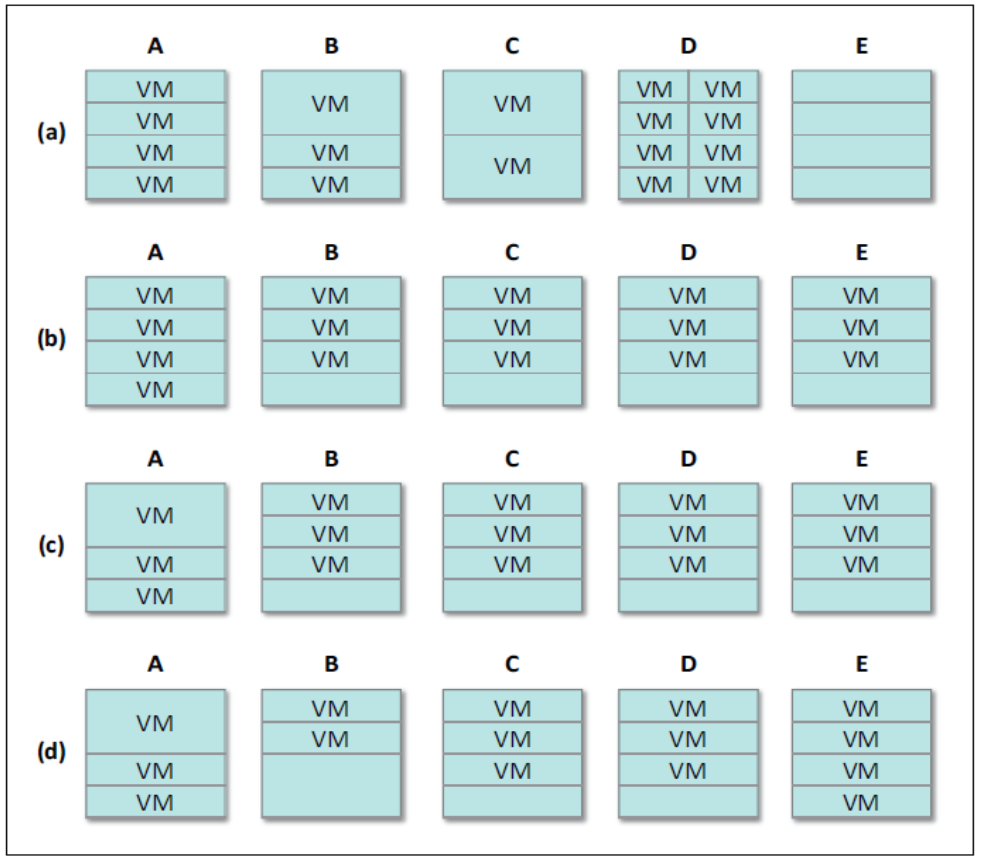



Figure 13.3: VM packing

# Buy in bulk, allocate fractions – Spare Capacity for Maintenance

- If a physical machine needs to be taken down for repairs, there has to be a place where the VMs can be migrated if you are to avoid downtime
- Always reserve capacity equivalent to one or more physical servers.
  - N+1 redundancy : Reserving capacity equivalent to one physical server
  - N+2 redundancy : Reserving capacity equivalent to two physical server
  - N+x redundancy : higher redundancy if there is a likelihood that multiple physical machines will need maintenance at the same time.
- One strategy is to keep one physical machine entirely idle
  - spare machine is entirely unused
- Another strategy is to distribute the spare capacity around the cluster

# Buy in bulk, allocate fractions – Unified VM/Non-VM Management

- Most sites end up with two entirely different ways to request, allocate, and track VMs and non-VMs.
- It can be beneficial to have one system that manages both.
- Some cluster management systems will manage a pool of bare-metal machines using the same API as VMs.
  - Creating a machine simply allocates an unused machine fromthe pool.
  - Deleting a machine marks the machine for reuse.
- Another way to achieve this is to make everything a VM, even if that means offering an an extra large size, which is a VM that fills the entire physical machine.
- While such machines will have a slight performance reduction due to the VM overhead, unifying all machine management within one process benefits customers, who now have to learn only one system, and makes management easier.

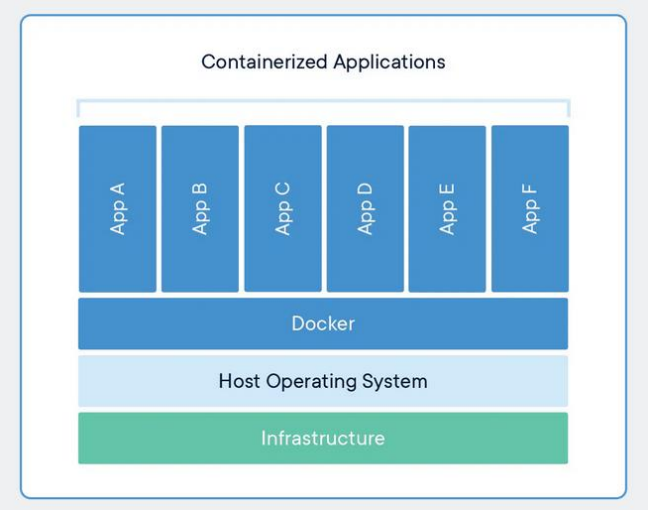
# Buy in bulk, allocate fractions – Containers

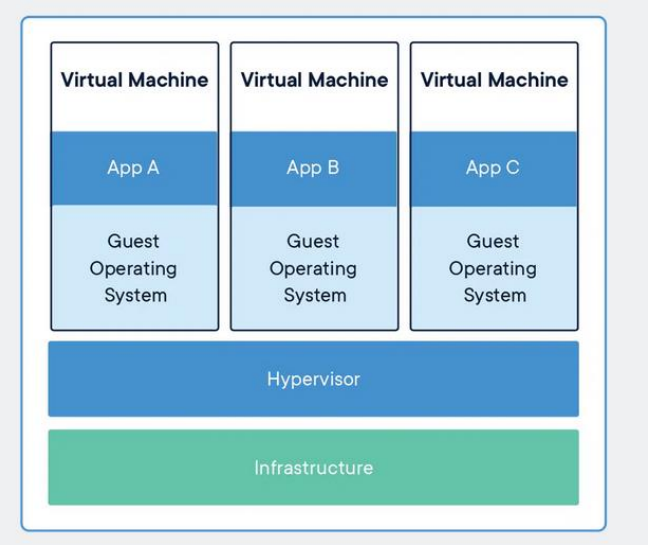
- Containers are another virtualization technique.
  - process level instead of the machine level.
  - VM is a machine that shares physical hardware with other VMs, each container is a group of processes that run in isolation on the same machine.

- All of the containers run under the same operating system, but each container is self-contained as far as the files it uses.
  - Therefore there is no dependency hell.

- Containers are much lighter weight and permit more services to be packed on fewer machines.

- Docker,
- Mesos
- Kubernetes



IT601 – System and Network Administration

# Grid Computing Approach

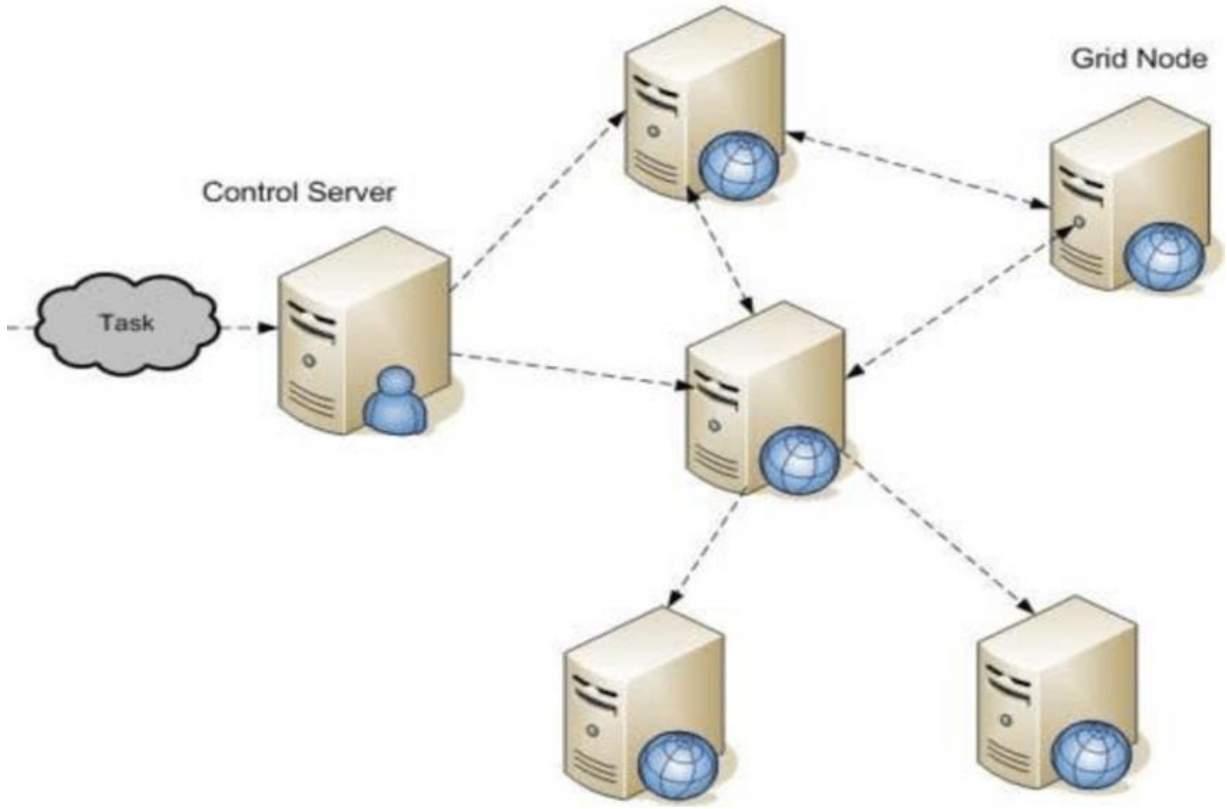
Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Grid computing

- Grid computing takes many similar machines and manages them as a single unit.
- To use the grid, a customer specifies how many machines are needed and which software package to run
  - The grid management system allocates the right number of machines, installs the software on them, and runs the software.
  - When the computation is done, the results are uploaded to a repository and the software is de-installed.




- scheduling algorithm
- grids are very controlled systems. All allocations are done though the grid management and scheduling system

# Grid computing – Key Components and Benefits

- Key Components

- Grid computing is more efficient than virtualization because it eliminates the virtualization overhead, which is typically a 5 to 10 percent reduction in performance.

- Grids are easier to manage because what is done for one machine is done for all machines. They are fungible units i.e each one can substitute for the others. If one machine dies, the scheduler can replace it with another machine in the grid

IT601 – System and Network Administration

# Blade Server Approach

Arif Husen

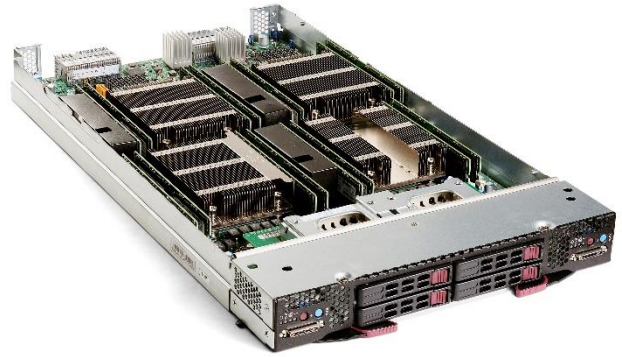
Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Blade servers

- There is a lot of overhead in installing individual machines.
  - Each machine needs to be racked, connected to power, networked, and so on. Blade servers reduce this overhead by providing many servers in one chassis.

- A blade server has many individual slots that take motherboards, called blades, that contain either a computer or storage.
  - Each blade can be installed quickly and easily because you simply slide a card into a slot.
  - There are no cables to connect; the blade's connector conveys power, networking, and I/O connectivity.
  - Additional capacity is added by installing more blades, or replacing older blades with newer, higher-capacity models.



- blade systems is that they are software configurable

- Blade systems are most cost-effective

IT601 – System and Network Administration

# Cloud Computing Approach

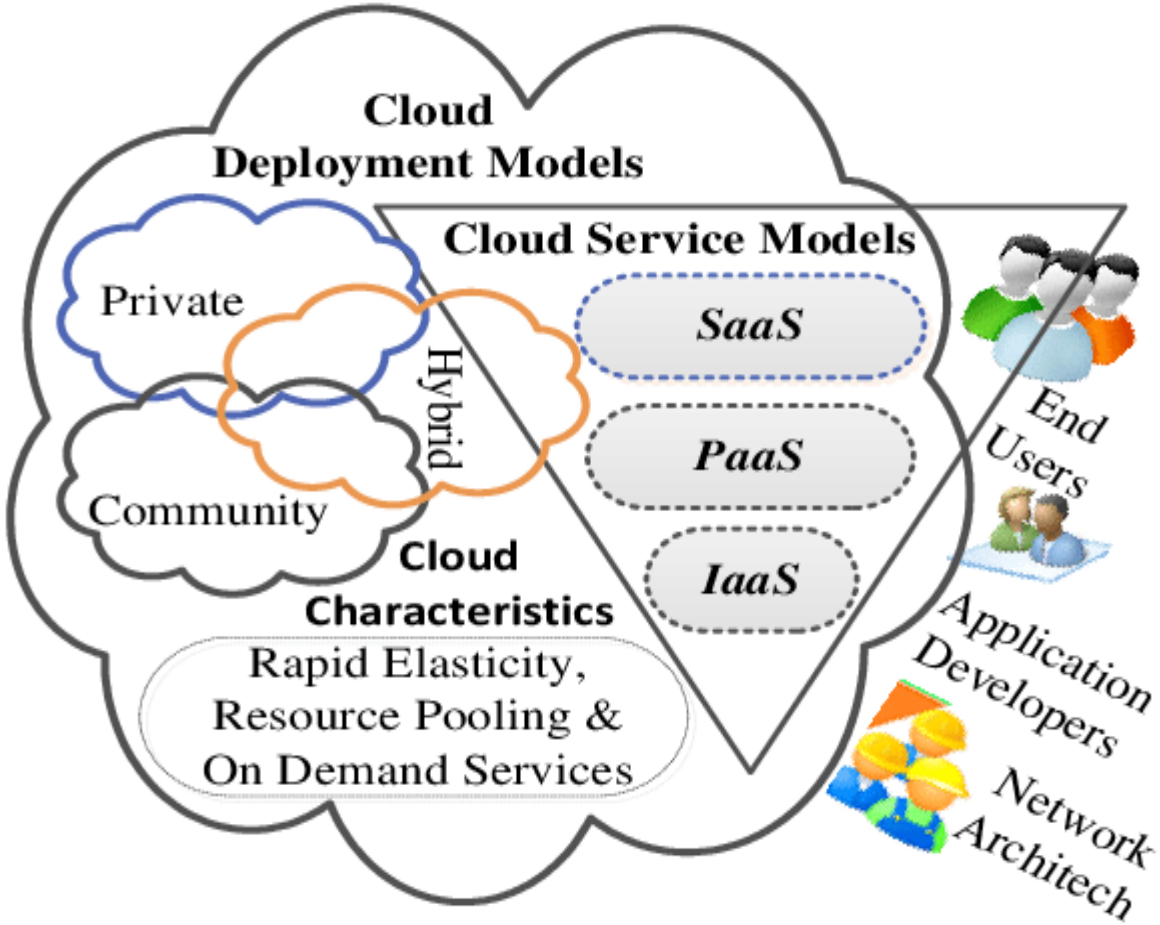
Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Cloud-based compute services

- Another strategy is to not own any machines at all, but rather to rent capacity on someone else's system.
  - Such cloud-based computing lets you benefit from the economies of scale that large warehouse-size datacenters can provide, without the expense or expertise to run them.
  - Examples of cloud-based compute services include Amazon AWS, Microsoft Azure, and Google Compute Engine



# Cloud Definitions - Consumers

- When a typical consumer uses the term the cloud, they meancputting their data on a web-based platform.
  - The primary benefit is that this data becomes accessible from anywhere.
  - For example, consumers might havecall their music stored in the cloud; as a result their music can be played on any
  - device that has Internet access.

# Cloud Definitions - Business

- Typically business people think of the cloud as some kind of rented computing infrastructure that is elastic.
  - That is, they can allocate one or thousands of machines; use them for a day, week, or year; and give them back when they are done.
  - They like the fact that this infrastructure is a payas- you-go and on-demand system.
  - The on-demand nature is the most exciting because they won't have to deal with IT departments that could take months to deliver a single new machine, or simply reject their request. Now with a credit card, they have a partner that always says yes.

# Cloud Definitions – IT Professionals

- When all the hype is removed (and there is a lot of hype), cloud computing comes down to someone else maintaining hardware and networks so that customers can focus on higher-level abstractions such as the operating system and applications.
  - It requires software that is built differently and new operational methods.
  - IT professionals shift from being the experts in how to install and set up computers to being the experts who understand the full stack and become valued for their architectural expertise, especially regarding how the underlying infrastructure affects performance and reliability of the application, and how to improve both.

# Cloud Computing's Cost Benefits

- Cloud-based compute services take that strategy to a larger scale than most companies can achieve on their own, which enables these smaller companies take advantage of these economics.

- Adoption of cloud computing is also driven by another cost: opportunity cost. Opportunity cost is the revenue lost due to missed opportunities. If a company sees an opportunity but the competition beats them to it, that could be millions of potential dollars lost.

IT601 – System and Network Administration

# SAS Approach

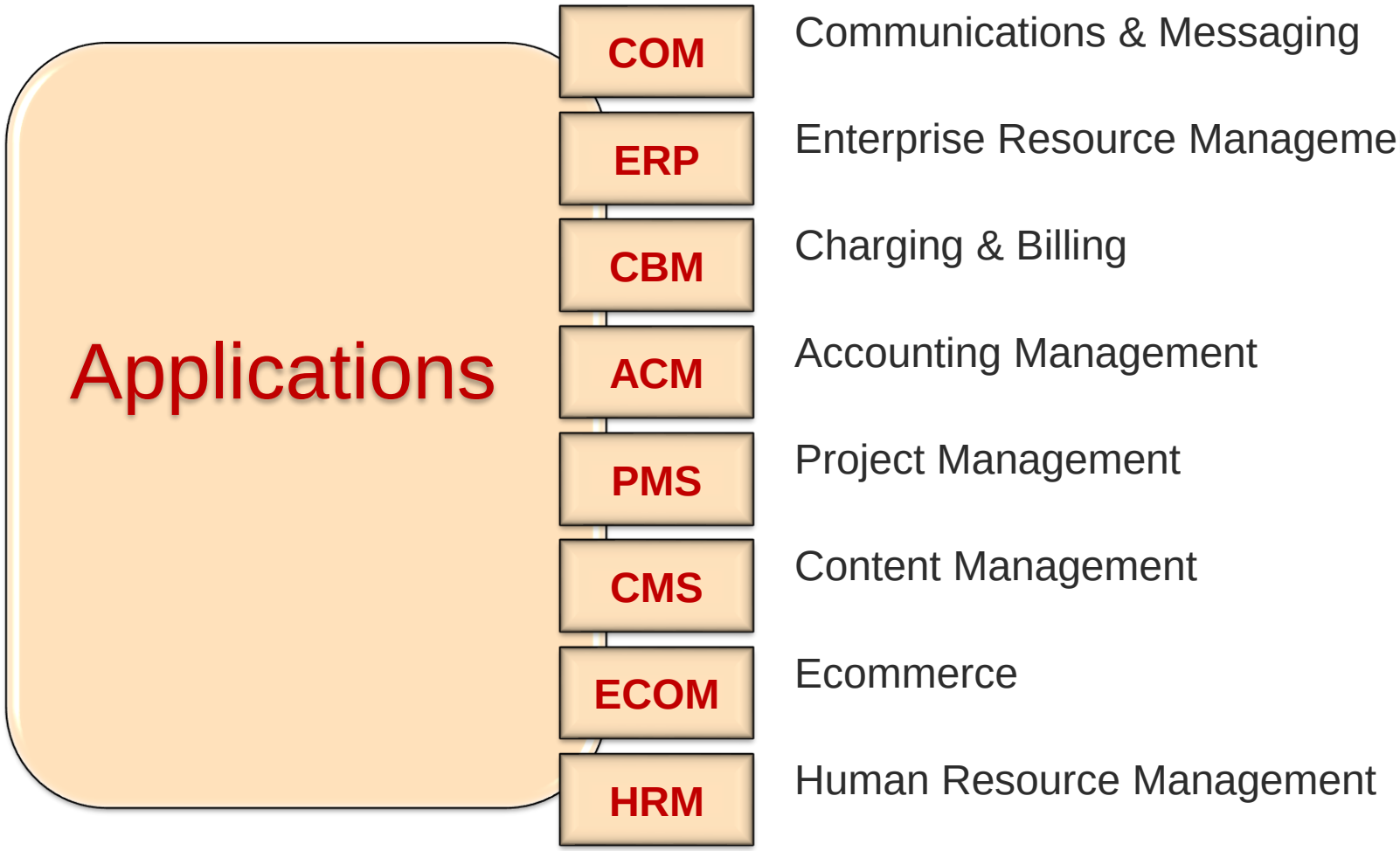
Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Software as a service (SaaS)

- Software as a service (SaaS) means using web-based applications. Many small companies have no servers at all. They are able to meet all their application needs with web-based offerings.



# Software as a service (SaaS)

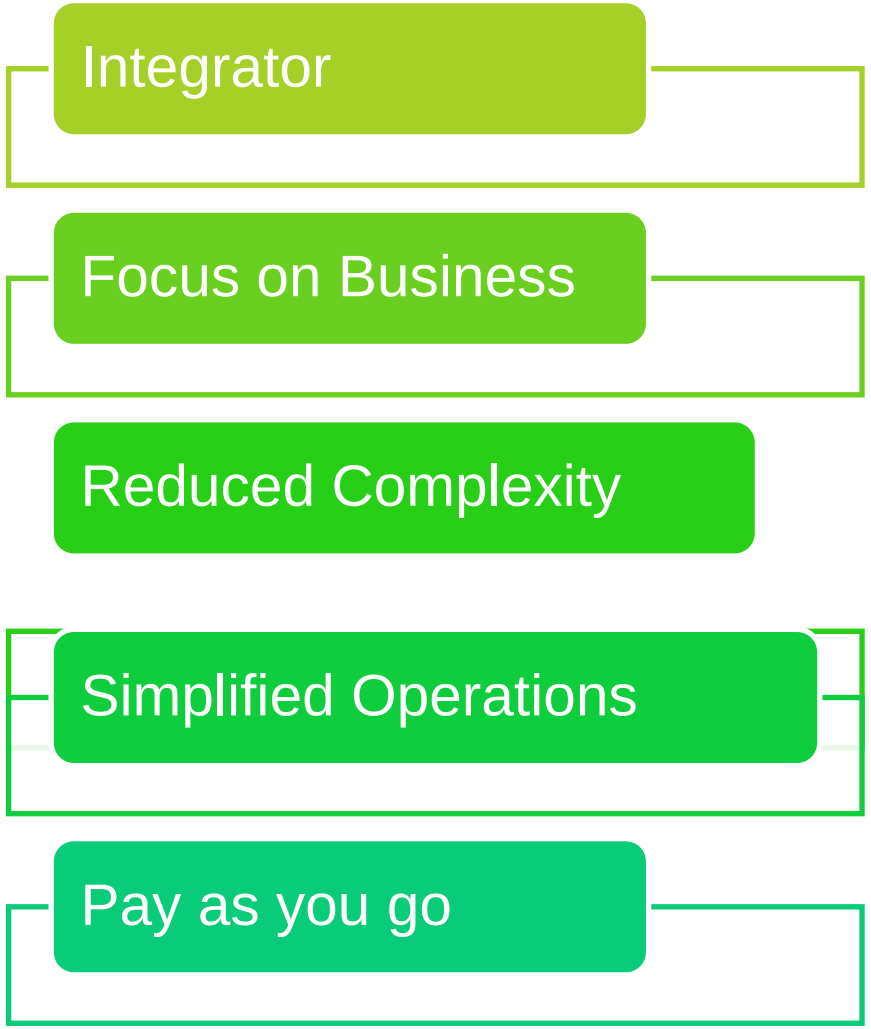
- Organizations host their web site using a hosting service, they use a web-based payroll provider, they share files using Dropbox, they use Salesforce for customer relationship management and sales process management, and they use Google Apps for Work for word processing, spreadsheets, and presentations.

- If they use Chromebooks, iPads, or other fixed-configuration webbrowsing devices, they don’t need traditional computers and can eliminate a lot of the traditional enterprise IT infrastructure that most companies require.

- This strategy was impossible until the early 2010s. Since then, ubiquitous fast Internet connections, HTML5’s ability to create interactive applications, and better security features have made this possible.

# Software as a service (SaaS): Impact

- When a company adopts such a strategy, the role of the IT department becomes that of an IT coordinator and integrator.
- Rather than running clients and services, someone is needed to coordinate vendor relationships, introduce new products into the company, provide training, and be the first stop for support before the provider is contacted directly.
- Technical work becomes focused on high-level roles such as software development for integrating the tools, plus low-level roles such as device support and repair management.



# IT601 – System and Network Administration

<End>

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

IT601 – System and Network Administration

# Server Appliance Approach

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Server appliances:

- An **appliance** is a device designed specifically for a particular task.

- **Examples**
  - Toasters make toast.
  - Blenders blend.

- **General Purpose Vs. Specialized**

  - We could build servers using general-purpose hardware with specific software packages

  - However, there are significant benefits in using a device designed to do one task very well.

# Server appliances

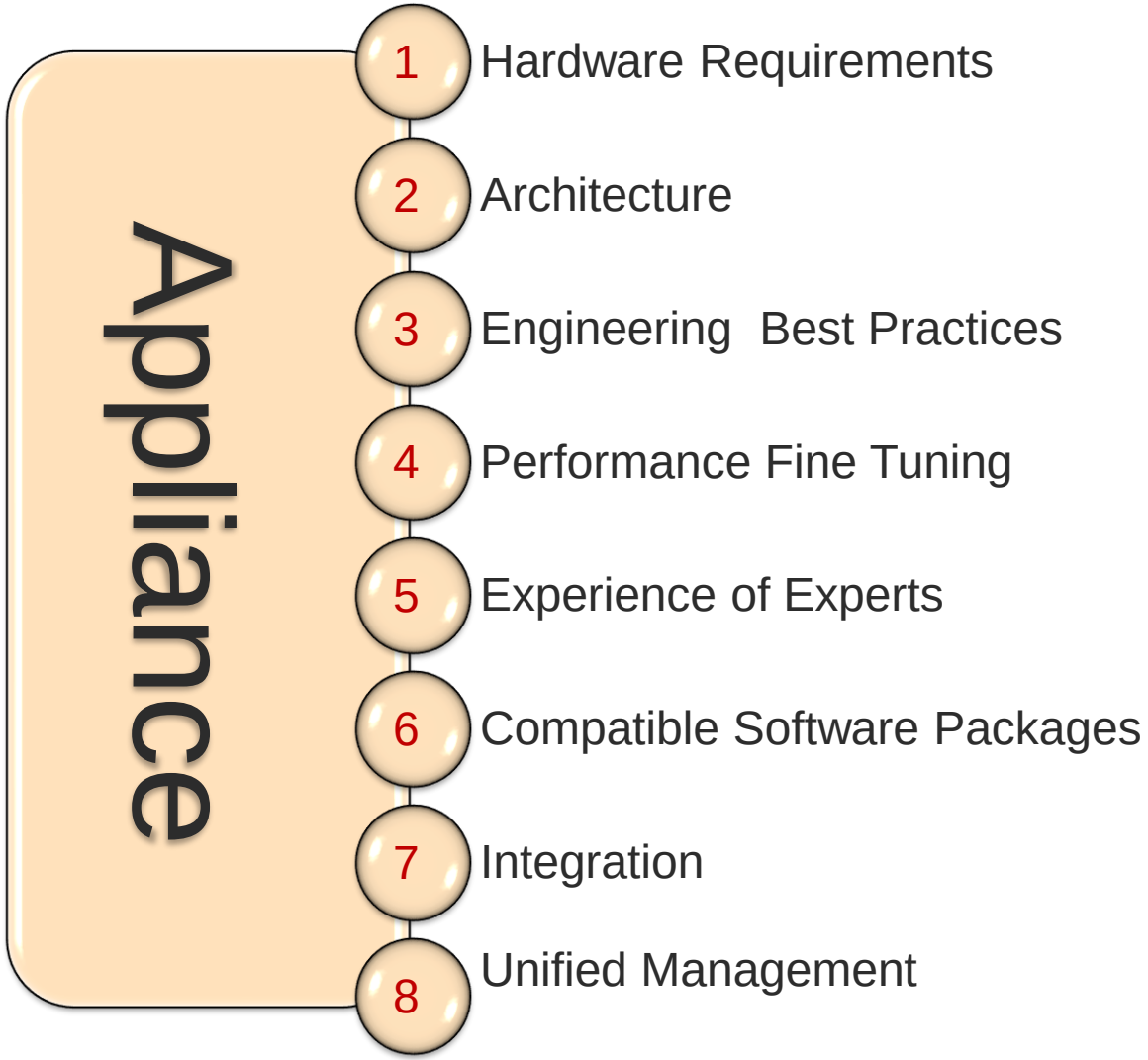
- **The computing area also has appliances:**
  - File server appliances,
  - Web server appliances,
  - Email appliances,
  - DNS/DHCP appliances,
  - and so on.

- **The first appliance was the dedicated network router.**
  - Is it feasible to spend all that money on a device that just sits there and pushes packets when we can easily add extra interfaces to our VAX and do the same thing?”

- **Common Opinion**
  - Some people have point of view that a box dedicated to doing a single task, and doing it well, was in many cases more valuable than a general-purpose computer that could do many tasks.
  - It allows you could reboot the VAX without taking down the network for everyone else.

# Server appliance consolidates

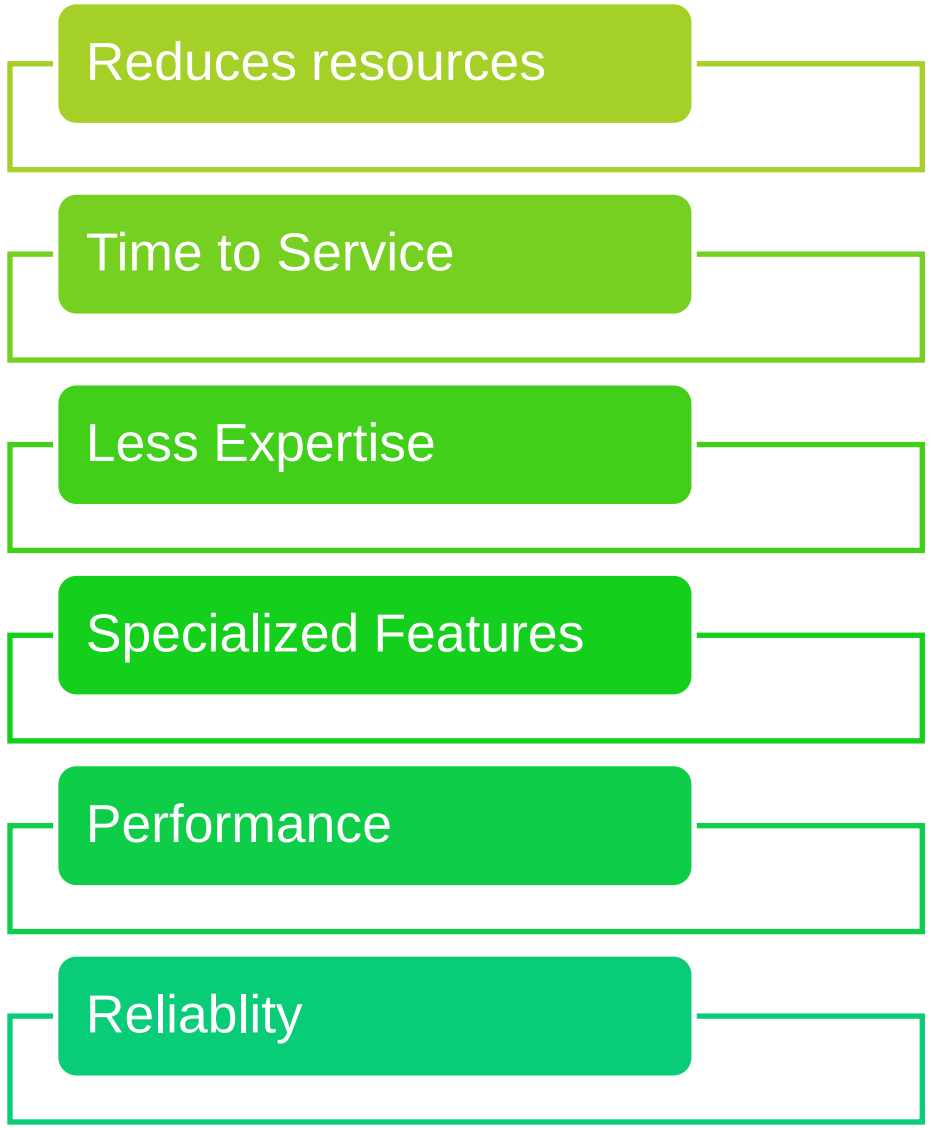
- **A server appliance brings lot of things together in one box.**
  - Architecting a server is difficult.
  - The physical hardware for a server has all the requirements
  - The system engineering and performance tuning that only a highly experienced expert can do.
  - The software required to provide a service often involves assembling various packages, gluing them together, and providing a single, unified administration system for it all.




- **It's a lot of work! Appliances do all this for you right out of the box.**

# Server appliances - Benefits

- Although a senior SA can engineer a system dedicated to file service or email out of a general-purpose server, purchasing an appliance can free the SA to focus on other tasks.
- Every appliance purchased results in one less system to engineer from scratch, plus access to vendor support in case of an outage.
- Appliances also let organizations without that particular expertise gain access to well-designed systems.
- The other benefit of appliances is that they often have features that can't be found elsewhere.
- Competition drives the vendors to add new features, increase performance, and improve reliability.
- For example, NetApp Filers have tunable file system snapshots that allow end users to "cd back in time," thus eliminating many requests for file restores.



# IT601 – System and Network Administration

<End>

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

IT601 – System and Network Administration

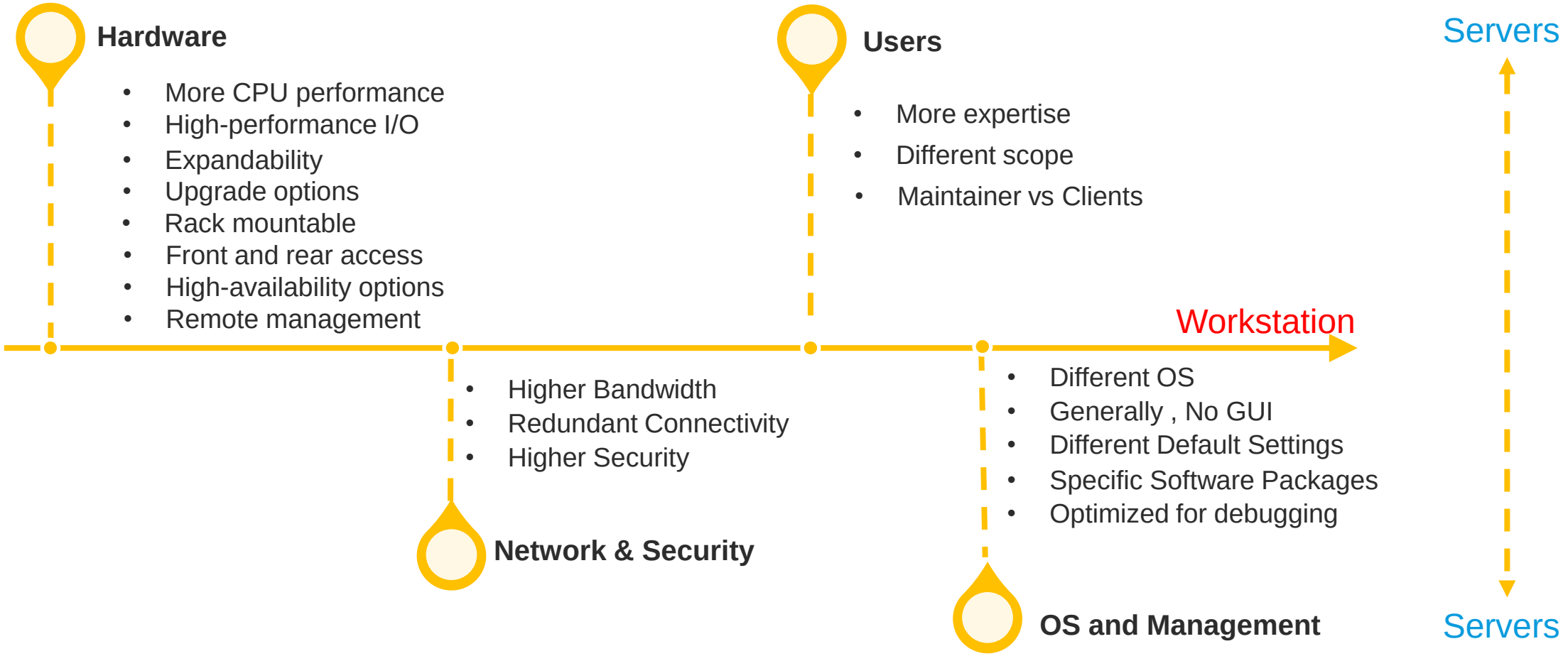
# Server Selection Aspects

Arif Husen

Department of Computer Science and Information Technology,

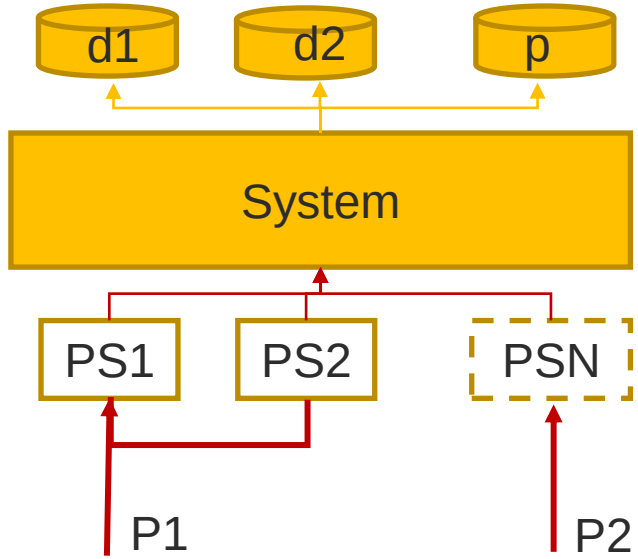
Virtual University of Pakistan

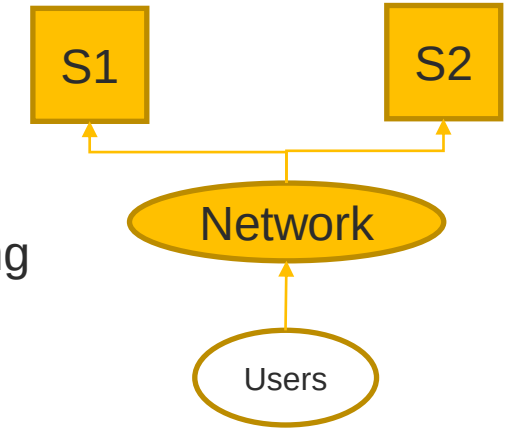
# The Differences



# Server Reliability

- **Levels of Redundancy**
  - N+0, N+1,…, N+r
  - Identify Redundant Units, Power Supplies, CPUs, Hard disks etc
- **Data Integrity**
  - ❑ Ensure data is available and valid in future
  - ❑ RAID , Service Failure Vs Component failure
  - ❑ Non-RAID Approaches,
    - Redundant Systems vs Redundant Components
    - Distributed systems e.g Google's GFS, Hadoop's HDFS, Cassandra
    - Backups
- **Hot-Swap Components**
  - the ability to add, remove, and replace a component while the system is running
  - Not All components are hot swapable
- **Servers Should Be in Computer Rooms**



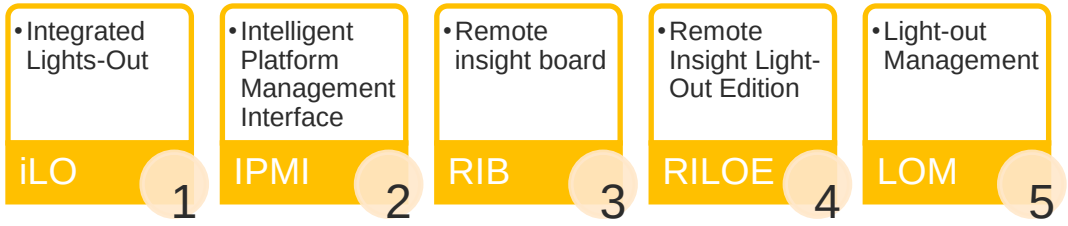



# Remotely Managing Servers

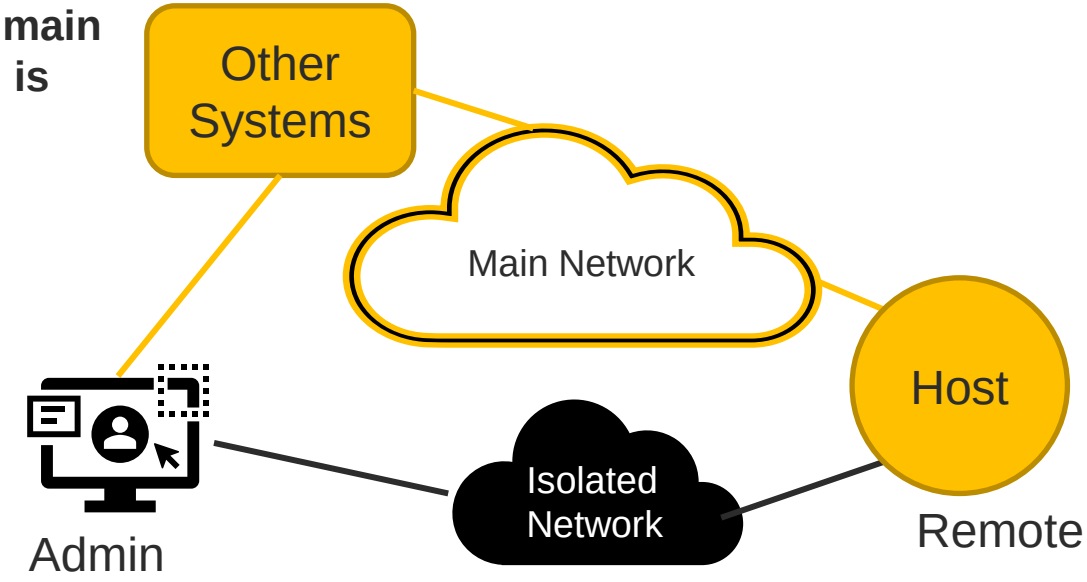
- Do all system administration tasks involving the machine from a remote location, except physical labor such as adding and removing physical hardware.
    - Ability to remotely access the machine's console and, optionally, have remote control of the power switch.
    - Ability to operate a system's console when the machine is in a bad state or otherwise disconnected from the network. e.g. Accessing a basic BIOS configuration, CTRL-ALT-DEL keys
- Saves time and cost
- Need to consider security aspects
- **Two Types of Remote management techniques**
    - Integrated Out-of-Band Management
    - Non-integrated Out-of-Band Management

# Integrated Out-of-Band Management

- **Integrated Out-of-Band Management**
  - Modern servers have remote management capabilities built-in. Such systems are generically called out-of-band (OOB) management and have an Ethernet port as an interface.
- **OOB interface requires an IP Address, and remote system console is accessed via browser or client.**
  - Remote access via a browser has limited functionalities as compared to clients.




- **The remote management systems must be isolated from the main network and must be accessible even when the main system is down or powered off**
- **Security Aspects**
  - OOB interface is equivalent to physical access to the machine
  - Built-in systems have been found to suffer from traditional security holes, such as buffer overruns
  - Don't assume that a password on the OOB interface and using SSL is sufficient protection
  - Put OOB interfaces on a dedicated protected network, use a web proxy, authentication and authorization



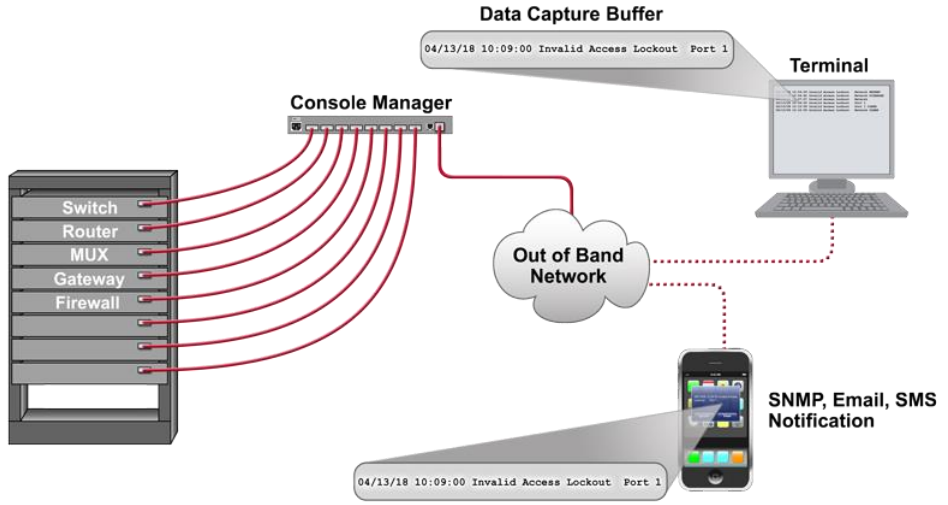
# Non-integrated Out-of-Band Management

- The built in OOB has some limitations in terms of remote power cycling and console access. Third party tools can address these issues.

  - **Remote Power Cycle**
    - Switched power distribution unit (PDU) provide the remote power cycling. A PDU is the fancy name for a power strip

  - **Remote Console with IP-KVM**
    - Keyboard, video screen, and mouse (KVM) switch is a device that allows many machines share a single.
    - An IPKVM switch provides the remote console access.
    - An IP-KVM is a KVM switch that can be accessed remotely. This eliminates the need for any monitors, keyboards, or mice in the computer room. You simply run a client on your workstation that connects to the IP-KVM.

  - **Remote Console with Serial Consoles**
    - Network equipment, appliances, and older servers have serial consoles. They have no video, keyboard, or mouse connector.
    - In the old days one would attach a physical VT-100 terminal to the serial console port of each device
    - In the 1990s it became popular to replace physical terminals with a terminal console concentrator.
    - Also, serial console concentrators usually have the option to require authentication—for example, using RADIUS or TACACS—before allowing someone to connect to a console.



# Separate Administrative Networks

- **It is common for servers to have a separate NIC that is connected to a dedicated administrative network. Separating administrative traffic from normal service traffic has a number of benefits.**
  - For servers, the primary benefit is often to isolate disruptive traffic.
  - Backups, for example, consume a large amount of bandwidth and can interfere with normal service traffic.
    - Sometimes servers have a dedicated backup interface that uses a high-speed network dedicated to backups.
- **Characteristics of administrative networks**

| Stable & Static | Strict Security | Dedicated | Separation | Simplicity | Availability |
|---|---|---|---|---|---|
| • Should be more stable<br>• More static<br>• Service Networks are more dynamic | • More Restrictive firewall rules<br>• Serves more critical systems | • Only for administrative purpose | • Should be separated from other service network | • Simple Engineering<br>• No VLANs<br>• Simple Topology | • Should be always available<br>• Immune to outages |

# Maintenance Contracts and Spare parts

- **Vendor SLA**
  - There is a variety of maintenance contract options, with different service level agreements (SLAs).
    - Replacement for a bad part with a 4-hour response time, a 12-hour response time, or next-day options.
      - Sometimes the options include 24/7 response, but other times just specify some number of business day
    - Other options include having the customer purchase a kit of spare parts and receive replacements after a spare part gets used.
      - Usually, the installation of the replacement part is by organization itself, though vendors offer onsite repairs for a higher price. Sometimes the service offerings vary by country or region, depending on where the vendor has local presence.

- **Response time**
  - Response Start
  - Fast Response Vs Cost
  - Plan for redundancy

- **Critical Servers**
  - Lowest level maintenance contract
  - Redundant Server
  - Maintenance contract with warranties
    - Vendors to maintain spares in same city

- **Non-Critical Servers**
  - Default maintenance contract
  - Flexible Response time
  - Low cost maintenance contract

# Maintenance contracts and spare parts

- **Spare Parts**
  - In house  repair department vs maintenance contract
    - Cost
    - Trainings, Certifications on the hardware
    - Maintaining inventory of spares
  - Self-Support Plans
    - Repair kit with maintenance contract
      - Homogeneous Infrastructure to allow sharable spares
      - No Licenses
      - Cold Spares
      - Suitable for High End Servers

# Maintenance contracts and spare parts

➢ **Tracking Service Contracts**

- **Machines not covered in maintenance contract**
  - Discovered during outage
  - Generally , coordinating with sales staff of vendor will add  to maintenance contract

- **Improvement Strategies**

  ❑ Service contracts with 10%  additional price
    - allows adding new machines as added
    - vendors can easily extend maintenance contract

  ❑ Ensure each machine is purchased with maintenance contract
    - Avoid situations where vendor can agree to  add new machines into existing maintenance contracts

  ❑ Tract expiry of the initial maintenance contract that came with purchase
    ❑ Remove dismantled machines
    ❑ Add new machines

# Maintenance contracts and spare parts

- **Cross-Shipping**
  - A type of shipping in which vendors requires the faulty units to be received before shipping replacement unit.
    - Alternate approach  may be  that vendor ship the replacement immediately and faulty unit is sent back afterwards.
    - Vendors use it to avoid  unnecessary replacements
  - Cross shipping should be part of maintenance contract.

# Selecting vendors with server experience

- Some vendors have years of experience designing servers, and it shows
- They build hardware with the common server-related features, as well as include little extras that one can learn only from years of market experience.
- Servers from inexperienced vendors may lead to .
  - Issues to integrate with enterprise systems such as authentical systems
  - Management of the device is manual and cumbersome, lacking the ability to manage many servers at once, and with no kind of remote management
  - No maintenance contracts or degraded maintenance contracts
  - Low quality technical support
  - Issues in replacement of parts
- Select vendors that are known for building reliable hardware. Some vendors cut corners by using consumer-grade parts; others use premium-quality parts.

# IT601 – System and Network Administration

<End>

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# IT601 – System and Network Administration

# Services

Arif Husen

**Department of Computer Science and Information Technology,**

**Virtual University of Pakistan**

# Service

- **Services are the applications that customers need and use.**
- **Different components that make an application work includes**
  - the software,
  - the hardware,
  - and the operations that bring it all together.
- **A service may have direct customers,**
  - such as a web-based application,
  - or it may be invisible to most people, such as a database that is used by other services.
- **The fundamental purpose of System Administration is to provide services**
  - Computers and software are useful only when they are actively providing a service, not sitting in a box.
  - Services do not run themselves; it is a human process to create a service as well as to maintain it.
  - A good service meets customer requirements, is reliable, and is maintainable.
- **A service begins life as a set of requirements.**
  - Customers and stakeholders have needs, and the service is created to fulfill them.
  - These requirements are used to create the technical design.
  - The service is deployed and launched.
  - Launching a service is more complex than one would expect.

- **The launch of a service is not  the end of this process, but it is really just the beginning.**
- Services usually run much longer than it takes to create them.
- A running service requires maintenance and upkeep.
  - As demand grows it is scaled to handle the larger workload.
  - Requirements change over time, and new software releases must be deployed to fix bugs and add features.
  - systems fail, disaster planning is part of any service. This includes basics such as data backups and more advanced practices such as service relocation.

- Services need to be supported.
  - They must be monitored, so that problems are detected and fixed.
  - There needs to be a team that handles alerts from the monitoring system as well as end-user reports of problems.
  - For many services, the customers can request moves, adds, changes, and deletes.
  - There needs to be a mechanism for them to do so, and a team that is responsible for handling those requests.

# Types of Services

- **Generally, the computing environment and processes depends on the type of services being offered.**
  - Homes and very small offices typically have a few services; often they simply rely on their ISP for foundational services.
  - Larger organizations have a rich environment of services: from the applications that run the company, to the foundational services and infrastructure on which they depend.
  - A large organization's fundamental services are run like an in-house ISP.
- **Services can be categorized into following categories**

| Foundational services | Basic services | Primary Applications | Back Office Systems |
|---|---|---|---|
| ▪ Create the platform that other services rely on. | ▪ User-visible applications that most people have come to expect in an organization. | ▪ Applications generally drive business functions | ▪ include the databases and other behind-the-scenes |
| ▪ DNS, DHCP, directory services, network access (WAN and LAN), and Internet gateways | ▪ Examples include printing, email, file storage, chat/IM, and VoIP/phone service. | ▪ Examples are payroll, inventory management, enterprise resource planning (ERP), supply chain management, and so on | ▪ services that support applications |
| ▪ While they are generally invisible to users, failures are highly visible because they affect many services. | ▪ Because of their visibility and pervasiveness, people have developed expectations that these services will be reliable and always available. | | |

- Together they form the company's critical path The company cannot function without them.

IT601 – System and Network Administration

# Service Requirements

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Service Requirements Process and their Types

- **The determination of requirements for a services follows two steps.**

  1 – Kick off Meetings , 2 – Written Requirements

  **1 - Starting with a Kick-Off Meeting**

  - Creating and launching a new service cannot be done alone.
  - Good to start with requirements gathering via email, IM, teleconference, and video chat.
    - Nevertheless, it is critical to start out with a face-to-face meeting; the kick-off meeting
      - A kick-off meeting should have all the stakeholders present
    - Stakeholders are all the key people affected by or involved in the project

  - **What to expect from Kick off Meeting?**
    - agreement on the goal of the new service and the problem being solved
    - A timeline for completion
    - an approximate budget
    - It is no possible to resolve all these issues.
      - Mark them open.
      - Delegate each unresolved issue to a participant who will be accountable for its resolution.

  - **Benefits of Kick off Meeting?**
    - Introduce stakeholders and their roles
    - Improves collaboration
    - Resolves the issues faces in emails, IM e.t.c.

# Service Reequipments and their Types

## 2 - Gathering Written Requirements

- The next step is gathering requirements.

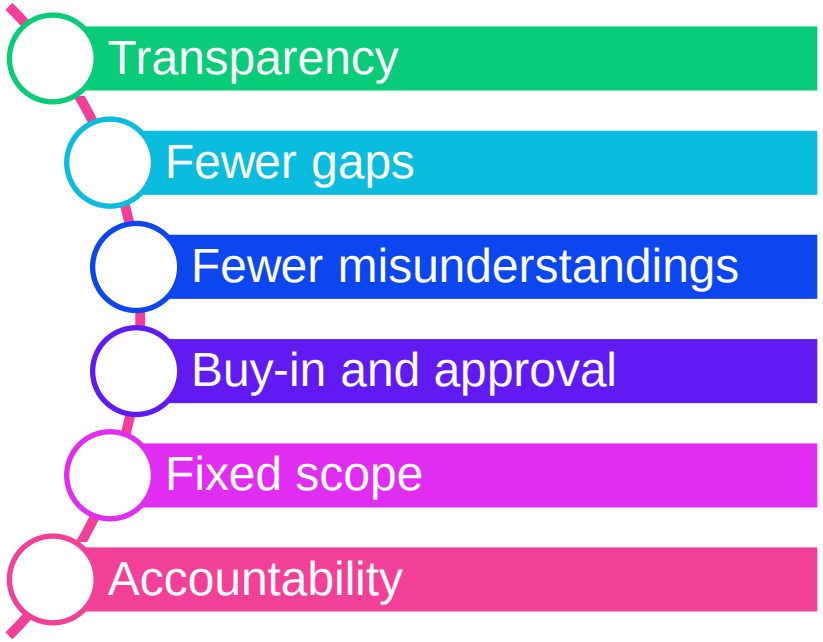
- **Goal of Requirements**
  - Requirements are a list of what the service will be able to do. Requirements should list desired functionality, features, and capabilities. Focus on the end goal: what the system will enable people to achieve.

- **Serve as checklist for other steps**
  - The list of requirements guides all the other steps: design, engineering, implementation, testing, and so on.
  - It avoids confusion and finger-pointing.
  - It sets expectations with customers. It can even be a checklist that lets you know when you are done.
  - Avoid boiling the ocean.
    - Some requested features will be too difficult or expensive to implement.
    - Some may not be appropriate for this service to provide but may be better suited to a different service.
    - Some may be required by too few people to be worth implementing, at least in the initial release.
    - Some requirements may have external dependencies on other groups, projects, or services.
    - Make sure to flag each of these external dependencies and have a plan for what to do if the external dependency is not delivered on time.

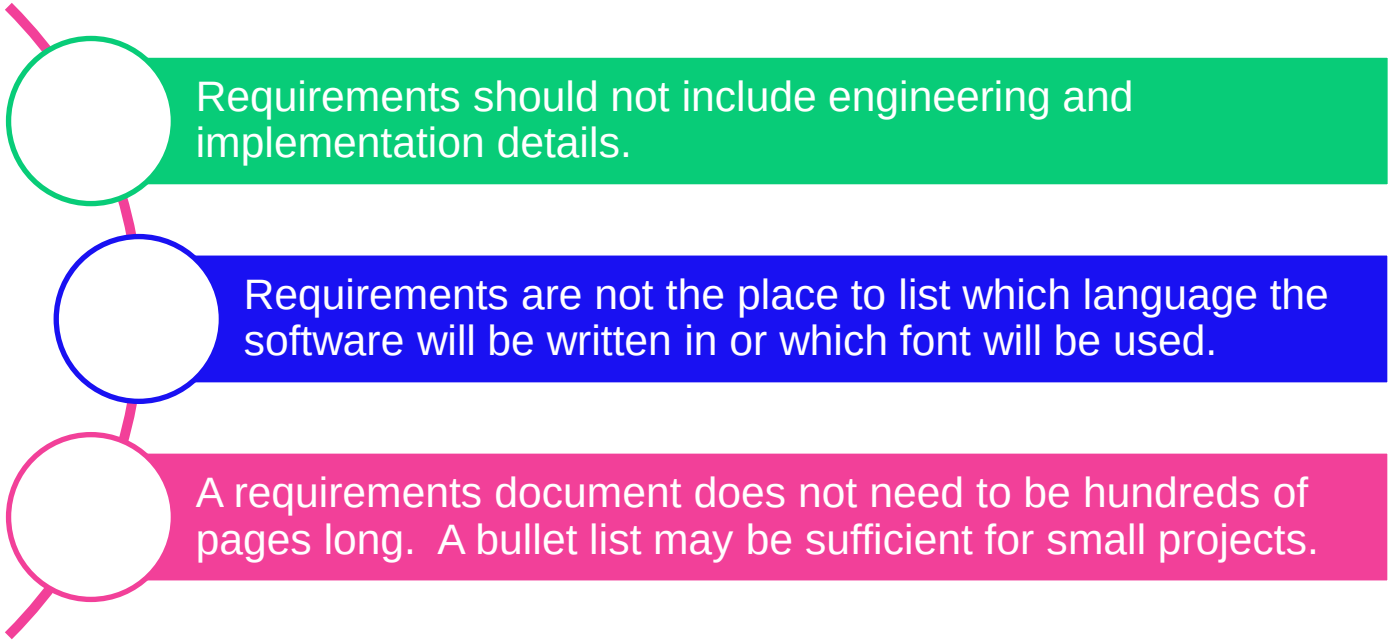
# 2 - Gathering Written Requirements

- **Requirements are written down. They are not simply agreed to in a verbal discussion, tracked on a dry-erase board in your office, or kept in your head.**
  - Writing them down in a shared requirements document has many benefits:

- **Benefits**




- **What to avoid**



# 2 - Gathering Written Requirements

- **Guidelines**
  - **Define terminology early in the document. Getting agreement to the ontology that will be used is very important.**
    - Ontology is the system of terms and definitions that define the system and its parts.
    - Often during a heated debate, one realizes that everyone is using the same words but meaning different things.
    - Pausing to get agreement on terminology helps everyone see eye-to-eye.
    - At that point we often realize we were closer to agreement than we had previously realized.
  - **Annotate the importance of each requirement.**
    - Marking each as “must have,” “desired,” or “optional/nice to have” communicates intent.
    - It gives the system designers flexibility and helps implementors prioritize work when deadlines are looming, and features must be cut.

# Types of Requirements

- **The requirements can be  classified into following types**

Customers

Scope, Schedule, and Resources

Operations

Open Architecture

# 1 - Customer Requirements

**A - Describing Features**

**B - Questions to Ask**

**C - Service Level Agreements**

**D- Handling Difficult Requests**

# A - Describing Features

- **Record the "what," not the "how."**
  - The requirements should focus on the list of features, stated from the perspective of what the customer should be able to accomplish using business terms, not technical terms.
  - **Example**

    It is better to record a requirement such as "the user should be able to send email" than "there should be a button on the left that the user clicks to send email." The latter assumes a lot about the interface. Why should the designers be required to include a button even though they have a better idea?

- **Do not proscribe particular technology**

# B - Questions to Ask

- **Ask how, why, where, and when, as if you are a journalist conducting an interview.**
- **Example Questions**
  - How do customers intend to use the new service?
  - Why do they need each feature?
  - Where, when, and how will the system be accessed?
  - How critical is the service to them, and which levels of availability and support do they need for service?
- **Determine how large the customer base for this service will be and what sort of performance they will expect from it.**
- **Another way to record requirements is through use cases that tell a story. As long as the story can be completed by the final design, the use case is sufficiently covered.**
  - In Agile methodology, the format is “As a <type of user>, I want <some goal> so that <some reason>.”
  - Examples

    For example, one might specify: “As a payroll clerk, I want to be able to enter future payroll adjustments so that I don’t need to do them all on the day they take effect.”

    Another example: “As a user, I want to indicate which folders should not be backed up so that my backup drive isn’t filled up with things I don’t need saved.”

# C - Service Level Agreements

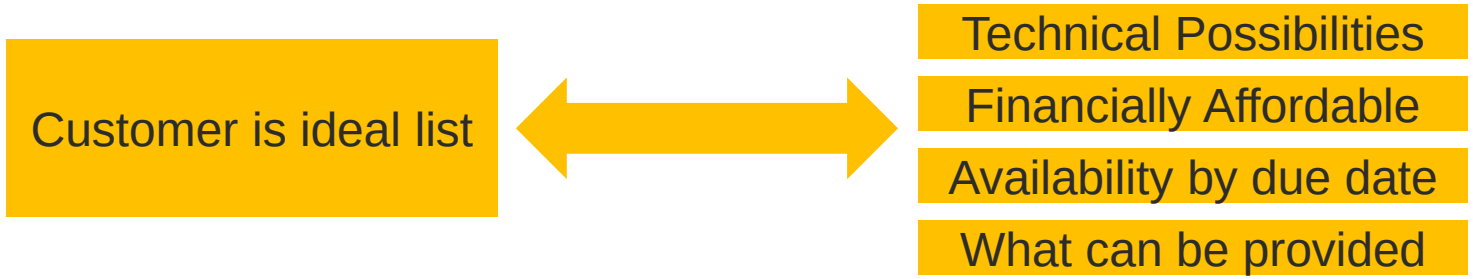
- **The requirements document should include the service level agreement (SLA), or at least some general specifications that can be used to build an SLA.**
  - What SLA Should Incudes.

| Enumerate Services | Categories Problems | Escalation Process | Penalties | Resources | Customer Needs | Future Needs |
|---|---|---|---|---|---|---|
| It should enumerate the services with the level of support. | Categorizes problems by severity and commits to response times for each category. | Defines an escalation process to increases the severity of non-resolved problems. | Specifies penalties if the service provider fails to meet a given standard of service. | A tool to plan the resources for project. | Document the customers' needs and set realistic goals for the SA team in terms of features, availability, performance, and support. | Document future needs and capacity so that all parties understand the growth plans, and their approximate costs. |

- The SLA is always discussed in detail and agreed on by both parties.
- The SLA creation process is a forum for the SAs to understand the customers' expectations and to set them appropriately, so that the customers understand what is and isn't possible and why.
- The SLA is a document that the SA team can refer to during the engineering process to make sure that they meet customers' and their own expectations and to help keep them on track.

# D - Handling Difficult Requests

- **Requirements gathering is a collaborative process.**
- **The ultimate goal is to find the middle ground.**




- **Your job is to educate as much as it is to record features.**
  - Don't become upset when a customer asks for something technically unreasonable; if the customer knew technology as well as you do, the customer would be an SA.
  - Try to understand the end goal and see how that can be achieved instead.
    - Rather than say "no," focus on the need, not the technology.

- **Examples**
  - A feature that will take years to develop is not reasonable for a system that must be deployed next month.
  - A feature that will cost $1 million is not reasonable for a project with a budget that's in the thousands of dollars.
  - A small company with only one or two SAs will not get 24/7 support, no matter how much the company wants that.

IT601 – System and Network Administration

# Services – Requirement Gathering

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# 2 - Scope, Schedule, and Resources

- **A project plan generally has three elements.**


- If a project is at risk of not making its deadline, one or more of these will have to change. Features will need to be dropped, the deadline will need to be extended, or the budget will need to be increased.

- **Flexible Vs Inflexible Features**

  - For example, if the project needs to be complete in time for the Eid shopping season, the schedule is inflexible; so the features and/or budget needs to be flexible. If the budget is inflexible, it must be possible to remove features or extend the schedule.
  - If the project is to send humans to the moon and safely return them, the features are inflexible; a one-way trip is not an option. In this case, either the budget or the schedule must be flexible.

  - **Management Buy-in is important**
    - management to decide as to which of these elements are flexible and which are inflexible at the start of the project.
    - The buy-in process helps communicate priorities to all stakeholders, prevents a lot of confusion all around, and helps focus people on the right priorities.
    - It also has the benefit of discovering early on if management is unwilling to be flexible in any of the three areas.

# 3 - Operational Requirements

Operational requirements are the things that make a system maintainable. These are the features required if we are to have a system that is reliable, scalable, and secure. The features are often invisible to the customers; yet reliability is the most visible aspect of a service.

A - System Observability

B - Remote and Central Management

C - Scaling Capacity

D - Software Upgrades

E - Environment Fit

F - Support Model

G - Service Requests

H - Disaster Recovery

# A - System Observability

- **System Observability means  the ability of a manageable system that we can see what it is doing.**
  - The system must provide enough visibility and introspection to debug it when there are problems, optimize it when it is slow, and detect when resources are low before they become a problem.
  - There should be situational awareness to reason about it, predict its future behavior, and do capacity planning.
  - A system may runs itself, or that a company's goal may be to build a system that it doesn't need maintenance.
    - It should be their goal, but until it is achieved vendors must build observable systems.
- **A system which is not observable, we cannot manage it, fix it, or scale it**
- **Basic Tools of Visibility**
  - **System Logging**
    - The most basic kind of visibility is logging. All parts of the system should log events, transitions, and operations.
    - Logging should be adjustable to provide more detail when debugging. The level of detail may include logging every API call or function call.
  - **System Monitoring**
    - Ability to monitor not just the system 's up or down, but also internal resources, timing, bandwidth, disk space usage and requests  delays.
    - The system is up or down information cannot prevent outages; you can only respond to them. There  should be warnings if too late to prevent an outage.
    - The new system should integrate into existing monitoring systems, dashboards, trouble-ticket systems, and other appropriate visibility tools.

# B - Remote and Central Management

- **Services should be remotely managed. If you have to be on the console of a machine to do rudimentary maintenance and configuration, each task will be more time consuming.**
  - This is also an indication that automating management tasks may not be possible.
- **Two basic reasons for remote servers**
  - **Specialized Premises**
    Servers are generally located in machine rooms where power and cooling can be managed more effectively. Remote management permits servers to be located anywhere. It is less expensive to rent colocation space elsewhere than to build a computer room in your office. It is often also more reliable.
  - **Geographic Redundancy**

    Services that are replicated in many places must be manageable in a centralized fashion. Remote management opens up the possibility of deploying instances of the service around the world. Configuration changes and upgrades should not require manually updating each instance.

- **Consider high latency over long-distance connections**
  - Protocols may time out. User interfaces may work but can be unusably slow. This is not a limit of the technology, but rather bad software engineering. Plenty of APIs and user interfaces work well over high-latency links—and so should management tools.

# C - Scaling Capacity

- **If a service is successful, more people will want to use it. Over time it will need to scale to handle more users, more transactions, or more capacity.**

- **There are two ways services generally scale.**
  - **Scale-up**
    means getting a bigger system. To process more requests, more users, or more disk space, the main system is replaced with one that is bigger and faster.
    - For example, to scale up a web server so that it can process more requests per second, you might replace the computer with one that has a faster CPU and I/O system.
  - **Scale-out**
    means the system is expanded by adding more replicas.
    - For example, a web-based service is often made up of web servers behind a load balancer. Scaling out means adding more redundant web servers (replicas). Each web server scales out the system to handle more requests per second.

# D - Software Upgrades

- **A system must have a reasonable way to upgrades.**
  - The software
  - The firmware

- **Two types of upgrades.**
  - Interruption Free
  - With Interruption

- **Ideally, it should be possible to test a new software release outside of production and upgrade the service without interrupting it.**

- **new software releases**
  - Software is never done. There will always be the need for new software releases, even if every feature is complete and every bug is fixed.
  - New security holes are discovered constantly, and may not even be in the vendor's code but in a library or framework it used.
  - More importantly, software is malleable and vendors should be providing new features and improvements over time. There's always room for improvement.

# D - Software Upgrades

- **Upgrades should be a Process**
  - Upgrade processes should exist and be automatable.
  - The process should integrate into the organization's software patching automation.
  - It should not involve walking to each desktop machine for a manual update.
  - Roll out the upgrade slowly
    - It should be possible to upgrade some clients, but not all, so that we can mitigate risk by rolling out the upgrade slowly.
    - If all clients have to be upgraded simultaneously, then there is no way to test the upgrade.
    - As number of clients grows, the concept of upgrading clients at the same time becomes more and more inconceivable.
- **Test environment**
  - The more people depend on a service, the more visible a failed upgrade will be.
  - Ensure to make it possible to deploy the service to a test environment for the purpose of both testing the upgrade process and testing the new release itself.
- **Production assurance test (PAT) or user acceptance test (UAT)**
  - PAT and UAT environments are run at production standard, but upgraded ahead of the main production environment.
  - Select volunteers who use PAT or UAT environment, rather than the production environment, at all times, to ensure that when an upgrade is rolled out in that environment, it is really used and production tested.

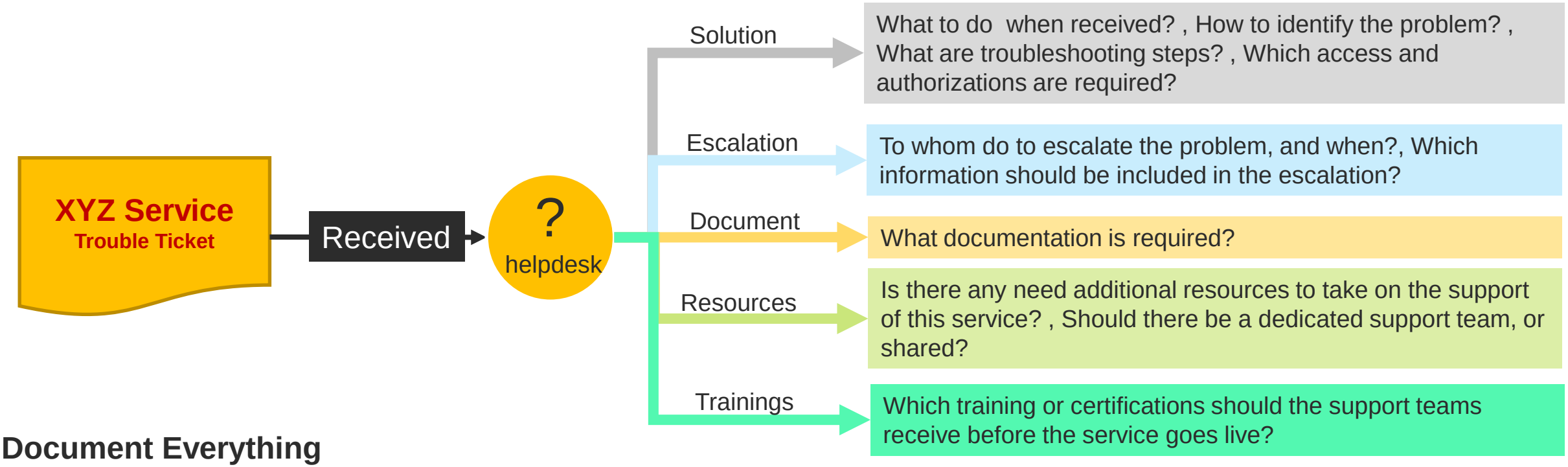
# E - Environment Fitness

- **The better a service fits into the existing IT environment, the easier it is to adopt it in your environment. Integration issues are reduced, and less training or skill development is required.**
  - For example, if you use ActiveDirectory, then using a product that interfaces with it will be easier than using a product that has its own directory service. If the service runs on an operating system with which the team is unfamiliar, using it will require not just additional training, but entirely new procedures that have to be developed for tasks such as data backups, account creation, and configuration.
- **For a small project, it is reasonable to make it an operational requirement that the service fit into the existing OS infrastructure, directory service, and so on. Larger projects require more flexibility.**
  - For example, a large SAP ERP deployment is a self-contained environment unto itself. Therefore it is more acceptable that it may introduce a new operating system to your environment. Most likely it will have its own system administration team.
- **Environmental Factors**

| | |
|---|---|
| ▪ Operating system<br>▪ Backup/restore facilities<br>▪ Monitoring system<br>▪ Network equipment vendor and routing protocols<br>▪ Security auditing systems | ▪ Trouble-ticket systems<br>▪ Dashboards and consoles<br>▪ Network directory services<br>▪ DNS and other name services<br>▪ Printing systems |

# F - Support Model

- **A key component in the success of a new service is support processes.**
  - This aspect is often forgotten or neglected by the engineers, who are intent on developing the technical solution.
- **Problems arise as service goes into production**
  - When they have problems with it, they will raise tickets or call the helpdesk.
  - If the helpdesk does not know about the new service, the users of the service will receive very poor support, and will have a bad impression of the service, no matter how well it has been implemented.




- **Document Everything**
  - The project plan will need to include time and budget resources to implement the answers to these questions. Documentation may need to be written, new staff hired, training completed, helpdesk operations updated, monitoring systems expanded, and so on.

# G - Service Requests

- **Most services will include standard service requests that people can raise.**
  - Depending on the service, these may be access or authorization requests, data changes, configuration changes, resource requests, and so on.
- **A checklist for building requirements might look like this:**

| Checklist | |
|---|---|
| | • What should people be able to request? |
| | • How do people raise these requests? |
| | • Which changes need to be made to the service request system to support these new service requests? |
| | • Which approvals are required for each of these service requests? |
| | • Who will process the requests? |
| | • Which access and authorizations will that team need? |
| | • Which runbooks does that team need to correctly service the requests? |
| | • To what extent can, or should, these requests be self-service or fully automated? |

- **The answers to these questions will provide some new requirements.**
  - For example, depending on whether the requests should be self-service or automated, there are API requirements that need to be taken into consideration. The requirements and dependencies on other systems, such as the service request system, also need to be documented, agreed to, signed-off on, budgeted, and planned.

# G - Disaster Recovery

- **Failures cannot be eliminated.**
  - Hardware fails, buildings lose power, Internet access gets cut, and so on.
  - Assuming everything will always function perfectly is irrational. Instead, we need to prepare for failures so that we know how to recover from them.

- The most basic disaster recovery requirement is that there must be a way to perform backups and restores. This includes total system restores, as well as the ability to restore an individual file or customer data point.

- The most common data restore requirement is for something that was deleted by accident. As a result, the operational requirement of restoring individual data points has been implemented in most systems, often with no SA intervention required.

- It can be useful to be able to separately back up the system configuration and the user data. Configuration files that are human-readable are preferable to large binary blobs. Being human-readable makes these files easy to store in source code repositories, makes it easy to produce historical diff listings, and so on.

# 4 – Architectural Requirements

- **A new service should be built around an architecture that uses open standards and open protocols.**
- **Open Vs Closed Service Architectures**
  - **Open Architecture**
    - Protocols, file formats, and APIs that are publicly documented so that others can write to those standards and make interoperable products without having to worry about royalties or patent restrictions.
    - Any service with an open architecture can be more easily integrated with other services that follow the same standards.
  - **Closed Architecture**
    - uses standards, protocols, APIs, and file formats that are owned by one company, are controlled by that one company, and do not interoperate with other products.
    - Other products are prevented from using the standard because the standard is not publicly documented, because it requires licensing, or because the vendor forbids it.
    - Vendors use proprietary protocols when they are covering new territory or are attempting to maintain market share by preventing the creation of a level playing field.
- **What is difference between a Protocol Versus a Product?**

# 4 – Architectural Requirements

- **Standard Vs Extended Products**
  - That's not very customer oriented
- **Benefits  Of deploying open products**
  - Promotes Competition
  - Multi-Vendor Environment
  - Interoperability
  - More Choices and Freedom
  - Easier Support
  - don't require gateways to the rest of the world.
- **A better way is to select protocols based on open standards, permitting users and operations to select their own software.**
  - Open protocols and file formats typically change only in upwardly compatible ways and are widely supported, giving you the maximum product choices and maximum chance of obtaining reliable, interoperable products.
  - Such as Internet Engineering Task Force (IETF) and Institute of Electrical and Electronic Engineers (IEEE) Standards.

# IT601 – System and Network Administration

<End>

Arif Husen

**Department of Computer Science and Information Technology,**

**Virtual University of Pakistan**

IT601 – System and Network Administration

# Service Planning and Design

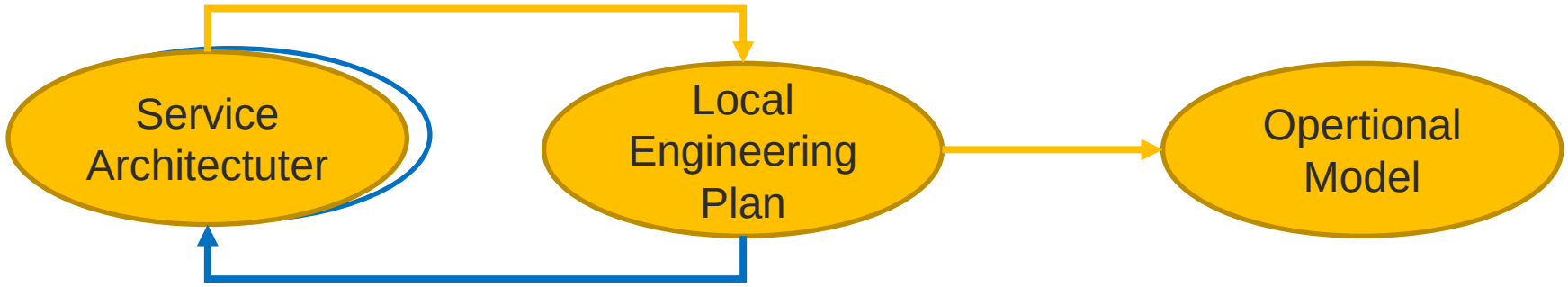
Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

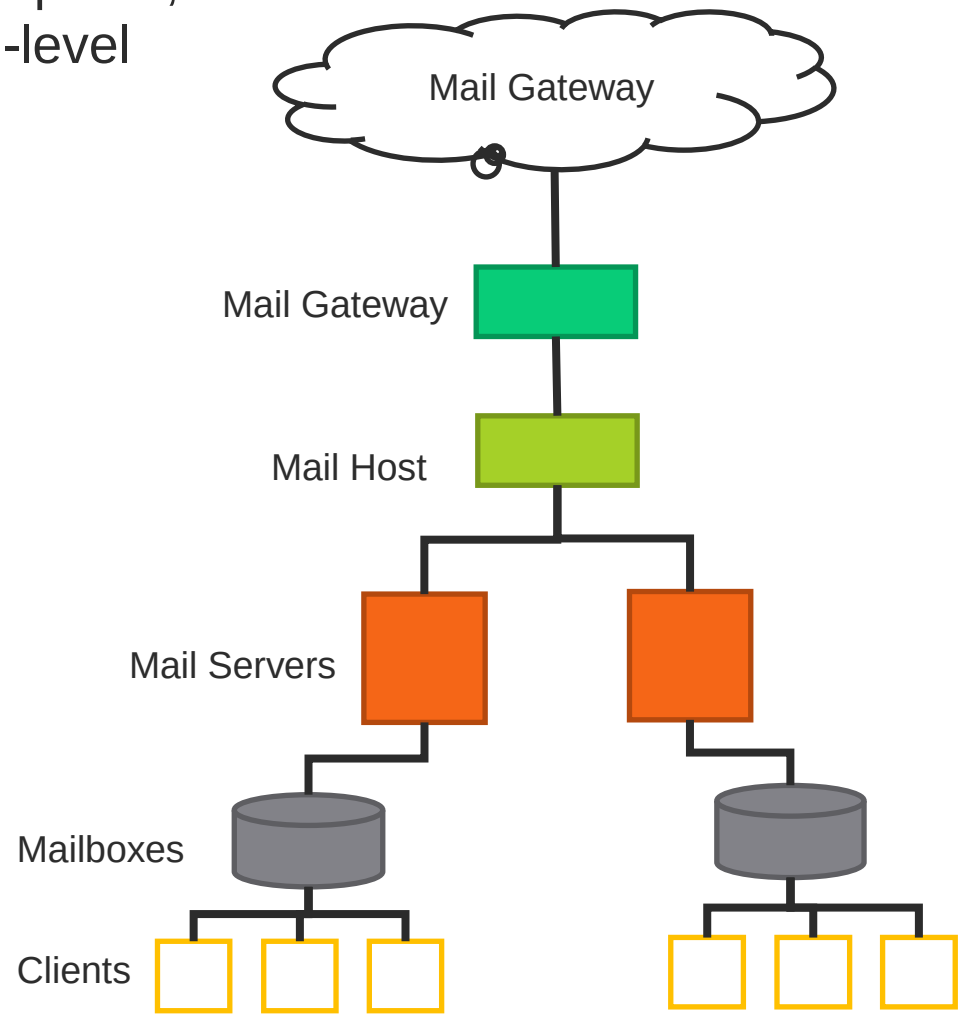
# Designing a Service

- **Service engineering involves designing how a product will be configured and used in your environment. With a simple service, it is a matter of installing the software.**
- **But most services are not simple. In such a case, we must design the service architecture and local engineering plans.**
- **The service design includes the following plans. All these must be defined and documented.**
  - The service architecture
  - The local engineering plans
  - The operational model or support model



# Service Architecture

- The service architecture is the high-level design. It describes how the parts will work together, which resources are required, and so on. Service Architecture  is also  known as High-level description




- It Includes:
  - Service Type
  - Sites and Clusters
  - Type of Machines, Technologies, Topologies and Storage, Networks
  - IP addressing Design

# Local Engineering Plan

Local Engineering Plan is also  known as Low Level Design description

It Includes:

- Specific Component Levels  Plan
- Product Ordering and Product Codes
- Cable, Mounting Brackets and Connectors their types
- Detailed IP addressing Hirerachy

# Operational Plan

Operational Plan is lso  known as Service Support Plan.

It Includes:

- Operational level agreement (OLA)
- The service level agreement (SLA) for the service
- Tho supports the various components of the service

# Guidelines for Service Designing

- Simplicity
- Vendor-Certified Designs
- Dependency Engineering
- Decoupling Hostname from Service Name
- Support

# Simple Designs

- When engineering a service, your foremost consideration should be simplicity. Strive to create the simplest solution that satisfies all the requirements.
- **Simple Services**

| Maintenance | Expansion | Integration |
|---|---|---|
| ▪ Easiest to maintain | ▪ Easiest to expand | ▪ Easiest to integrate with other systems |

- **Complex Services**

| Confusion | Mistakes | Usage | Slower | Costly |
|---|---|---|---|---|
| ▪ Leads to confusion | ▪ Prone to mistakes | ▪ Difficult to use | ▪ Makes everything slower | ▪ More expensive in setup cost and maintenance costs. |

- During the engineering phase, create multiple design proposals. Consider each of them for its simplicity, manageability, cost, and so on. Revise, revise, revise.
- Good engineering is not about having the ability to create perfection in the first draft, but rather about being able to iterate over many revisions, improving each one until we have the design we want.

# Use Vendor-Certified Designs

- **Vendors of IT products have the engineering guidelines.**
  - They proscribes minimum CPU, RAM, and disk requirements.
  - They also publish certified engineering designs or vendor best practices.
  - There may be separate engineering recommendations for use with up to one dozen, one hundred, or one thousand users.

- **Vendor designs are**

  best practice engineering designs | Real-world experience | labs and real world Tested | Stress Tested

- **Trust vendor designs but verify them**
  - Straying too far from the vendor's recommendations will result in the loss of support.

- **But even with the use of the certified engineering design exactly, some local decisions are part of local engineering design such as;**

  hardware vendor | model number | specific configurations, | IP addresses | network switch connections | deployment racks | Optional parameters

# Dependencies

- **The reliability of a service is greatly influenced by its dependencies.**
  - A system is never more reliable than the most unreliable system it has a hard dependency on.

- **The larger and more complex a system becomes, the more dependencies it has.**
  - These dependencies can be managed during the engineering process in ways that greatly improve the resulting system's reliability

- **Primary Dependencies**
  - Primary reliability refers to the core system reliability.
  - SAs do not have much control over reliability of software itself especially for 3rd party products.
  - Sas can control are their decisions about the hardware it runs on.
  - Generally, enterprise software tends to be written assuming the hardware will never fail thus for SAs, the options are reliability features such as mirrored boot disks, ECC RAM, dual NICs etc.

# External Dependencies

- **SAs can control many of the external dependencies that influence a service's reliability.**
- **Know what those dependencies are. A dependency matrix is a list of which subsystems depend on other subsystems.**
  - Usually the dependency matrix resembles a tree diagram, with each level relying on services in the levels below, and no cross-dependencies.
  - Used to find high-risk
  - Useful for operational tasks such as basic debugging as well as disaster planning
- **Types of External Dependencies**
  - **Hard dependency**

    The dependent service will fail if the other service fails or is unavailable
  - **Soft dependency**

    does not directly cause a failure, though it may reduce performance
  - **Graceful degradation**

    if that server is unavailable, the application may work around it by using cached data, disabling a specific feature, or requiring the data to be manually entered
- Improve the reliability by identifying hard dependencies and either eliminate them or reengineer them to be soft dependencies

# Dependency Alignment

- **Another engineering trick is to realign dependencies into smaller failure domains.**
- **A failure domain is all the services, locations, and users that are adversely affected when a particular component fails or is taken out of service.**
- **Good engineering requires taking the time to think through how we can shrink failure domains, or divide large failure domains into smaller, isolated failure domains.**

| Hardware Failure Domains | Service Failure Domains | Location Failure Domains |
|---|---|---|
| • Make Redundant domains including power strips, CPUs, and storage and VPN devices | • Align all dependencies on same machine<br>• All startup scripts should be on the same machine<br>• Rack aligned dependenies | Make sites self dependent |

# Decoupling Hostname from Service Name

- **Decouple the purpose of the machine from the name of the machine.**
- **A better choice is to design your systems to use aliases for the service name**
- **do not hardcode services to IP addresses**

# Support

- **During planning and engineering phase, consider how service will be supported during its lifetime.**
- **Support includes**
  - **Technical aspects**
    detecting problems, patching, upgrading, performing backups, and scaling the service
  - **People and processes**
    identifying and training the support groups, developing mechanisms for customers to report problems and submit requests, and providing documentation for the customers and the support teams.

- **Support planning aspects can be divided into followings**

| Monitoring | Support Model | Service Request Model | Documentation |
|---|---|---|---|

# Monitoring

- **It isn't a service if it isn't monitored.**
- **A service should be monitored for availability problems, performance, and capacity-planning.**
- **The helpdesk, or front-line support group, must be automatically alerted to problems with the service.**
- **The SA group should monitor the service on an ongoing basis from a capacity-planning standpoint.**
- **Followings must be defined and developed**

| | | | |
|---|---|---|---|
| what to monitor and what events to trigger notifications. | the process for adding new components into the monitoring system as the system is scaled up or upgraded. | how component are removed from the monitoring system. | Automation needs to be developed, and the processes need to be clearly defined and documented |

# Support Model

- **The support model needs to specify who supports the various components of the service, and what the OLAs and SLAs for each component are.**
- **The OLA and SLA for the service need to consider the OLAs and SLAs of the constituent parts, as well as the design of the service itself.**
  - In small or midsize company, many of these roles will be performed by the same people, who all know each other.
  - In large companies, there will be different teams supporting each component, they won't all know each other, and the processes and communication paths may not be clear

# Service Request Model

- **The local engineering plan should also include a service request model (SRM).**

  - The SRM defines
  - which requests users can make relating to this service,
  - who can make those requests,
  - how they make those requests,
  - which approvals are required
  - who acts on the requests
  - which change control policies apply
  - which SLA is used for turning around those requests.

- **Work with the vendor and stakeholders to define which tasks are required, document them, and then practice them in a test environment.**

- **Try to estimate the volume of requests of each type and understand from the stakeholders what would trigger a request of each type and what kind of turnaround time they would expect on each request.**

# Documentation

- **As part of support planning, operational procedures must be defined.**
- **These are different for every service, but generally include**
  - backups and restores,
  - Business continuity or disaster recovery plans,
  - tasks related to onboarding new users
  - Disconnecting users who are leaving
  - performing periodic tasks
  - Anything else required to keep the service running.
- **For each routine task, there must be a runbook that details how the task should be performed. Define the runbooks by performing each task in a test environment.**

IT601 – System and Network Administration

# Primary Services

Arif Husen

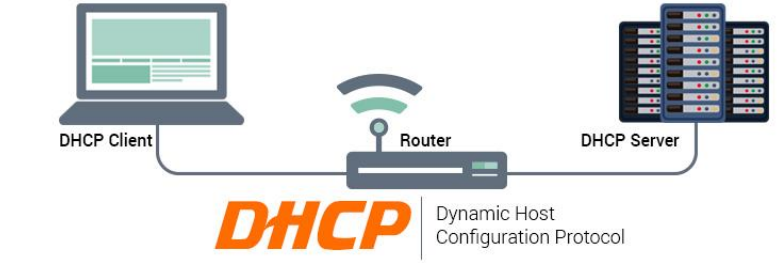
Department of Computer Science and Information Technology,

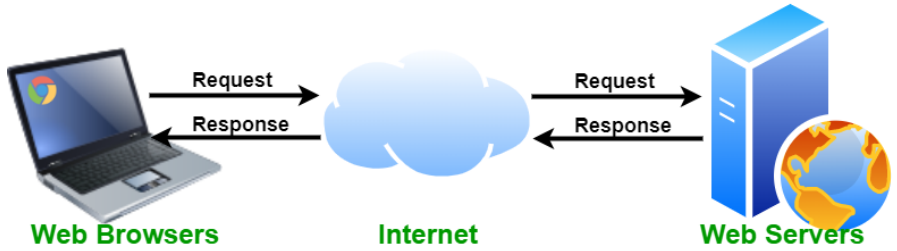
Virtual University of Pakistan

# Primary Services

- There are several common services used in almost all IT deployments.

  - DHCP Service
  - DNS Service
  - NTP Service
  - Web Service
  - File Sharing Service
  - SSH Server


IT601 – System and Network Administration

# DHCP Service

Arif Husen

Department of Computer Science and Information Technology,

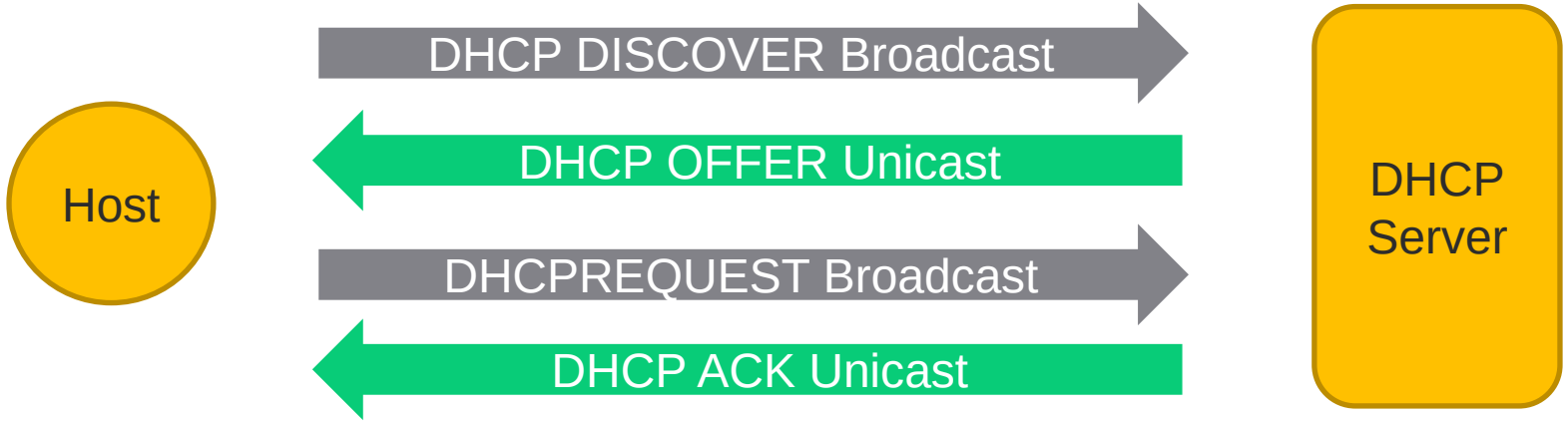
Virtual University of Pakistan

# DHCP Service

- The Dynamic Host Configuration Protocol (DHCP) is a network service that enables host computers to be automatically assigned settings from a server as opposed to manually configuring each network host.
  - It was initial defined in RFC 2131 and later on it was superseded by RFC 1541.
- Its Components are DHCP Client, DHCP Relay Agent and DHCP Server

**1 - DHCP Client**

- A DHCP client is an Internet host using DHCP to obtain configuration parameters such as an IP address.
- The basic steps that occur when a DHCP client requests an IP address from a DHCP server are as below.




- The client, Host A, sends a DHCPDISCOVER broadcast message to locate a DHCP server. A DHCP server offers configuration parameters such as an IP address, a MAC address, a domain name, and a lease for the IP address to the client in a DHCPOFFER unicast message

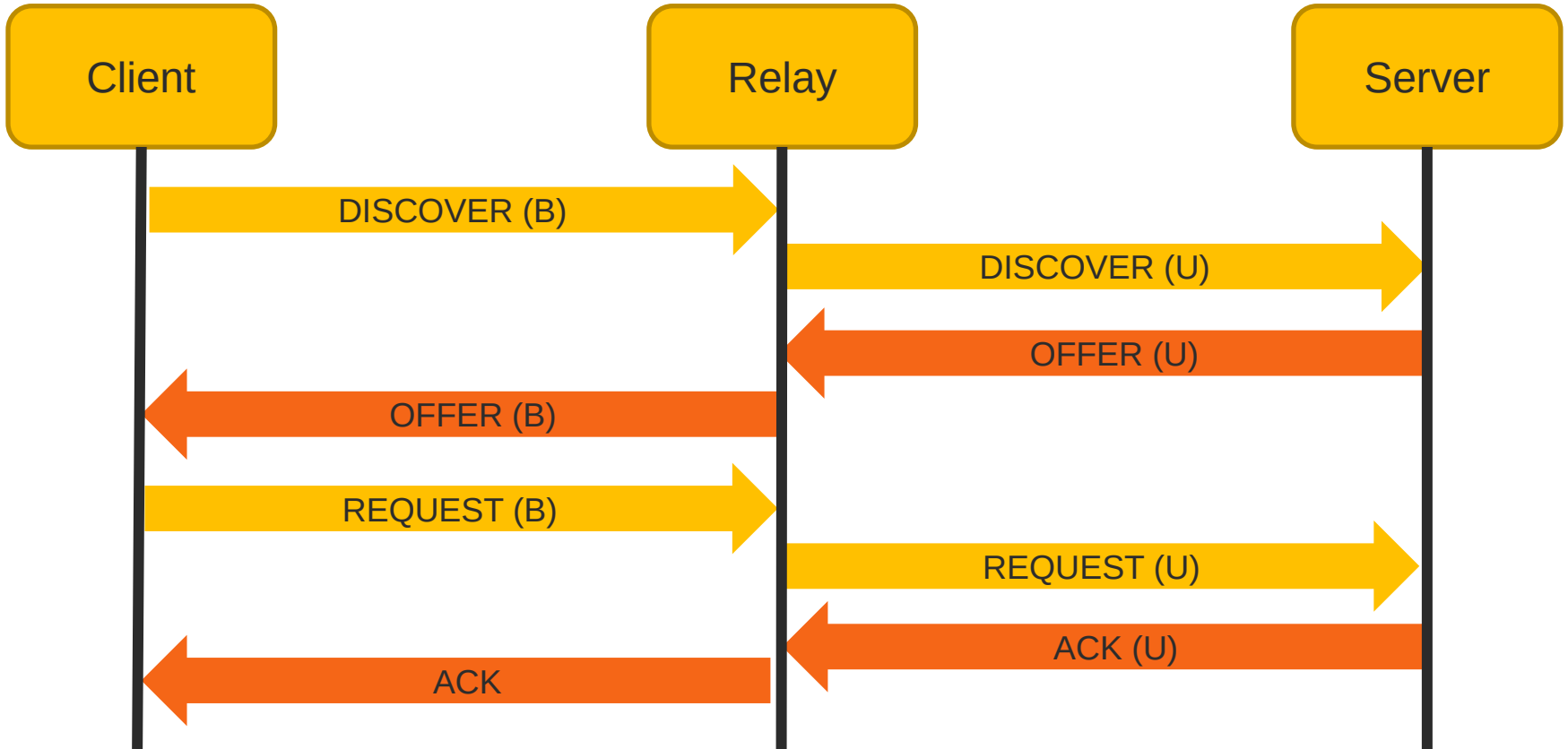
# DHCP Client Options

The DHCP client provides flexibility by allowing the following options to be configured for a DHCP client:

- Option 12—This option specifies the name of the client. The name may or may not be qualified with the local domain.
- Option 51—This option is used in a client request (DHCPDISCOVER or DHCPREQUEST) to allow the client to request a lease time for the IP address.
- Option 55—This option allows the DHCP client to request certain options from the DHCP server.
- Option 60—This option allows the user to configure the vendor class identifier string to use in the DHCP interaction.
- Option 61—This option is used by DHCP clients to specify their unique identifier. DHCP servers use this value to index their database of address bindings. This value is expected to be unique for all clients in an administrative domain.
- Option 120—This option is used to specify a 32-bit (binary) IPv4 address to be used by the Session Initiation Protocol (SIP) client to locate a SIP server.
- Option 121—This option is used to configure classless static routes by specifying classless network destinations; that is, each routing table entry includes a subnet mask. Upto ten classless static routes are supported using option 121 on the DHCP client.
- Option 125—This option is used by DHCP clients and servers to exchange vendor-specific information

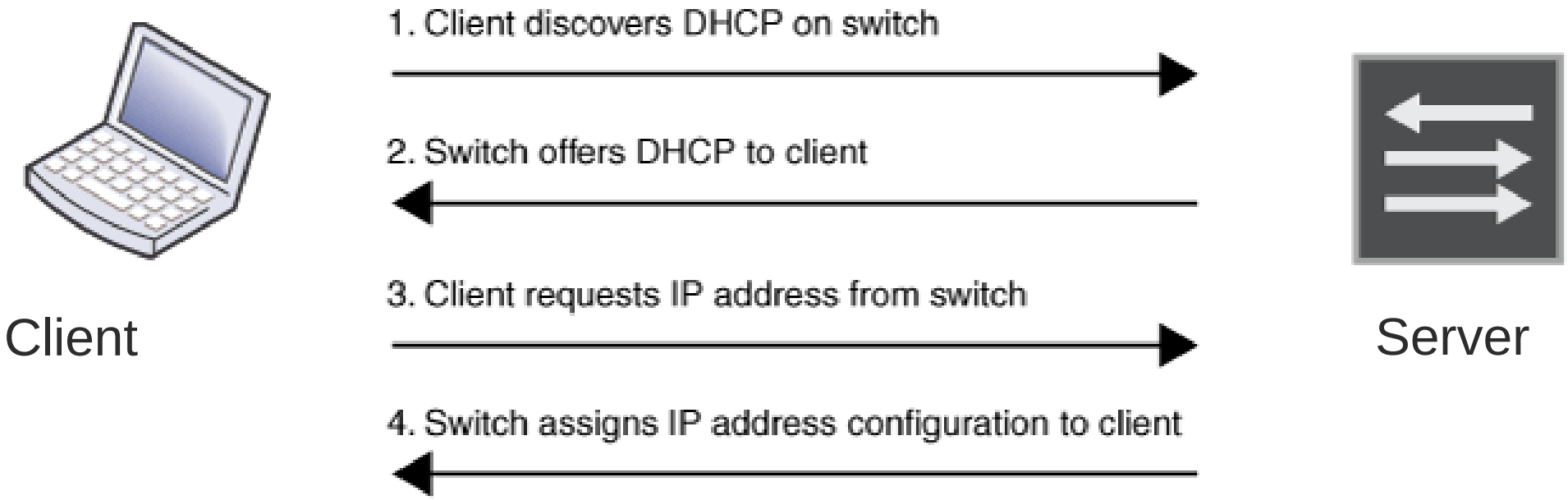
# DHCP Relay Agent

- The function of the DHCP relay agent is to forward the DHCP messages to other subnets so that the DHCP server does not have to be on the same subnet as the DHCP clients.
- The DHCP relay agent transfers DHCP messages from the DHCP clients located on a subnet without a DHCP server, to other subnets.



# DHCP Service

- Computers configured to be DHCP clients have no control over the settings they receive from the DHCP server, and the configuration is transparent to the computer's user.



# DHCP Service

- The most common settings provided by a DHCP server to DHCP clients include:


- A DHCP server can also supply configuration properties such as:


- The advantage of using DHCP is that any changes to the network, such as a change in the DNS server address, only need to be changed at the DHCP server, and all network hosts will be reconfigured the next time their DHCP clients poll the DHCP server.

- As an added advantage, it is also easier to integrate new computers into the network, as there is no need to check for the availability of an IP address. Conflicts in IP address allocation are also reduced.

# DHCP Service

- A DHCP server can provide configuration settings using the following methods:

## Manual allocation (MAC address)

1 - Identify the unique hardware address of each network card

2 - Supplies a constant configuration each time the DHCP client makes a request to the DHCP server

3 - ensures that a particular address is assigned automatically to that network card, based on its MAC address

## Dynamic allocation (address pool)

1 - The DHCP server assigns an IP address from a pool of addresses for a period of time

2 - clients receive their configuration properties dynamically and on a “first come, first served” basis

3 - Server Releases the address to address pool after client is no more on networks or lease time expires

4 - After this period, client must renegotiate the lease with the server to maintain use of the address.

## Automatic allocation

1 - the DHCP automatically assigns an IP address permanently to a device, selecting it from a pool of available addresses.

2 - Usually, DHCP is used to assign a temporary address to a client, but a DHCP server can allow an infinite lease time.

# Configuring DHCP Server on Linux

**1. Install DHCP Server**

- sudo apt install isc-dhcp-server

**2. Configure the DHCP Server**

- Backup Original Configuration file
  sudo mv /etc/dhcp/dhcpd.conf{,.backup}
- Create and edit the new configuration file
  sudo nano /etc/dhcp/dhcpd.conf
- Assigning Random IP Addresses from a pool

```
# a simple /etc/dhcp/dhcpd.conf
default-lease-time 600;
max-lease-time 7200;
authoritative;

subnet 192.168.1.0 netmask 255.255.255.0 {
 range 192.168.1.100 192.168.1.200;
 option routers 192.168.1.254;
 option domain-name-servers 192.168.1.1, 192.168.1.2;
#option domain-name "mydomain.example";
}
```

# Assigning Static IP Address to a client

Assigning Static IP Address to a client

get the MAC Address of a client

ip a

```
[21:13:42] $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp6s0f0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group defaul
t qlen 1000
    link/ether 00:90:f5:b2:bd:cd brd ff:ff:ff:ff:ff:ff
3: wlp5s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether e0:91:53:31:af:ab brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.218/24 brd 192.168.1.255 scope global dynamic noprefixroute wlp5s0
       valid_lft 83879sec preferred_lft 83879sec
    inet6 fe80::edb7:8f6:3d9e:8ee9/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

# Verifying the DHCP Server

```
host archmachine {
hardware ethernet e0:91:53:31:af:ab;
fixed-address 192.168.1.20;
}
```

**3. Bind the DHCP Server to an interface**

**4. Restart the DHCP Server** sudo systemctl restart isc-dhcp-server.service

**5. Check the status of the DHCP Server** sudo systemctl status isc-dhcp-server.service

```
LinuxForDevices> systemctl status isc-dhcp-server.service
● isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabl>
     Active: active (running) since Fri 2020-09-18 16:50:50 UTC; 26s ago
       Docs: man:dhcpd(8)
   Main PID: 18083 (dhcpd)
      Tasks: 4 (limit: 2282)
     Memory: 4.5M
     CGroup: /system.slice/isc-dhcp-server.service
             └─18083 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /e>

Sep 18 16:50:50 localhost dhcpd[18083]: ** Ignoring requests on docker0.  If this is not what
Sep 18 16:50:50 localhost dhcpd[18083]:    you want, please write a subnet declaration
Sep 18 16:50:50 localhost dhcpd[18083]:    in your dhcpd.conf file for the network segment
Sep 18 16:50:50 localhost dhcpd[18083]:    to which interface docker0 is attached. **
Sep 18 16:50:50 localhost dhcpd[18083]:
Sep 18 16:50:50 localhost dhcpd[18083]: Listening on LPF/eth0/[redacted]/24
Sep 18 16:50:50 localhost dhcpd[18083]: Sending on   LPF/eth0/[redacted]/24
Sep 18 16:50:50 localhost dhcpd[18083]: Sending on   Socket/fallback/fallback-net
Sep 18 16:50:50 localhost sh[18083]: Sending on   Socket/fallback/fallback-net
Sep 18 16:50:50 localhost dhcpd[18083]: Server starting service.
lines 1-20/20 (END)
```

# Best Practices for DHCP Administration

- Following a few DHCP best practices will keep your network running at it's best.
  - **Have an appropriate amount of established IP addresses.**
    - Get a good idea of the number of IP addresses that are going to need to have IP addresses assigned.
    - Remember that in today's networks, we're talking about more than just computers.
    - Other devices that may be requesting IP addresses are VoIP phones and mobile devices, just to name a few.
    - On top of all that, your DHCP scope should leave room for future growth as well.
  - **Avoid overlapping static addresses.**
    - While you can use DHCP server settings to assign static reservations to most devices, there will still be some devices on your network that need to keep the same IP address via manual configuration.
    - When creating your DHCP scope, be sure to understand any IP addresses that are already being used with manual configurations.
    - You can do a network scan or refer to your network map to see what IP addresses are already being used.
  - **DHCP security best practices**

    You need to make sure that you are not allowing unwelcome devices to infiltrate your network. There are a few things to do to prevent this. Here's a list of DHCP security best practices:
    - Keep your business networks and guest networks separate.
    - If you are using a managed switch, be sure to disable unused ports.
    - Generate alerts from your DHCP server when an unrecognized device sends a DHCP request.

IT601 – System and Network Administration

# Web Service

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Web Server and Web Services

A web service  makes an application (s) or content available to the users over the internet using internet browsers.


- Running a web server is more than just installing Apache or IIS and providing content.

## 1 - Open Standards

- The web is based on open standards, which are developed by an international committee, not a single corporation.
- You can use them without paying royalty or licensing fees.
- Web standards are defined by the World Wide Web Consortium (W3C), and the underlying Internet protocols are defined by the Internet Engineering Task Force (IETF).

## 2 - Benefits

- The benefit of web-based applications is that one browser can access many web applications. The web browser is a universal client.
- Cost , No transfer of software product, Universal availability

## 3 - Usage

- Web applications and small web servers are also present in firmware in many devices, such as small routers and switches, smart drive arrays, and network components.

# Types of Web Servers

There are four basic web server types:

## Static web server

1. Serves only documents that don't change or change only rarely.
2. The documents are read from disk and are not altered by the web server.

## CGI servers

1. Common Gateway Interface (CGI) is interface specification to enables servers to execute an external program, typically to process.
2. CGI servers generate pages dynamically.
3. Dynamic page generation uses more CPU and memory than reading from disk.

## Database-driven web sites

1. Generate each web page from database.
2. Each page is generated by reading information from a database and filling that information into a template.

## Multimedia servers

1. Content includes media files, such as video or audio.
2. Have high performance requirements for CPU, memory, storage, and network.

# Multiple Web Servers on One Host

## Small Vs Large sites

| Small Sites | Large Sites |
|---|---|
| 1. Small sites may want to start with a single web server that is used for all applications.<br>2. The amount of traffic that each application receives does not justify the cost and overhead of additional machines. | 1. Large sites that load-balance web services across multiple machines often want all the machines to be identical clones of each other, rather than having N machines dedicated to one application and M machines dedicated to another.<br>2. Identical clones are easier to manage.<br>3. There are many ways to host multiple web services on the same machine. |

## Multiple Web service Problems

**Single Port**

1. HTTP-based services usually listen for requests on TCP ports 80 (unencrypted) and 443 (encrypted via SSL, often called HTTPS).
2. Multiple services cannot share a port. That is, only one process can be listening to port 80 at a time.

**Multiple IP Addresses**

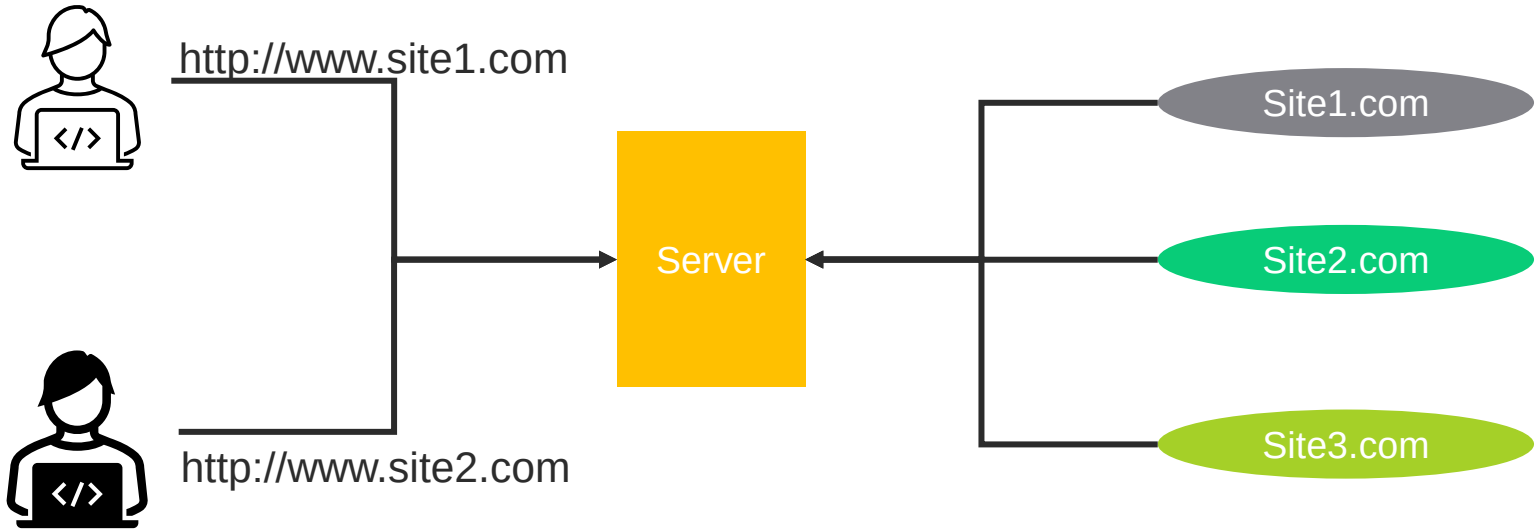
1. The easiest way to resolve this is to configure multiple IP addresses on a machine. A less attractive option is to use different ports for each server, with URLs that list the port, such as http://myservice:8000.
2. Configuring multiple IP addresses on a machine does not scale well. This scheme is complex and not well supported in some OSs. Also, IPv4 addresses are scarce. URLs containing the port information look unprofessional.

# Scalable Techniques for Multi Services

There are two techniques for permitting many services to share the same port on the same IP address. Both techniques provide a scalable solution with a professional look.

## 1 - Virtual hosting

- The first technique is called virtual hosting. Originally one machine could host only one domain.



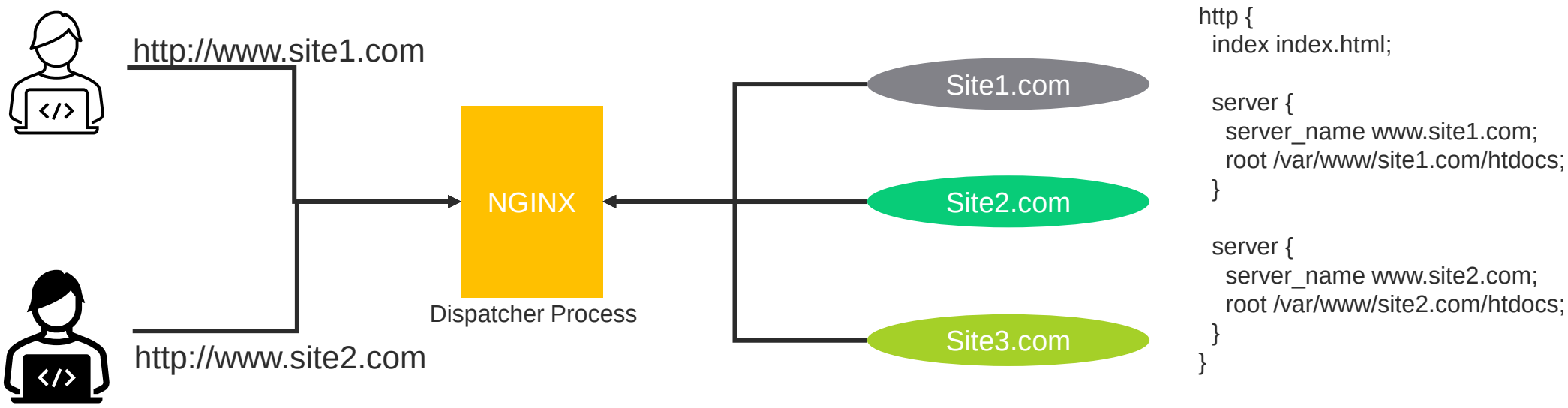

```
<VirtualHost *:80>
   ServerAdmin admin@site1.com
   ServerName site1.com
   ServerAlias www.site1.com
   DocumentRoot /var/www/ site1.com/html
</VirtualHost>

<VirtualHost *:80>
   ServerAdmin admin@site2.com
   ServerName site2.com
   ServerAlias www.site2.com
   DocumentRoot /var/www/site2.com/html
</VirtualHost>
```

- Virtual hosting permits one system to listen for incoming HTTP and HTTPS connections but serve different web pages depending on which domain the request is for.
- This required a change to the HTTP protocol, but it is supported by all web browsers.

# 2 - Dispatcher

- The second technique is to use one process as a dispatcher for many individual services. A single process listens for connections and forwards them to the proper service.




```
http {
  index index.html;

  server {
    server_name www.site1.com;
    root /var/www/site1.com/htdocs;
  }

  server {
    server_name www.site2.com;
    root /var/www/site2.com/htdocs;
  }
}
```

- Nginx  is often used to listen for incoming HTTP and HTTPS connections, determine which domain they are intended for, and forward the request as appropriate.
  - The individual services listen on different ports  such as 8000, 8001 etc , but this is hidden from the users.
  - This has another benefit in that Nginx can decrypt SSL connections so that the individual services must process only the simpler HTTP protocol.
  - It also isolates all cryptographic issues to one place, where they can be managed separately.

# HTTPS and Multiple Servers

## Catch-22 Problem

Serving multiple domains on the same IP address is more complex for HTTPS (encrypted) connections due to a Catch-22 in how HTTPS works. Each domain has its own SSL certificate. The software needs to know which SSL certificate to use to decrypt the request, but the intended domain is in the encrypted portion of the request.

## Techniques to Address Catch 22

### Share Certificate

1. An early solution to this Catch-22 was to permit one certificate to apply to multiple domains.
2. These multidomain certs are called subject alternative name (SAN) certificates.
3. All but the oldest, most broken web browsers support it. Since there is one certificate, the expiration data and other information are the same for all domains, which isn't usually a problem.
4. However, the certificate must be reissued to add and remove domains, which is inconvenient.

### Server name indication

1. A newer technique is called server name indication (SNI).
2. This allows for the use of multiple unique certificates on the same machine, as would be expected. In order for SNI to function, the client must first submit the intended domain in an unencrypted form. To decrypt the session, the server can choose which certificate to apply.
3. The desired domain must match the unencrypted request and is nevertheless provided in the encrypted part for security reasons. Sadly, SNI requires a modification to the SSL protocol, therefore it is incompatible with older hardware. These gadgets have, however, largely vanished.

# SLA for Web Services

- **A web service needs an SLA.**
- **Generally, web as a 24/7 critical service, but the SLA of an individual web service might be quite different.**
  - Most internal web services will have the same SLA as other office services, such as printing or storage.
- **Ideally, as with any SLA, the service level should be set by collaborating with the customer community.**

  1 - Resist setting any SLA that does not allow for periodic maintenance, unless the service is built out on redundant infrastructure.

  2 - If the service is provided by a single host or a shared web host and is required to be available around the clock, it is time to discuss increasing the redundancy of the service.

  3 - Metrics that are part of a web SLA should include the latency for a certain level of QPS.

  - For example , how long should a typical query take when the system is under a particular load? Latency is usually measured as the time between receipt of the first byte of the request and sending of the last byte of the answer.

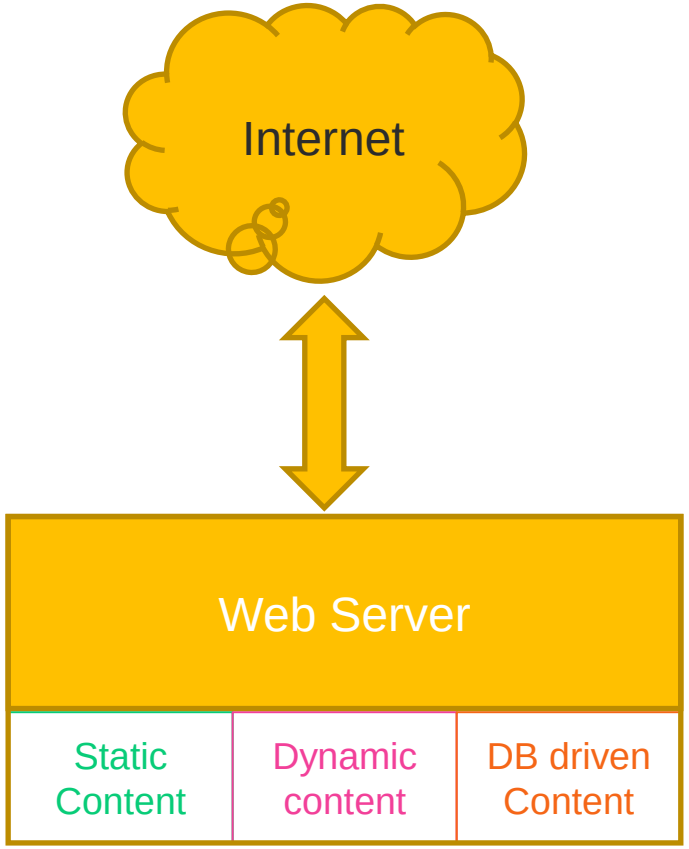
# Web Service Monitoring

- **Why monitoring web services?**
  - How well it is scaling?
  - Which areas need improvement?
  - Whether the service is meeting your SLA?
- **Be specific about which web-specific elements to your monitoring.**
  - Web server errors are most often related to problems with the site's content and are often valuable for the web development team.
  - Certain errors or patterns of repeating errors can be an indication of customer problems with the site's scripts. Other errors may indicate an intrusion attempt. Such scenarios are worth investigating further.
- **Typically, web servers allow logging of the browser client type and of the URL of the page containing the link followed to your site (the referring URL).**
- **Web servers may have server-specific information that would be useful as well, such as data on active threads and per-thread memory usage.**
- **Become familiar with any special support for extended monitoring available on your web server platform.**

# Scaling for Web Services

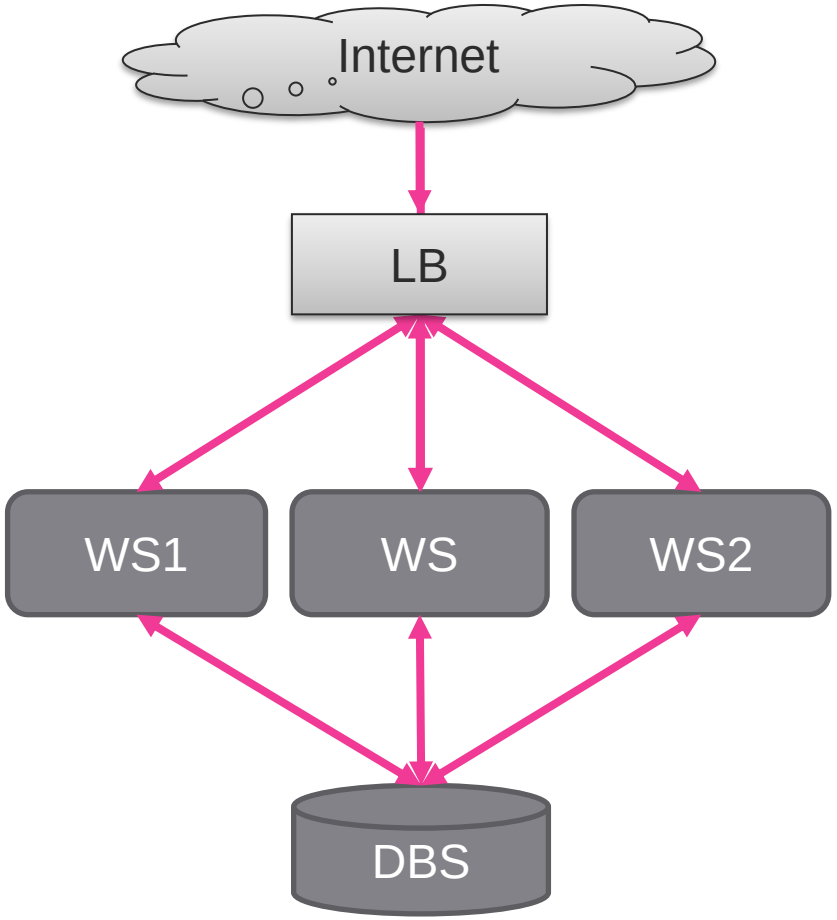
**“Scaling is the only problem on the Internet. Everything else is a subproblem.”**

- **slashdot effect**
  - The Slashdot effect, also known as slashdotting, occurs when a popular website links to a smaller website, causing a massive increase in traffic.
  - If the  web server is successful, it will get overloaded by requests. You may have heard of the Slashdot effect.
- **Single machine web server Scaling**
  - A small organization with basic needs can improve a web server’s performance by simply upgrading the CPU, disks, memory, and network connection.
- **Multi machine web server Scaling**
  - For multiple machines, there are two main types of scaling are horizontal and vertical. They get their names from web architecture diagrams.
    - When drawing a representation of the web service cluster, the machines added for horizontal scaling tend to be in the same row, or level.
    - For vertical scaling, they are in groups arranged vertically, as they follow a request flowing through different subsystems.



# Horizontal Scaling

- **In horizontal scaling, a web server or web service resource is replicated and the load is divided among the replicated resources.**
- **Example**
  - Consider two web servers with the same content, each getting approximately half the requests.
  - The typical solution is to use a device called a load balancer. A load balancer sits between the web browser and the servers.
  - The browser connects to the IP address of the load balancer, which forwards the request transparently to one of the replicated servers.
  - The load balancer tracks which servers are down and stops directing traffic to a host until it returns to service.
  - Other refinements, such as routing requests to the least-busy server, can be implemented as well
- **Load balancers are often general-purpose protocol and traffic shapers, routing not only HTTP but also other protocol requests, as required.**
  - This allows much more flexibility in creating a web services architecture. Almost anything can be load balanced, and this approach can be an excellent way to improve both performance and reliability.



# Vertical Scaling

- **Separates out the various kinds of subservices, rather than duplicating a whole machine.**
  - **It allows you to create an architecture with finer granularity, so that you can put more resources at the most intensively used stages of page creation.**
  - **It also keeps different types of requests from competing for resources on the same system.**
- **Examples**
  - Consider the single web server case , split the different types of content onto separate machines, and change the web pages to refer to those other machines for that content.
  - Using an application to complete a brief survey on a video clip after it has been viewed is one example of a website with numerous huge video clips. It is inefficient to try to write numerous minor database changes while reading huge video files from the same disc.
  - OS may have caching algorithms that are automatically tuned for one or the other but perform badly when both happen. In this case, all the video clips might be put on a separate web server, perhaps one with a storage array customized for retrieving large files. The rest of the web site would remain on the original server. Now that the large video clips are on a separate server, the original server can handle many more requests.
- **As you might guess, horizontal and vertical scaling can be combined. The video survey web site might need to add another video clip server before it would need to scale the survey form application.**

# How to choose a Scaling Method

- A site may need horizontal or vertical scaling or some combination of both.

## 1 - Classify the various components

- Classify the various components that are used in conjunction with your web server according to the resources they use most heavily.

# How to choose a Scaling Method

## 2 - Identify Interfering Components

- Then look at which components compete with one another and whether one component interferes with the function of other components.

  - A site may include static files, CGI programs, and a database.
  - Static files can range from comparatively small documents to large multimedia files.
  - CGI programs can be memory-intensive or CPU-intensive processes and can produce large amounts of output.
  - Databases usually require the lion’s share of system resources.
  - In some cases, such as the video survey site, you might choose to move part of the service to another server.

# How to choose a Scaling Method

## 3 - Use Diagnostics

- Use system diagnostics and logs to see which kinds of resources are being used by these components.

  - Consider an IS department web server that is also being used to create graphs of system logs.

  - This can be a very CPU-intensive process, so the graphing scripts and the log data can be moved to another machine, leaving the other scripts and data in place.

# How to choose a Scaling Method

4 - Decide All or Gradual Upgrades

# Web Service Security

## Importance of Web server security

- Implementing security measures is a vital part of providing web services.
- Security is a problem because people you don't know are accessing your server.
- Some sites believe that security is not an issue for them, since they do not have confidential documents or access to financial information or similar sensitive data.
- However, the use of the web server itself and the bandwidth it can access are, in fact, a valuable commodity.

## The Uses of Web Server Attacks

- Intruders often break into hosts to use them for money-making purposes, and sometimes they do so simply for entertainment
- Intruders usually do not deface or alter a web site, since doing so would lead to their discovery. Instead, they simply use the site's resources.
- **Common uses of hijacked sites and bandwidth**
  - Distribution of pirated software
  - Generating advertising email
  - Launching automated systems to try to compromise other systems
  - Competing with other intruders to see who can run the largest farm of machines

# Why to Secure Internal Websites?

**Internal web services should be secured as well. Although you may trust employees of your organization, there are still several reasons to practice good web security internally:**

- Stopping virus transmission
- Protecting privileged information that requires authentication.
- Protect from visitors—temps, contractors, vendors, speakers, interviewees
- Protect Network resources such as WIFI
- Security and reliability go hand-in-hand such avoiding DOS

# Secure Connections and Certificates

- **Usually web sites are accessed using unencrypted, plaintext communication.**
  - The privacy and authenticity of the transmission can be protected by using HTTP over Secure Sockets Layer (SSL) to encrypt the web traffic. SSL 4.0 is also known as Transport Layer Security (TLS) 1.0. Confusingly, SSL 2.0 and 3.0 predate TLS 1.0.
  - Use encryption to prevent casual eavesdropping on our customers' web sessions even if they are connecting via a wireless network in a public place, such as a coffeeshop. URLs using https:// instead of http:// are using SSL encryption.
- **Implementing HTTPS on a web server is relatively simple, depending on the web server software being deployed. Properly managing the cryptographic certificates is not so easy.**
  - The private key of certificate must be kept secret. If it is leaked to outsiders, they can use it to pretend to be your site.
  - One role of the web system administrator is to maintain a repository, or key escrow, of certificates for disaster-recovery purposes. Treat this data the same way as you manage other important secrets, such as root or administrator passwords.
- **Places to store private keys**

  Private keys must be secret, but the web server needs to read the private key to use it. There are a number of ways to solve this issue:

| Isolated storage | Encrypted storage | Network-accessible key repository | Protected file |
|---|---|---|---|

# Self-Signed-Certificates

- **A cryptographic certificate is created by the web system administrator using software that comes with the encryption package; OpenSSL is one popular system.**
- **The certificate at that point is "self-signed," which means that when someone connects to the web server using HTTPS, the communication will be encrypted, but the client that connects has no way to know that it has connected to the right machine.**
  - Anyone can generate a certificate for any domain. It is easy to trick a web client into connecting to a "man in the middle" attacker instead of to the real server; the client won't know the difference. This is why most web browsers, when connecting to such a web site, display a warning stating that a self-signed certificate is in use.
    - The solution to this dilemma is to use an externally signed cryptographic certificate from a registered certification authority (CA).
    - The public half of the self-signed certificate is encrypted and sent to a trusted CA, which signs it and returns the signed certificate.
    - The certificate now contains information that clients can use to verify that the certificate has been certified by a higher authority.
    - When it connects to the web site, a client reads the signed certificate and knows that the site's certificate can be trusted because the CA says that it can be trusted.

# Protecting the Web Server Application

- **A variety of malicious efforts can be directed against the web server itself in an attempt to get login access to the machine or administrative access to the service.**

- **Any vulnerabilities present in the operating system can be addressed by standard security methods.**

- **Web-specific vulnerabilities can be in multiple layers of the web server implementation such as**

  - The HTTP server
  - modules or plug-ins that extend the server
  - web development frameworks running as programs on the server.

  This last category is distinct from generic applications on the server, as the web development framework is serving as a system software layer for the web server.

- **The best way to stay up-to-date on web server security at those layers is through vendor support.**

  - The various HTTP servers, modules, and web development environments often have active mailing lists or discussion groups and almost always have an announcements-only list for broadcasting security exploits, as well as available upgrades.

# Protecting the Content

- **Some web-intrusion attempts are directed toward gaining access to the content or service rather than to the server..**
  - There are many types of web content security exploits, and new ones are always being invented.
- **SAs should educate themselves on current exploits via Internet security resources.**
  - The knowledge allows properly evaluate a server for complex threats is a significant undertaking.
  - Fortunately, open source and commercial packages to assist you are available.

## Software Vulnerabilities

- largest threat is security vulnerabilities frameworks used.
- The code may be perfect, but an old framework may have security vulnerabilities.
- Most of security incidents are not due to the newest security threats, but old unfixed vulnerabilities.

## Directory Traversal

- A technique generally used to obtain data.
- Data may be useful itself or used to enable method of direct intrusion.
- Efficient with servers generating auto indexes.
- Modern servers don't allow to traverse root directory, but old servers are vulnerable.
- common variation of directory traversal uses a CGI query

## Form-Field Corruption

- DB fields may appear in form or contents.
- A legitimate web form can be copied and altered to gain access to data or services.
- form validations strictly may prevent it but newer methods are always coming.

## SQL Injection

- A variant of form-field corruption is SQL injection.
- Intruders can even perform entire SQL queries, updates, and deletions.
- Some database systems
- include debugging options that permit running arbitrary commands on the
- operating system.

# Application Security

- The efforts of malicious people can be rendered less likely to succeed by following good security practices when developing applications.
- The following are some of the fundamental practices to use when writing web code or extending server capabilities.

## Limit the Potential Damage

- the best protections is to limit the amount of damage an intruder can do.
- Store Contents and programs on well protected servers and copy to web server on need.
- If web server is defaced by intruder, it can quickly be reimaged.
- Another technique is to isolate the webserver to its own network, disallow connections to internal hosts.
- Backups, logging and installations must be from internal hosts

## Validate Input

- Validate the input to interactive web apps so as to maximize security.
- Input should be checked for length, to prevent buffer overflows where executable commands could be deposited into memory.
- Disallow using quotes or escape characters.
- It is better to validate input by inclusion than by exclusion.
- adopt programming paradigms that do not reinterpret or reparse data for you

## Automate Data Access

- Access to DBs should be specific.
- If a web application needs to read data, allow with read only access.
- If your DB supports stored precompiled queries, use them instead of executing SQL input.
- preparation function of DB should be used to convert potentially executable input into a form.

## Use Permissions and Privileges

- Set permissions to local sever for authentication methods.
- Apply the least-privilege security principle to web servers and web applications

## Use Logging

- Logging is an important protection of last resort.
- After an intrusion attempt, detailed logs will permit more complete diagnostics and recovery.
- Logs should be stored on other machines or in nonstandard places to make them difficult to tamper with.
- Another way of storing logs in a nonstandard place is to use network logging

# Content Management

- **Providing a content-management system (CMS) empowers users to perform self-service updates of the web content.**
  - A CMS lets you create privileged accounts for those users who are permitted to perform these updates, often in some kind of controlled manner.

- **It is not a good idea for an SA to be directly involved with content updates. Assuming this responsibility not only adds to the usually lengthy to-do list of the SA, but also creates a bottleneck between the creators of the content and the publishing process.**
  - Form a web council makes to attach domains of responsibility for web site content much easier, because the primary "voices" from each group are already working with the webmaster or the SA who is being a temporary webmaster.
  - The web council is the natural owner of the change control process.

- **This process should have a specific policy on updates. Ideally, the policy should distinguish three types of alterations that might have different processes associated with them:**
    - Update: Adding new material or replacing one version of a document with a newer one
    - Change: Altering the structure of the site, such as adding a new directory or redirecting links
    - Fix: Correcting document contents or site behavior that does not meet the standards

# Web Server Packages

- There are several web server package available for Linux and windows. The commonly used packages are…

## Apache HTTP

- second most popular web server software.
- It's an open-source project that uses HTTP protocol.
- operates across various OSs, including Windows and Linux.
- Best features of Apache is its customizability.
- comprised of several modules.
- Key Features for Apache: IPv6, Session tracking, FTP and HTTP/2,Customizable modules

## NGINX

- Most highly used
- NGINX is compatible with both Linux and Windows.
- most famous for its high-performance features.
- designed to handle multiple connections simultaneously.
- NGINX is less customizable. You can't disable some of its modules,

## Microsoft IIS

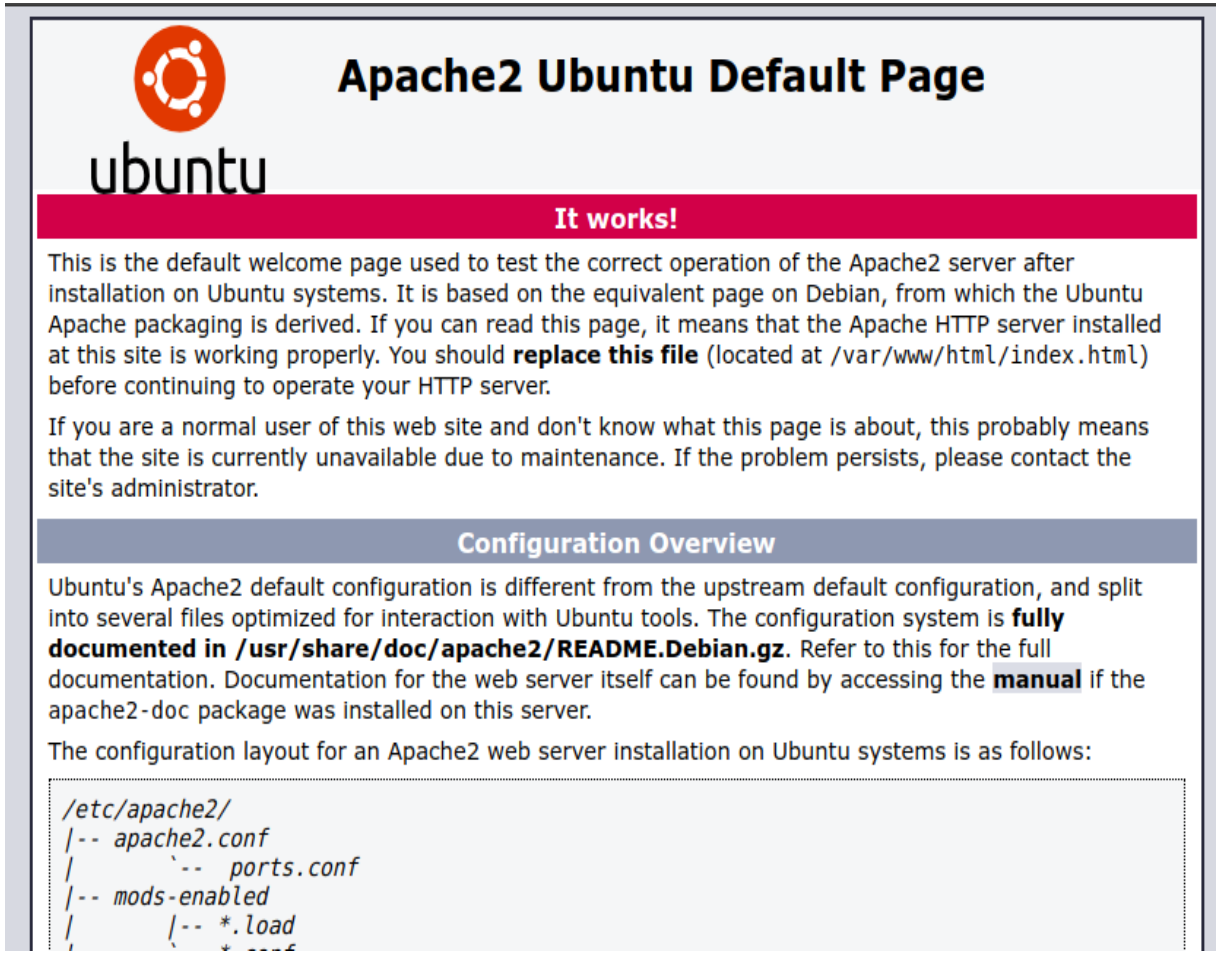
- an excellent server software option that's specifically designed for Windows.
- includes many native Windows security features, such as Azure Active Directory
- it has integrated website and server management tools.
- native support for dynamic ASP.NET applications, spanning CSS, JavaScript, and HTML

## Apache Tomcat

- Best web server software options for Java applications.
- uses multiple Java specifications in an open-source environment.
- Tomcat comes from the same company as Apache.
- Key Features for Tomcat: Customizable modules, Multiple Java technologies, including Jakarta WebSocket, Performance-enhanced data processing, Open-source design

# Installing Apache Package

- **Install apache2 Package**
  - sudo apt update
  - sudo apt install apache2
- **By default, Apache comes with a basic site**
  - Web Content : /var/www/html
  - Configurations :
    - /etc/apache/apache2.conf
    - /etc/apache/ ports.conf
    - /etc/apache/mods-enabled
    - /mods-available
    - /sites-enabled
    - /sites-available
    - /conf-available
- **Operational Commands**
  - systemctl <*start*|*restart*|*reload*|*stop*> apache
  - a2ensite <config-name>
  - a2dissite <config-name>
  - a2dismod <module-name>
  - a2enmod <module-name>
  - apachectl <options>



# Apache Modules

- **Active Apache modules**

  apache2ctl -t -D DUMP_MODULES

  apache2ctl -M

- **Commonly used Apache modules**
  - Mod_security
  - Mod_rewrite
  - Mod_deflate
  - Mod_cache
  - Mod_proxy
  - Mod_ssl

# Mod_security

- Deals with the security of your server.

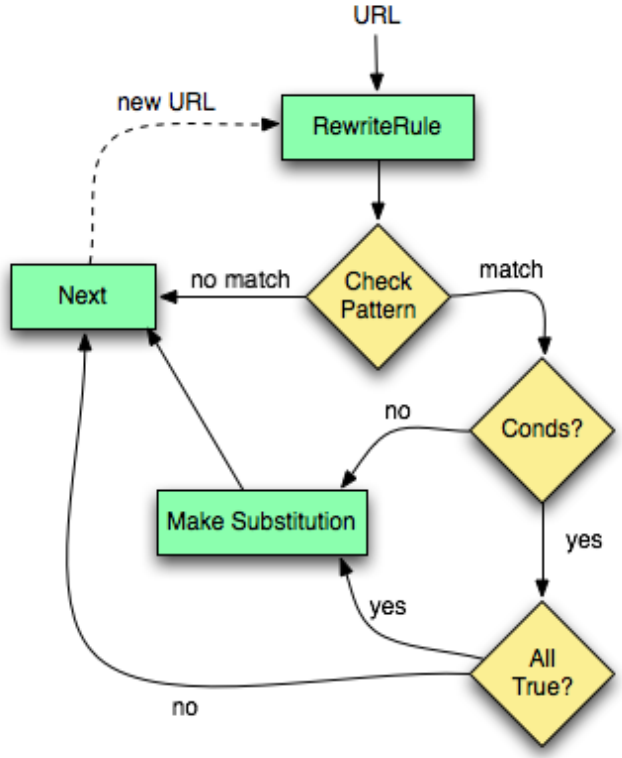
  - It can protect the server from various attacks.
  - It uses the regular expressions and rule sets to block the attacks.
  - It works as a firewall.
  - It could work either embedded or as a reverse proxy.

- A reverse proxy is a proxy server that accesses the servers on behalf of a client. They retrieve the resources from the servers and return to the client as they from the proxy server and not from the original server.

- It is very efficient in blocking SQL injection attacks. When a SQL injection attack is completed, it will return a 406 error.

# mod_rewrite

- The mod_rewrite is also popular in the web hosting industry.
  - It is used to rewrite the URLs and so that the redirection can be achieved.
  - The module has a rewrite engine which will rewrite a requested URL based on a PCRE regular expression parser.




- The mod_rewrite uses unlimited rules. Each rule can have unlimited attached rule conditions which enables the rewriting based on server and environment variables, HTTP headers, etc.

- example of a rule for redirecting a url that starts with ‘http’ to ‘https’ is given below.

```
RewriteEngine On

RewriteCond %{HTTPS} off

RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

- https://httpd.apache.org/docs/2.4/mod/mod_rewrite.html

# mod_deflate

- This module is used to compress the output from the webserver before sending to the client. It reduces the size of the output file, so that the client can download it faster.
  - The following directive will enable the compression for documents in the container where it is placed.

```
SetOutputFilter DEFLATE
```

  - If Apache is being used as a reverse proxy and you need to process the content which passes through the proxy, you can use the mod_deflate for decompressing purpose as well.
  - However, it is rarely used and the common use of mod_deflate is to compress the web server output. The mod_deflate uses a combination of LZ77 algorithm and Huffman coding.
  - This ensures no data is lost while compressing the files. If the output file size is less than 120 bytes (approximately), then the output file will be larger after being processed by the mod_deflate.
  - This happens because this module does not have a lower bound for the file size. The mod_gzip is similar to mod_deflate.

# mod_cache

- The mod_cache is the Apache module that is used for content caching.

  - Web caching is a way to improve the performance of the server.
  - The commonly requested content will be stored in an easy to access locations so that the client can access the data faster and client does not need to retrieve the data every time when the request is demanded.
  - We can create caching rules for making caching effective. However, highly dynamic content will be served to the client normally from the server.
  - There are many other methods also used by Apache for the purpose of web caching and one of them is using mod_file_cache module.

# mod_proxy

- **This is an optional Apache module.**
  - This module implements a proxy, gateway for Apache server.
  - It supports many commonly used protocols and many load balancing algorithms.

- **To enable this feature, a set of modules will need to be loaded onto the server. This include the following.**

1) mod_proxy
2) mod_proxy_balancer
3) One or more proxy scheme, or protocol, module

# mod_ssl

- Mod_ssl also is an optional module of the Apache.
- It is used in Apache version 1.3 and version 2.
- It enables encryption via the Secured Sockets Layer (SSL) and Transport Layer Security (TLS) with the help of Open-Source SSL/TLS toolkit OpenSSL.
- Its original version was created for Apache version 3 in 1998.
- The intention of this module is to provide SSL v3 and TLS v1.x support for the Apache server. The SSL v2 is no longer supported.

# .htaccess

- **The .htaccess file in Apache is a tool that allows configurations at the directory and subdirectory level.**

- **Using .htaccess enables you to configure website permissions without altering server configuration files.**

- **Enabling .htaccess**
  - sudo vi /var/www/my_website.com/.htaccess
  - sudo vi /user/safe_location/.htpasswd

- **Common Uses**
  - Manage IP Addresses
  - Block Visitors by Referrer
  - Redirect Traffic
  - Set a 404 Page

# .htaccess Usage

## ➢ Enable authentication:

```
AuthUserFile /user/safe_location/.htpasswd
AuthGroupFile /dev/null
AuthName "Please Enter Password"
AuthType Basic
Require valid-user
```

## ➢ Redirect Traffic

```
Redirect301/Other_Website.com/index.html/My_Website.com/index.html
```

## ➢ Allow. Deny IP Addresses

```
order deny, allow
deny from 192.168.0.54
allow from 192.168.0
```

## ➢ Block Visitors by Referrer

```
RewriteEngine on
# Options +FollowSymlinks
RewriteCond %{HTTP_REFERER} blockeddomain\.com [NC]
RewriteRule .* - [F]
```

## ➢ Set a 404 Page

```
ErrorDocument 404 /404.html
```

# Minimal Basic Configuration

- **Listen:**
  - Used to bind Apache to specific IP addresses and/or ports. HTTP server, by default, runs on port 80 for production.
  - For testing, you could choose a port number between 1024 to 65535, which is not used by an existing application (you can run command "netstat" to check the existing connections). We shall run the Apache at port 8000.

```
# Listen: Allows you to bind Apache to specific IP addresses and/or ports.

Listen 8000
```

- **ServerName:**
  - Set to your DNS hostname, or IP address (to find out your IP address, run command "ipconfig"), or your computer name, or "localhost" followed by the port number chosen above.

```
# ServerName gives the name and port that the server uses to identify itself.
# If your host doesn't have a registered DNS name, enter its IP address here.

ServerName YourHostNameOrIPAddres:8000
```

# Minimal Basic Configuration

- **ServerRoot:**
  - The Apache installed directory "<APACHE_HOME>", e.g.,

```
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
# Assume that Apache HTTP server is installed in “/Project/apache2“


ServerRoot “/Project/apache2"
```

Note : use Unix-style forward slash (/) as the directory separator, instead of Windows-style backward slash (\) in the configuration file.

# Configuration options

- **DocumentRoot:**
  - It specifies document root directory or home directory of the server. It is set to <APACHE_HOME>\htdocs by default.

```
# DocumentRoot: The directory out of which you will serve your documents.
DocumentRoot “/Porject/apache2/htdocs"

# Access Control for the document base directory
<Directory "/Porject/apache2/htdocs">
    # Show directory listing, and allow symbolic links
    Options Indexes FollowSymLinks

    # Cannot override with .htaccess files.
    AllowOverride None

    # Controls who can get stuff from this server.
    Order allow,deny
    Allow from all
</Directory>
```

Caution: You MUST do a global search on "htdocs", before modifying the document root directory.

IT601 – System and Network Administration

# DNS Service

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# DNS Service

- **Domain Name Service (DNS) is an Internet service that maps IP addresses and fully qualified domain names (FQDN) to one another.**
- **DNS service alleviates the need to remember IP addresses. Computers that run DNS are called name servers.**
- **Ubuntu is shipped with BIND (Berkley Internet Naming Daemon), the most common program used for maintaining a name server on Linux.**
- **Fully Qualified Domain Name (FQDN)**

# DNS Terminology and Components

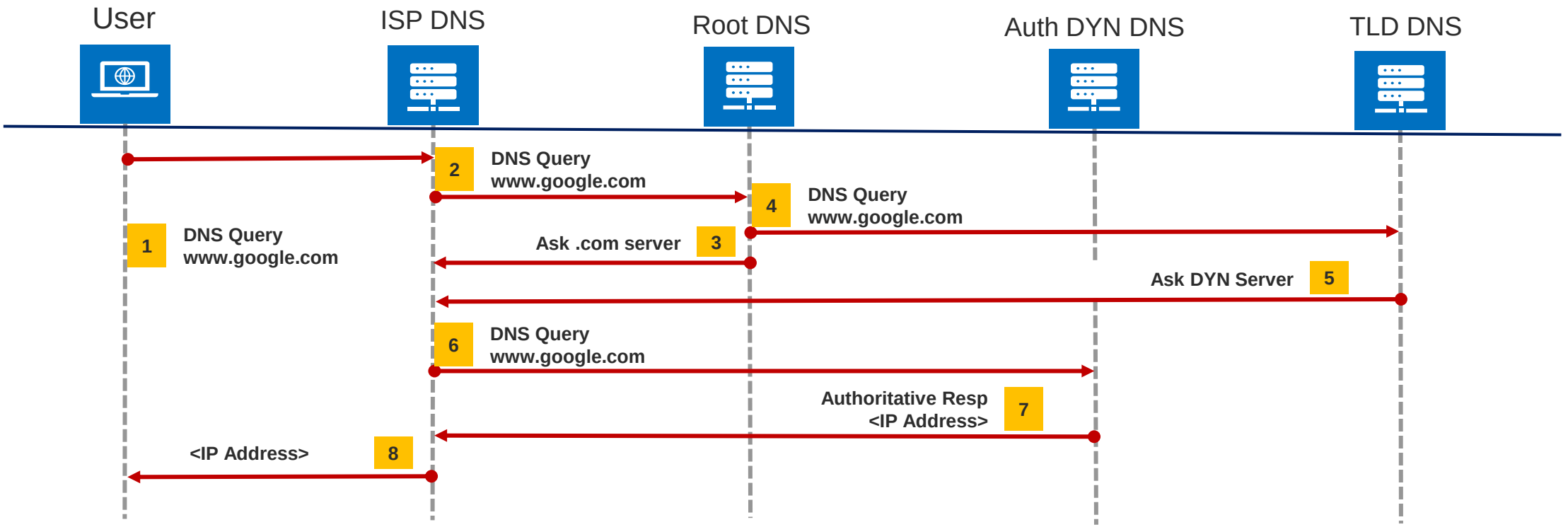
- **DNS Terminology**
  - Domain Name System
  - Domain Name
  - IP Address
  - Top-Level Domain
  - Hosts
  - Subdomain
  - Fully Qualified Domain Name
  - Name Server
  - Zone File
  - Records

**DNS Components**

- Root Servers
- TLD Servers
- Domain-Level Name Servers
- Resolving Name Server
- Zone Files

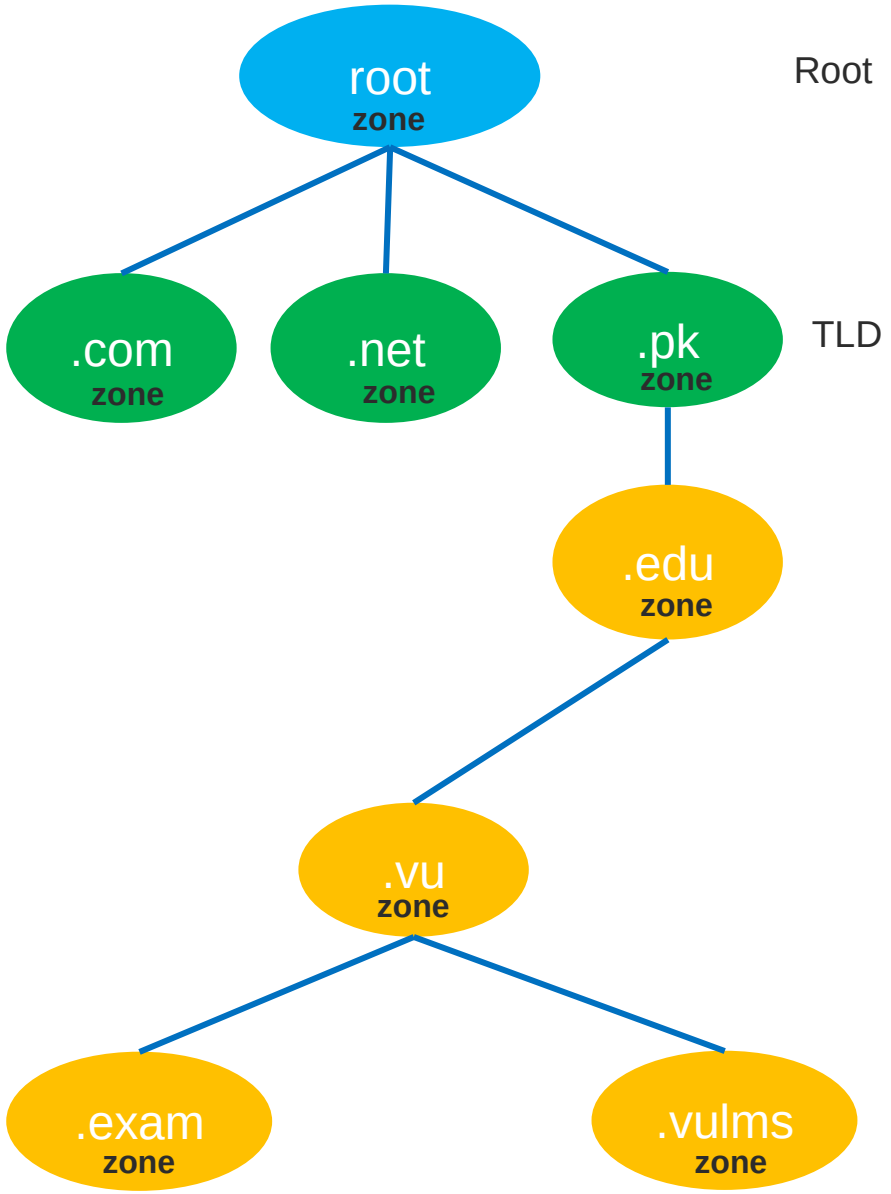
# DNS Resolution

- DNS resolution is completed in several steps.



# DNS ZONE

- The DNS is broken up into many different zones. These zones differentiate between distinctly managed areas in the DNS namespace.
- A DNS zone is a portion of the DNS namespace that is managed by a specific organization or administrator. A DNS zone is an administrative space which allows for more granular control of DNS components, such as authoritative nameservers.
- The domain name space is a hierarchical tree, with the DNS root domain at the top. A DNS zone starts at a domain within the tree and can also extend down into subdomains so that multiple subdomains can be managed by one entity.
- Don't associate a DNS zone with a domain name or a single DNS server. In fact, a DNS zone can contain multiple subdomains and multiple zones can exist on the same server. DNS zones are not necessarily physically separated from one another, zones are strictly used for delegating control.
- All of the information for a zone is stored in what's called a DNS zone file, which is the key to understanding how a DNS zone operates



# DNS Record Types

## 1 SOA

- Admin Email
- Last Update
- Refresh Rate

## 2 A and AAAA

- A is an address record.
- Shows the IP address of hostname or domain
- AAAA record, just like A record, point to the IPv6 address for a domain

## 3 CNAME

- CNAME or "canonical name“ is a record that points a domain name (an alias) to another domain.
- The subdomain ng.example.com can point to example.com using CNAME. Here example.com points to the actual IP address using an A record.

## 4 MX

- A mail exchange (MX) record, is a DNS record type that shows where emails for a domain should be routed to.
- An MX record makes it possible to direct emails to a mail server.

## 5 NS

- A nameserver (NS) record specifies the authoritative DNS server for a domain.
- The NS record helps point to where internet applications like a web browser can find the IP address for a domain name.

## 6 PTR

- A pointer (PTR) record provides a domain name for reverse lookup.
- It's the opposite of an A record as it provides the domain name linked to an IP address instead of the IP address for a domain.

## 7 CAA

- The CAA record is a type of DNS record used to provide additional confirmation for the Certification Authority (CA) when validating an SSL certificate.

## 8 TXT

- TXT stands for "text," and this record type lets the owner of a domain store text values in the DNS.
- Several services use this record to verify ownership of a domain

## 9 SRV

- Using this DNS record type, it's possible to store the IP address and port for specific services

## 10 CERT

- This record type stores public keys certificates.

## 11 DCHID

- This DNS record type stores information related to dynamic host configuration protocol (DHCP).

## 12 DNAME

- DNAME is "delegation name.“
- It works very similarly to CNAME.
- It points all the subdomains for the alias to the canonical domain name.

# SOA Records

- **The Start of Authority (SOA) record is a mandatory record in all zone files.**
- **It must be the first real record in a file (although $ORIGIN or $TTL specifications may appear above).**
- **It is also one of the most complex to understand.**
- **Example**

```
domain.com.     IN      SOA     ns1.domain.com.     admin.domain.com. (
                        12083   ; serial number
                        3h      ; refresh interval
                        30m     ; retry interval
                        3w      ; expiry period
                        1h      ; negative TTL
                        )
```

# A and AAAA Records

- **Both of these records map a host to an IP address.**
- **The “A” record is used to map a host to an IPv4 IP address, while “AAAA” records are used to map a host to an IPv6 address.**
- **The general format of these records is this:**

| | | | |
|---|---|---|---|
| host | IN | A | IPv4_address |
| host | IN | AAAA | IPv6_address |
| ns1 | IN | A | 111.222.111.222 |

| | | | |
|---|---|---|---|
| ns1.domain.com. | IN | A | 111.222.111.222 |
| www | IN | A | 222.222.222.222 |
| domain.com. | IN | A | 222.222.222.222 |
| @ | IN | A | 222.222.222.222 |
| * | IN | A | 222.222.222.222 |

# CNAME Records

- CNAME records define an alias for canonical name for your server (one defined by an A or AAAA record).
- For instance, we could have an A name record defining the “server1” host and then use the “www” as an alias for this host:

```
server1    IN  A      111.111.111.111
www        IN  CNAME  server1
```

- Be aware that these aliases come with some performance losses because they require an additional query to the server.
- Most of the time, the same result could be achieved by using additional A or AAAA records.
- One case when a CNAME is recommended is to provide an alias for a resource outside of the current zone.

# MX Records

- **MX records are used to define the mail exchanges that are used for the domain. This helps email messages arrive at your mail server correctly.**
- **Unlike many other record types, mail records generally don't map a host to something, because they apply to the entire zone. As such, they usually look like this:**

```
IN  MX  10   mail.domain.com.
```

- **Example**

The MX record should generally point to a host defined by an A or AAAA record, and not one defined by a CNAME. So, let's say that we have two mail servers. There would have to be records that look something like this:

```
IN      MX    10    mail1.domain.com.
IN      MX    50    mail2.domain.com.
mail1   IN    A     111.111.111.111
mail2   IN    A     222.222.222.222
```

```
IN      MX    10      mail1
IN      MX    50      mail2
mail1   IN    A       111.111.111.111
mail2   IN    A       222.222.222.222
```

# NS Records

- **NS record type defines the name servers that are used for this zone.**

  **Question :** if the zone file resides on the name server, why does it need to reference itself?". Part of what makes DNS so successful is its multiple levels of caching.

  - One reason for defining name servers within the zone file is that the zone file may be actually being served from a cached copy on another name server.

- **Like the MX records, these are zone-wide parameters, so they do not take hosts either. In general, they look like this:**

```
IN  NS    ns1.domain.com.
IN  NS    ns2.domain.com.
```

- You should have at least two name servers defined in each zone file in order to operate correctly if there is a problem with one server.
- Most DNS server software considers a zone file to be invalid if there is only a single name server.
- As always, include the mapping for the hosts with A or AAAA records:

```
IN  NS    ns1.domain.com.
IN  NS    ns2.domain.com.
ns1    IN  A    111.222.111.111
ns2    IN  A    123.211.111.233
```

- **There are quite a few other record types you can use, but these are probably the most common types that you will come across.**

# PTR Records

- **The PTR records are used define a name associated with an IP address.**
- **PTR records are the inverse of an A or AAAA record.**
- **PTR records are unique in that they begin at the .arpa root and are delegated to the owners of the IP addresses.**
- **The Regional Internet Registries (RIRs) manage the IP address delegation to organization and service providers.**
- **The Regional Internet Registries include APNIC, ARIN, RIPE NCC, LACNIC, and AFRINIC.**
- **Example of a PTR record for 111.222.333.444 would look like:**

```
444.333.222.111.in-addr.arpa.    33692    IN    PTR    host.example.com.
```

- **Example of a PTR record for an IPv6 address shows the nibble format of the reverse of Google's IPv6 DNS Server 2001:4860:4860::8888.**

```
8.8.8.8.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.6.8.4.0.6.8.4.1.0.0.2.ip6.arpa. 86400IN PTR google-public-dns-a.google.com.
```

# PTR Records

- **The command line tool dig with the -x flag can be used to look up the reverse DNS name of an IP address.**

- **An example of a dig command. The +short is appended to reduce the output to the reverse DNS name.**

  dig -x 8.8.4.4 +short

  - The output for the dig command above will be the domain name in the PTR record for the IP address: google-public-dns-b.google.com.

- **Servers on the Internet use PTR records to place domain names within log entries, make informed spam handling decisions, and display easy-to-read details about other devices.**

- **Most commonly-used email servers will look up the PTR record of an IP address it receives email from.**

- **If the source IP address does not have a PTR record associated with it, the emails being sent may be treated as spam and rejected.**

- **It is not important that the FQDN in the PTR matches the domain name of the email being sent. What is important is that there is a valid PTR record with a corresponding and matching forward A record.**

# CAA Records

- **CAA records are used to specify which Certificate Authorities (CAs) are allowed to issue SSL/TLS certificates for your domain.**
  - As of September 8, 2017 all CAs are required to check for these records before issuing a certificate. If no record is present, any CA may issue a certificate.
  - Otherwise, only the specified CAs may issue certificates. CAA records can be applied to single hosts, or entire domains.
- **An example CAA record**

```
example.com.        IN        CAA        0 issue "letsencrypt.org"
```

  - The host, IN, and record type (CAA) are common DNS fields. The CAA-specific information above is the 0 issue "letsencrypt.org" portion. It is made up of three parts: flags (0), tags (issue), and values ("letsencrypt.org").
  - Flags are an integer which indicates how a CA should handle tags it doesn't understand. If the flag is 0, the record will be ignored. If 1, the CA must refuse to issue the certificate.
  - Tags are strings that denote the purpose of a CAA record. Currently they can be issue to authorize a CA to create certificates for a specific hostname, issuewild to authorize wildcard certificates, or iodef to define a URL where CAs can report policy violations.
  - Values are a string associated with the record's tag. For issue and issuewild this will typically be the domain of the CA you're granting the permission to. For iodef this may be the URL of a contact form, or a mailto: link for email feedback.
- **You may use dig to fetch CAA records using the following options:**

```
dig example.com type257
```

IT601 – System and Network Administration

# IAM and Directory Services

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Identity and Access Management

- An IAM framework often includes a variety of solutions, tools, processes, policies, and technologies.
  - to ensure the right individuals have the right access to organization assets
  - to help security teams manage and monitor the user lifecycle
  - to protect organization assets from both internal and external threats.

# Components of an IAM

- The components of an IAM framework are based on the following principles:

## Identification or Authentication

- Confirming or denying the identity of the user attempting to access an asset.
- Single sign on (SSO) is a form of authentication.

## Authorization

- Controlling what a user is able to do once they are operating within an enterprise asset.
- Role-based access controls (RBAC) are an example of an authorization approach.

## Administration and Management

- Provisioning and managing throughout the user account lifecycle.
- Includes setup to deactivation, administration and management of requirements related to compliance and regulation and access to different computing environments and architectures.

## Monitoring and Auditing

- Observing, tracking, managing, and reporting on a user's activities.
- The types of data and metrics that are often monitored or audited include password resets, uncorrelated accounts, number of accounts and associated roles and entitlements across applications and systems, login failures, uncorrelated privileged accounts, separation-of-duty violations, non-human identities and associated access.

## Security and Protection

- Protecting enterprise assets (corporate devices, systems, data, networks, or software applications) from threats, such as breaches and damage due to unauthorized access by external threat actors, as well as insiders, such as disgruntled employees.

# Identity and Access Management

- Lightweight Directory Access Protocol (LDAP) Service
- Kerberos Authentication Service
- Radius Authentication Service
- TACACS Authentication Service

IT601 – System and Network Administration

# Lightweight Directory Access Protocol (LDAP) Service

Arif Husen

Department of Computer Science and Information Technology,

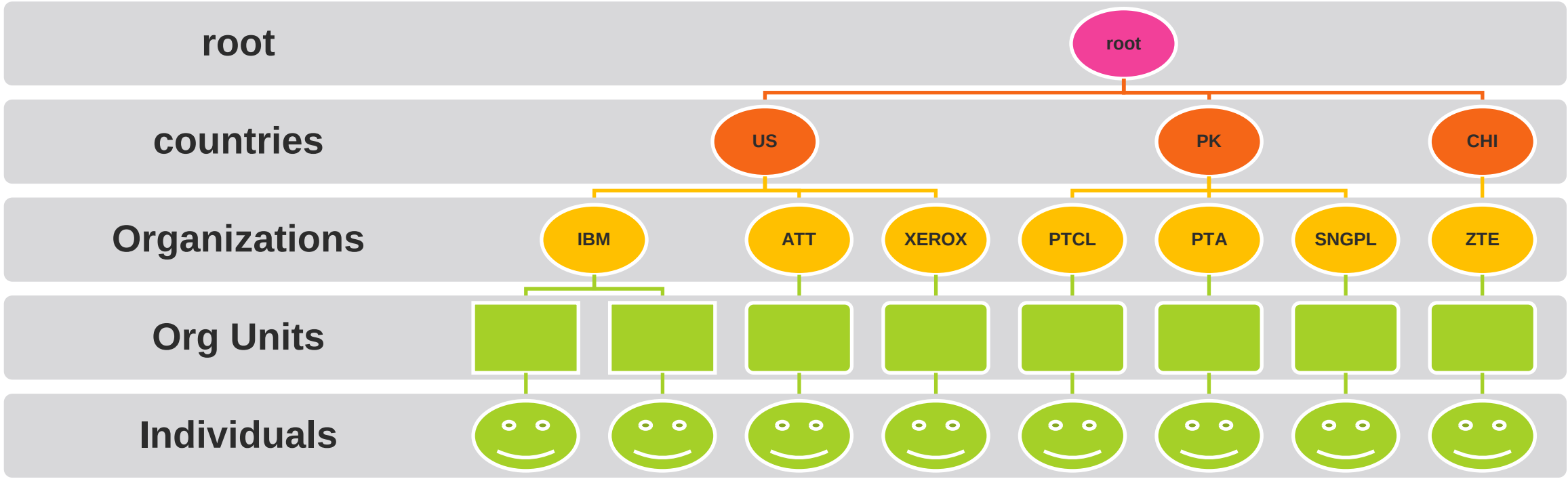
Virtual University of Pakistan

# Background and Motivation

- Increased reliance on networked computers
- Need in information
  - Functionality
    - To store data in the LDAP directory and authenticate users to access the directory.
    - Provide the communication language that applications require to send and receive information from directory services
  - Ease-of-Use
    - A directory tells the user where in the network something is located.
    - provide a central place for authentication, meaning it stores usernames and passwords.
    - LDAP can then be used in different applications or services to validate users with a plugin.
  - Administration (Application specific dirs)
  - Clear and consistent organization
  - Integrity
  - Confidentiality

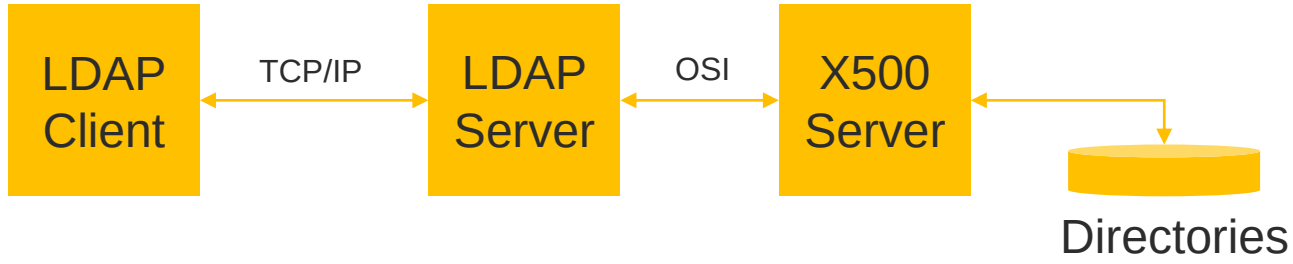
# X.500 Standard

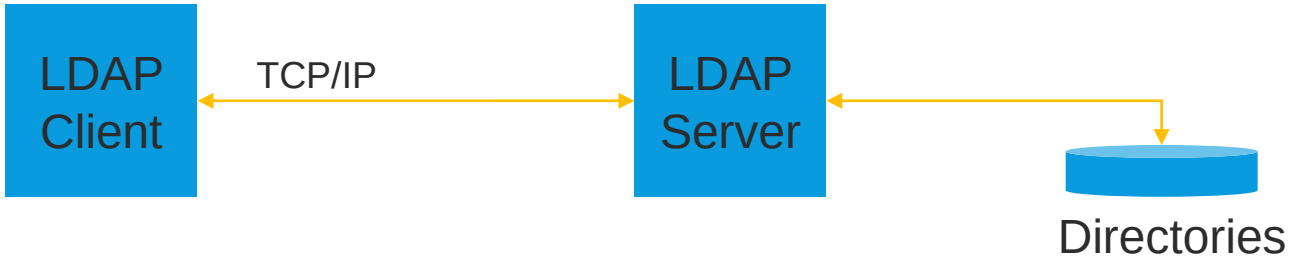
- Organizes directory entries into a hierarchical namespace
- Powerful search capabilities
- Often used for interfacing incompatible directory services
- Used DAP for c/s communication
- DAP (App. Layer) requires ENTIRE OSI stack to operate
- Too heavy for small environments



# What is LDAP?

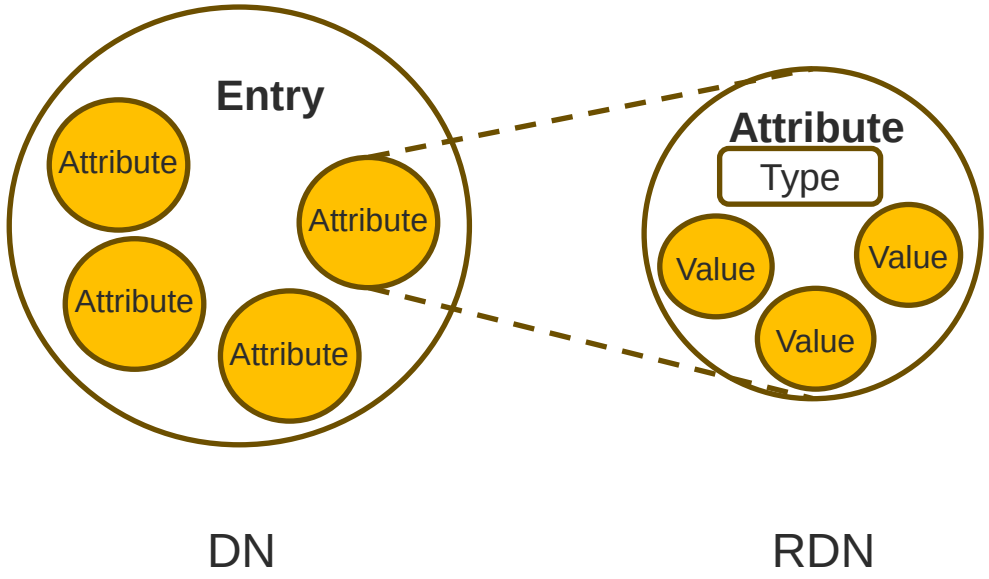
- Lightweight Directory Access Protocol
- Used to access and update information in a directory built on the X.500 model
- Specification defines the content of messages between the client and the server
- Includes operations to establish and disconnect a session from the server
- Lightweight alternative to DAP
- Uses TCP/IP instead of OSI stack
- Simplifies certain functions and omits others…
- Uses strings rather than DAP's ASN.1 (Abstract Syntax Notation One) notation to represent data.



# LDAP Conventions

- **Information**
  - How information is stored in ldap directory
- **Naming**
  - How information is organized and identified.
- **Functional / Operations**
  - Describes what operations can be performed on the information stored in an LDAP directory.
- **Security**
  - Describes how the information can be protected from unauthorized access.



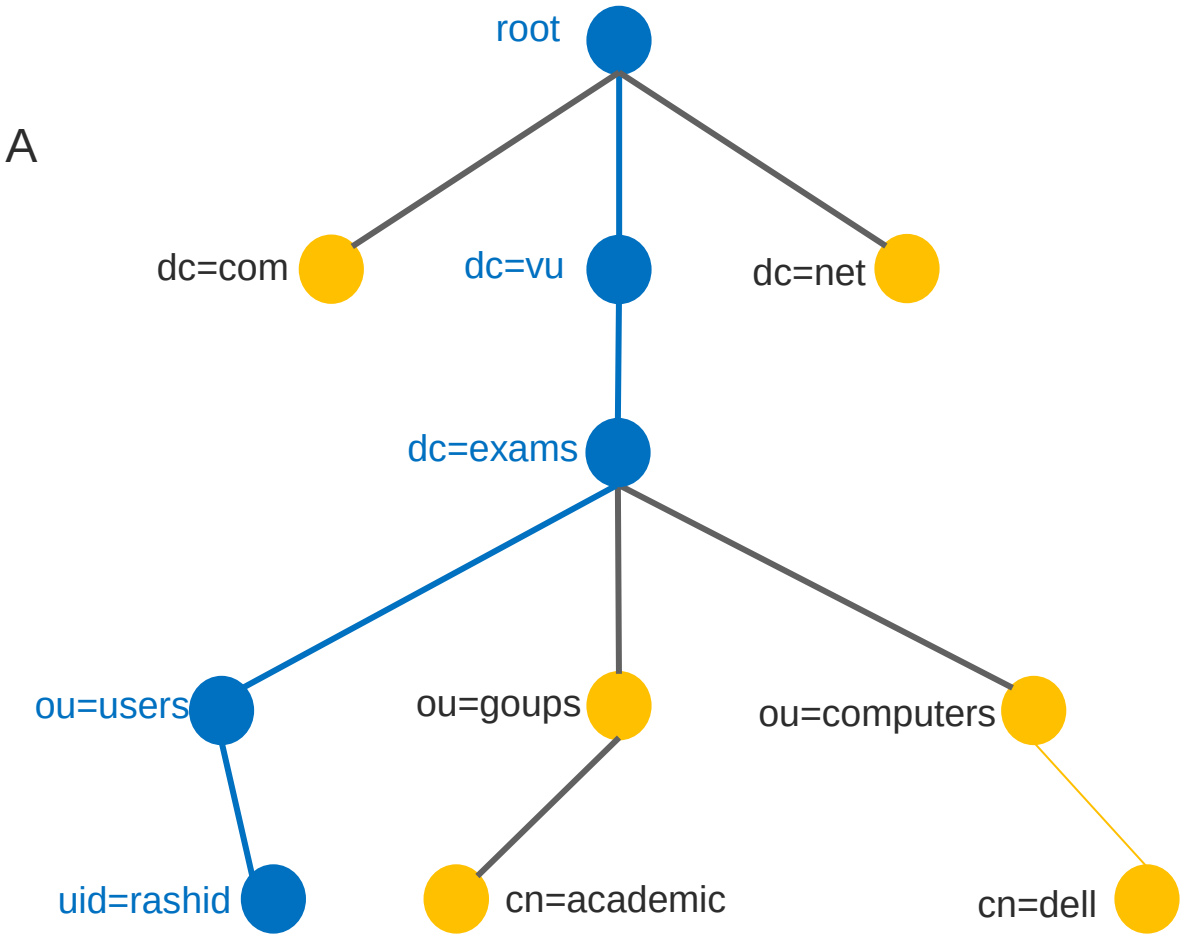

DN RDN

# LDAP Information Structure

- LDAP uses a DIT ( Directory Information Tree) based on X.500 which help present information in the hierarchical tree format.
  - Each node in the LDAP tree is called an entry and is uniquely identified by its Distinguished Name (DN)
  - RFC4514 provides the description of the DN format. A DN corresponding to an entry is highlighted in the diagram and represented by:
    - “ui=rashid,ou=users,dc=exams,dc=vu”

      dc stands for Domain Component
      cn stands for Common Name

- Objectclasses define the attribute structure of an LDAP entry.
- Both ObjectClasses and Attributes are defined within schemas

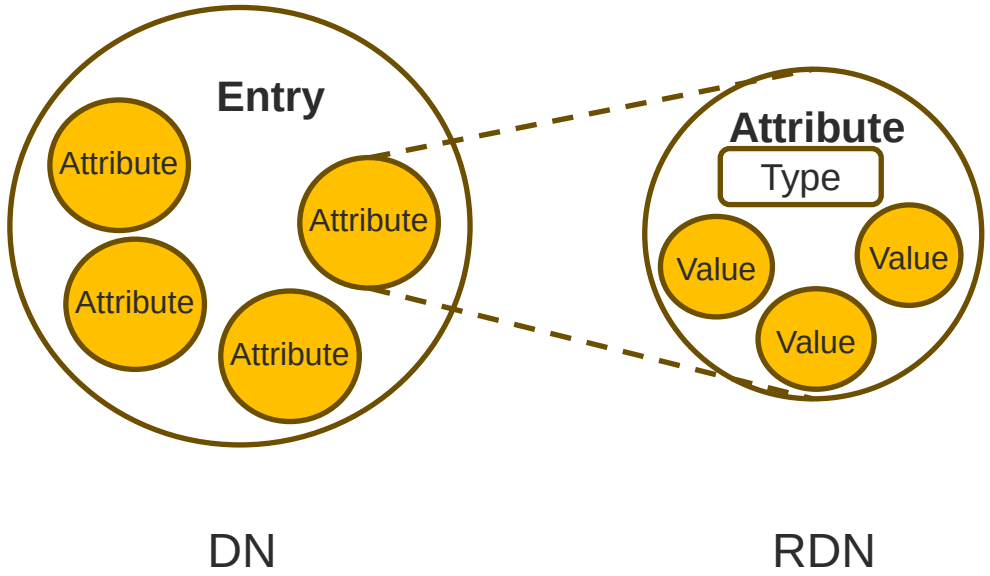


# Schema

- Defines what object classes allowed
- Where they are stored
- What attributes they have (objectClass)
- Which attributes are optional (objectClass)
- Type/syntax of each attribute (objectClass)

# LDAP Information Structure

- An entry consists of a set of attributes, each attribute has a name or type and one or more values.
  - Entry
    - Each entry consists of several attributes
    - Each entry represents  Distinguished Name (DN)
    - DN uniquely identifies an entry in the directory
  - Attributes
    - Each attribute has a type and set of values
  - Relative Distinguished Name (RDN) is an attribute that will make the entry unique in its hierarchy when combined with its parent's Distinguished Name



# LDPA Abbreviations

- O stands for organization
- OU stands for Organizational unit
- SN  stands for Surname
- Givenname stands for First Name
- UID stands for Userid
- Mail stands for Email address
- C stands for country
- L stands for location
- St stands for Status

# Entries Using LDIF Format

- Entries can be represented in a human-readable format by using the LDIF format as in example below.

```
dn: uid=rashid,ou=academics,dc=exams,dc=vu
objectClass: top
objectClass: person
objectClass: orgnizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
objectClass: nfsAccount
cn: Arif Rashid
sn: arifhrashid
givenName:Rashid
uid: arifhrashid
uidNumber: 3000
gidNumber: 500
homeDirectory: /home/arifhrashid
....
```

# Client and Server Interaction

- Client starts an LDAP session by connecting to an LDAP Server
- The default TCP port is 389
- Bind to the server through an authentication process
- Client then sends an operation request to the server
- The Server sends responses in return
- Interaction involves three phases

**Client establishes session with server (BIND)**

- Hostname/IP and port number
- Security
  - User-id/password-based authentication
  - Anonymous connection - default access rights
  - Encryption/Kerberos also supported

**Client performs operations**

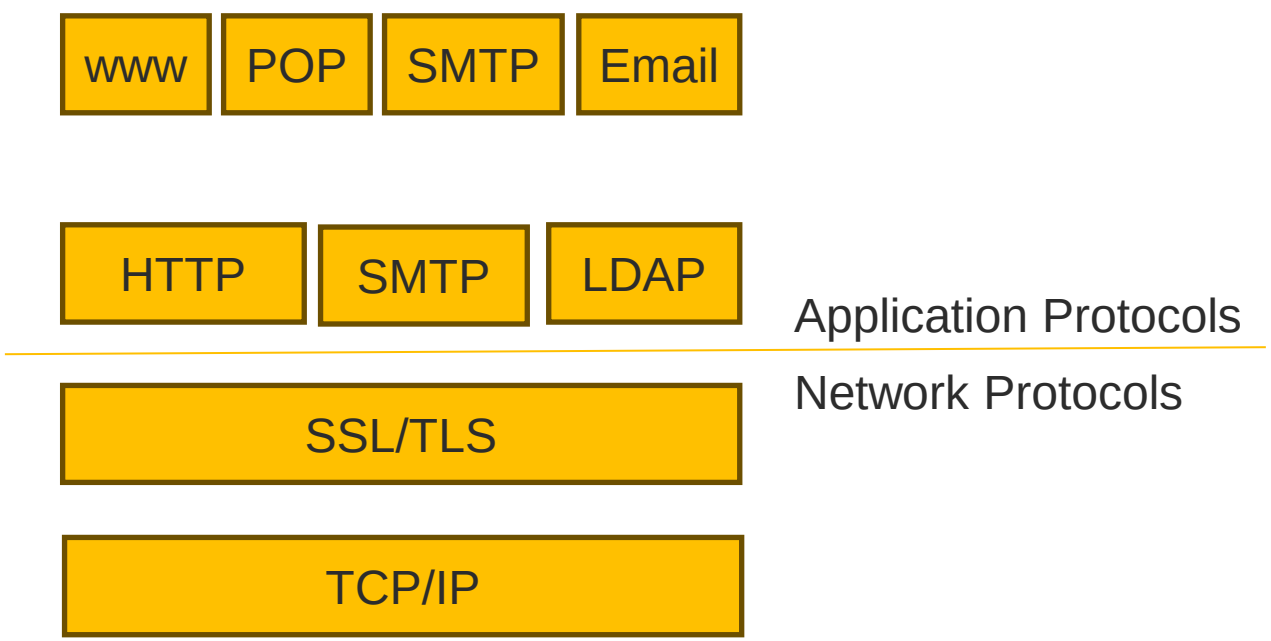
- Read/Update/Search
- SELECT X,Y,Z
- FROM PART_OF_DIRECTORY

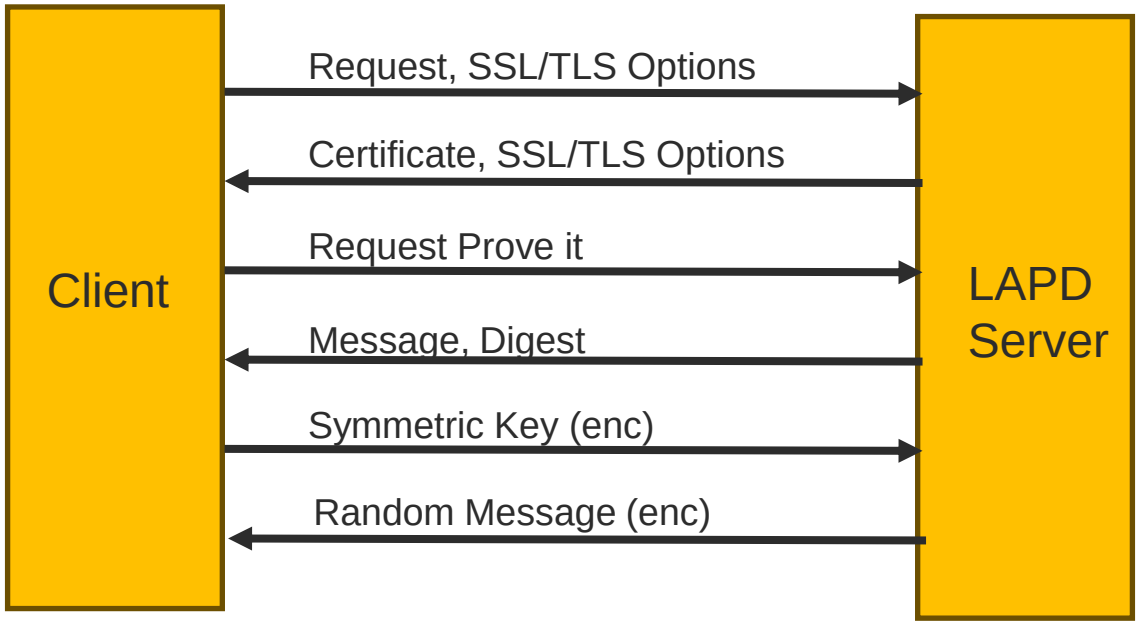
**Client ends the session (UNBIND)**

**Client can ABANDON the session**

# LDAP Security

- **LDAP using Simple Authentication and Security Layer (SASL )With SSL/TLS**



# LDAP Operations

- **Three types of LDAP Operations are Authentication, Query and Update**

**Search**

Search Directory for matching directory entries

**Compare**

Compare directory entry to a set of attributes

**Add**

Add a new directory entry

**Modify**

Updates a particular directory

**Delete**

Removes a specific directory entry

**Rename**

Renames or modify the DN

**Bind**

Start a session with an LDAP Server

**Unbind**

Ends a session with LDAP server

**Abandon**

Ends a session with LDAP server

**Extended**

Extended Operations

# LDAP Common Commands

- **ldapbind**
  - Used to authenticate to a directory server.
  - Also used to find out if the server is running.
  - **Usage**

    ldapbind [options]

  - **Scenario**

To authenticates user arifhrashid to the directory server ldap.vusna.com located at port 389, using the password mypassword.

```
ldapbind -h ldap.vusna.com -p 389 -D "cn=arifhrashid" -w mypassword
```

# LDAP Common Commands

- **ldapsearch**

  Used to search for specific entries in a directory. It opens a connection to a directory, authenticates the user performing the operation, searches for the specified entry, and prints the result in a format that the user specifies.

  - **Usage**

    ldapsearch [options] filter [attributes]

  - **Scenario**

    Searche the directory server ldap.vusna.com, located at port 389. The scope of the search (-s) is base, and the part of the directory searched is the base DN (-b) designated. The search filter "objectclass=*" means that values for all of the entry's object classes are returned. No attributes are returned because they have not been requested. Assume anonymous authentication.

```
ldapsearch -h ldap.vusna.com -p 389 -s base -b "ou=people,dc=vu,dc=com“  "objectclass=*"
```

# LDAP Common Commands

- **ldapadd**

  Used to add entries to the directory. It opens a connection to the directory and authenticates the user. Then it opens the LDIF file supplied as an argument and adds, in succession, each entry in the file.

  - **Usage**

    ldapadd [options] [-f LDIF-filename]

  - **Scenario**

    A user arifhrashid authenticates to the directory ldap.vusna.com, located at port 389. Open the file arifhrashid.ldif and adds its contents to the directory. For example, add the entry uid=arifhrashid,cn=exams,cn=vu,dc=com and its object classes and attributes.

```
ldapadd -h ldap.vusna.com -p 389 -D "cn=arifhrashid" -w mypassword -f arifhrashid.ldif
```

# LDAP Common Commands

- **ldapdelete**

  Used to remove leaf entries from a directory. It opens a connection to a directory server and authenticates the user. Then it deletes specified entries.

  - **Usage**

    ldapdelete [options] "entry DN"

  - **Scenario**

    Authenticates user arifhrashid to the directory ldap.vusna.com, using the password mypassword. Then deletes the entry uid=arifhrashid,ou=academics,ou=people,dc=vu,dc=com.

```
ldapdelete -h ldap.vusna.com -p 389 -D "cn=arifhrashid" -w mypassword \
"uid=arifhrashid,ou=academics,ou=people,dc=vu,dc=com"
```

# LDAP Common Commands

- **ldapmodify**
  - Used to modify existing entries. It opens a connection to the directory and authenticates the user. Then it opens the LDIF file supplied as an argument and modifies the LDAP entries specified by the file.
  - It uses a modified form of an LDIF file. Within the file itself, you use the attribute changetype to specify the type of change.
  - Four types of changes are possible:
    - add--adds a new entry
    - modify--changes an existing entry, that is, it adds, deletes, or replaces attributes of the entry
    - delete--deletes an existing entry
    - modrdn--modifies the RDN of an existing entry
  - **Usage**

    ldapmodify [options] [-f LDIF-filename]
  - **Scenario**

    A user arifhrashid authenticates to the directory ldpa.vusna.com, located at port 389and open the file arifhrashid.ldif and modifies the directory entries specified by the file. The file might, for example, change the telephone number attribute of entry uid= arifhrashid,cn=exams,cn=vu,dc=com.

```
ldapmodify -h ldpa.vusna.com -p 389 -D "cn=arifhrashid" -w mypassword -f arifhrashid.ldif
```

## ldapmoddn

- It is used to:
  - change the RDN of an entry
  - move an entry or subtree to another location in the directory

- **Usage**

  ldapmoddn [options] -b "current DN" -R "new RDN" -N "new Parent"

- **Scenario**

  To authenticates user arifhrashid to the directory ldap.vusna.com, using the password mypassword. Then assign to the entry uid= arifhrashid,ou=exams,ou=people,dc=vu,dc=com a new parent entry, ou=academics,ou=people,dc=vu,dc=com.

```
ldapmoddn -h ldap.vusna.com -p 389 -D "cn= arifhrashid " -w mypassword \
-b " uid= arifhrashid,ou=exams,ou=people,dc=vu,dc=com " \
-N " ou=academics,ou=people,dc=vu,dc=com"
```

# LDAP Security

- **Current LDAP version supports**
  - Clear text passwords
  - KERBEROS version 4 authentication

- **Other authentication methods possible**

- **SASL support added in version 3**
  - Simple Authentication and Security Layer (SASL) is a framework for authentication and data security in Internet protocols.

  - It decouples authentication mechanisms from application protocols, in theory allowing any authentication mechanism supported by SASL to be used in any application protocol that uses SASL.
  - Kerberos deemed stronger than SASL…

# LDAP Security

- **Security based on the BIND model**
- **Clear text -> ver 1**
- **Kerberos -> ver 1,2,3 (depr)**
- **SASL -> ver 3**
  - Simple Authentication and Security Layer
  - uses one of many authentication methods
- **Proposal for Transport Layer Security**
  - Based on SSL v3 from Netscape

# LDAP Security

- **No Authentication**

- **Basic Authentication**
  - DN and password provided
  - Clear-text or Base 64 encoded

- **SASL (RFC 2222)**
  - Parameters: DN, mechanism, credentials
  - Provides cross protocol authentication calls
  - Encryption can be optionally negotiated
  - ldap_sasl_bind() (ver3 call)
  - Ldap://<ldap_server>/?supportedsaslmechanisms

# LDAP Implementations

- **C Library API**
  - LDAPv2 - RFC 1823 ‘The LDAP API’
  - LDAPv3 – In Internet Draft stage

- **Java JNDI**

- **LDAP v3 uses the UTF-8 encoding of the Unicode character set.**

- **HTTP to LDAP gateway**

- **LDAP to X.500 gateway – ldapd**

# Version 2 v/s Version 3

- **Referrals**
  - A server that does not store the requested data can refer the client to another server.
- **Security**
  - Extensible authentication using Simple Authentication and Security Layer (SASL)
- **Internationalization**
  - UTF-8 support for international characters.
- **Extensibility**
  - New object types and operations can be dynamically defined and schema published in a standard manner.

# Common Uses Cases of LDAP

- Centralized User Management
- Centralized Authentication Servers
- Identity and Access Management  (IAM) Solution

# Installation and Configuration

- **Step 1: Set hostname for the Ubuntu server**
- **Step 2: Install OpenLDAP Server**

```
sudo apt update
sudo apt -y install slapd ldap-utils
```

Verify installation with command slapcat  to output SLAPD database contents.

- **Step 3: Add base dn for Users and Groups**
  - Add a base DN for users and groups. Create a file named basedn.ldif and the file by running the command ldapadd
- **Step 4: Add User Accounts and Groups**
  - Generate a password for the user account to add.
  - Create ldif file for adding users.
  - When done with edit, add account by running ldapadd command.
  - Create ldif file for groups and add groups with command ldapadd

# Install LDAP Account Manager

➢ **Step 5: Install LDAP Account Manager**

- Install Apache Web server & PHP

```
sudo apt -y install apache2 php php-cgi libapache2-mod-php php-mbstring php-common php-pear
```

- Then enable php-cgi PHP extension.

```
sudo a2enconf php*-cgi
sudo systemctl reload apache2
```

- Step 3: Install LDAP Account Manager

```
sudo apt -y install ldap-account-manager
```

```
sudo vim /etc/apache2/conf-enabled/ldap-account-manager.conf
```

- comment the line Require all granted and add subnet(s) allowed to access LDAP Account Manager administration interface.

IT601 – System and Network Administration

# File Transfer and Sharing Services

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# File Sharing and Transfer Services

- Following services shall be covered in this module.
  - NFS
  - SMB/CIFS
  - FTP/TFTP/SFTP Services
  - SCP Services

IT601 – System and Network Administration

# NFS Service

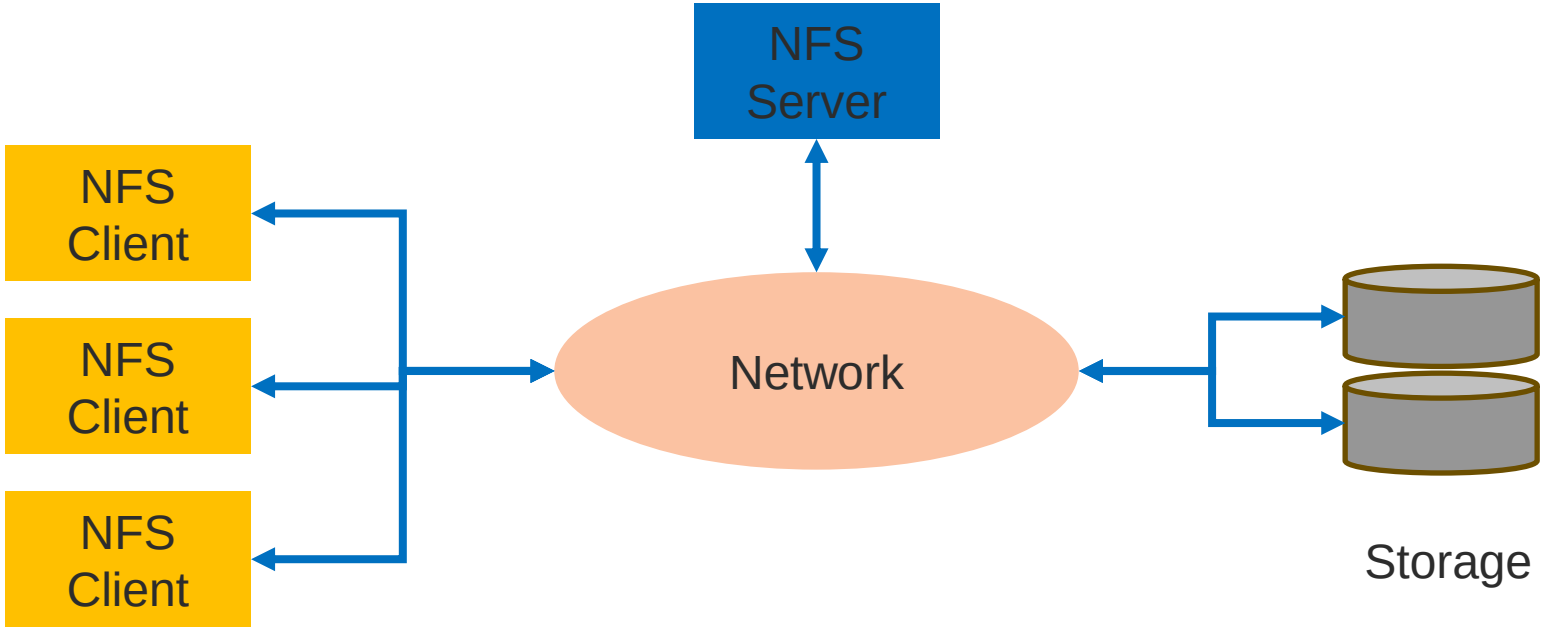
Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Introduction

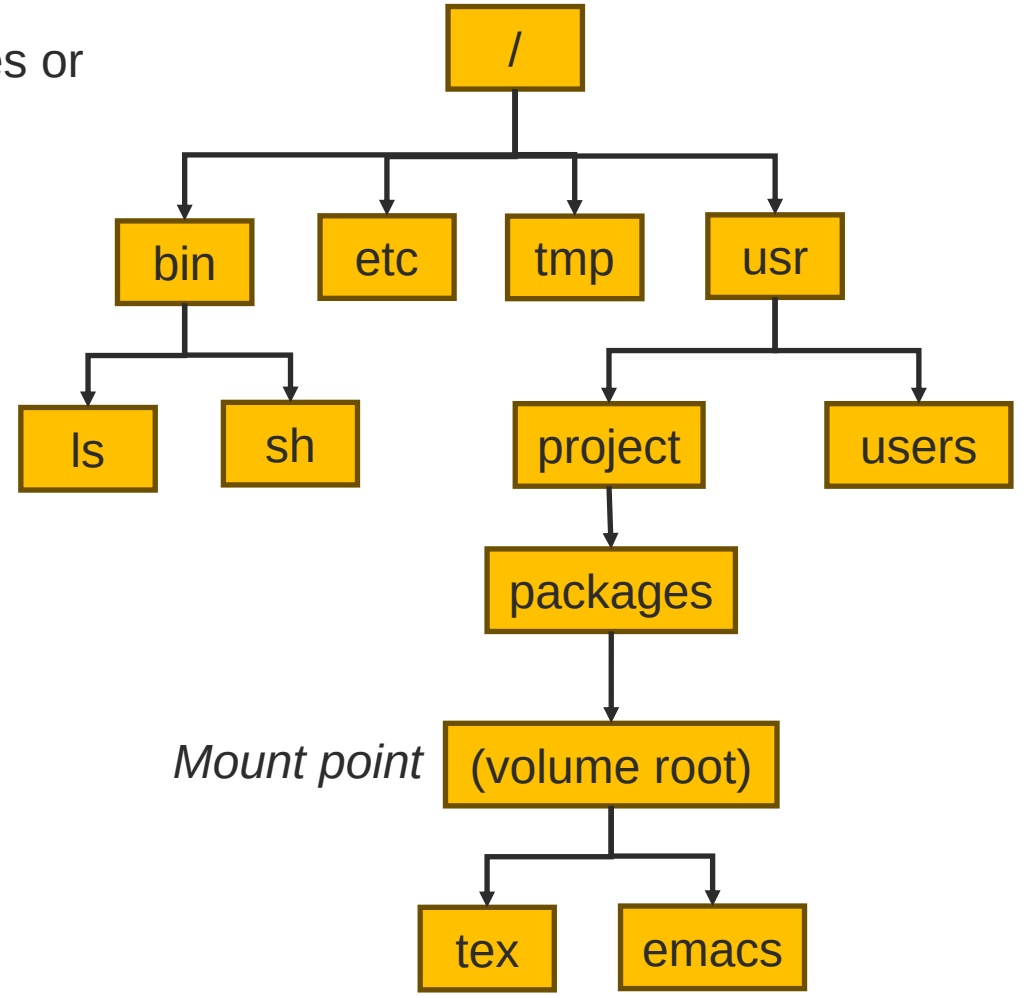
- Network File System (NFS) is a distributed file system protocol developed by Sun Microsystems (Sun) in 1984.
  - It allows a user on a client computer to access files over a computer network much like local storage is accessed.
  - It uses the Open Network Computing Remote Procedure Call (ONC RPC) system.
  - NFS is an open IETF standard defined in a Request for Comments (RFC), allowing anyone to implement the protocol.



# Review of Linux File System

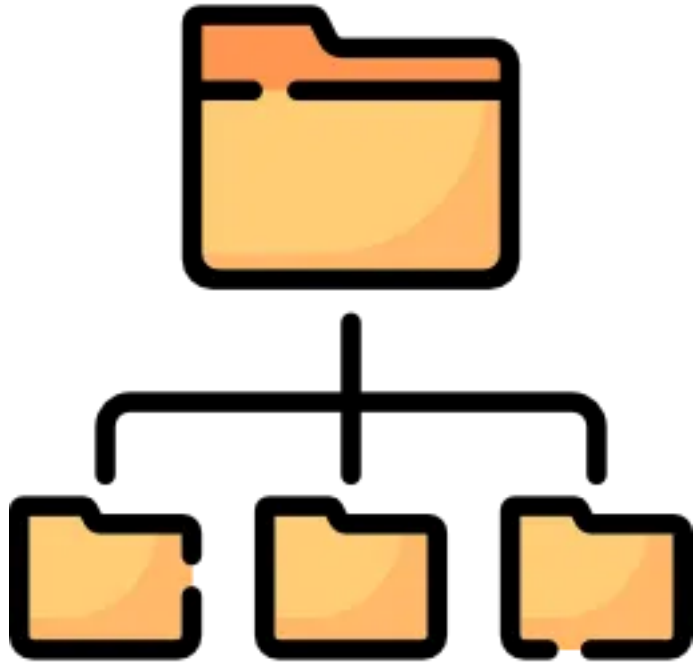
- **A volume is a set of directories and files; a host's file tree is the set of directories and files visible to processes on a given host.**
  - File trees are built by grafting volumes from different volumes or from network servers.
  - The graft operation is the privileged mount system call, and each volume is a filesystem.

- **Mounting volumes**
  - mount (dirs, volume)
    - dirs : directory pathname
    - volume: device specifier or network volume
    - volume root contents become visible at pathname dirs



# Filesystems

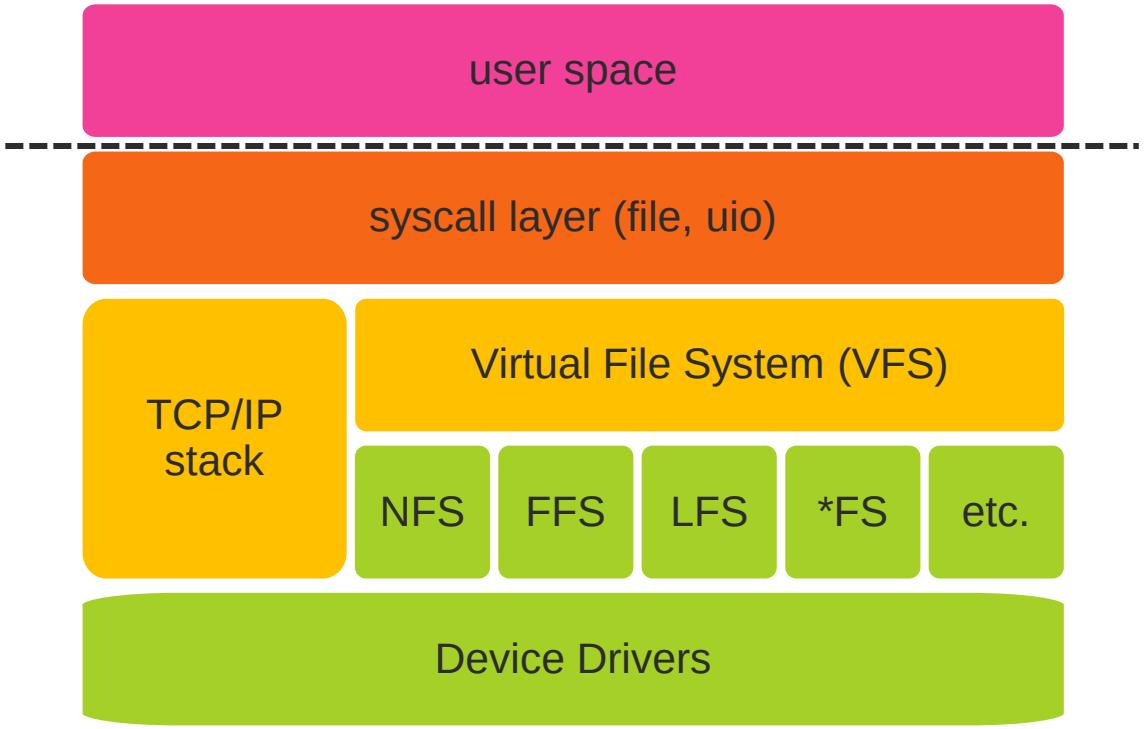
- **Each file volume (filesystem) has a type, determined by its disk layout or the network protocol used to access it.**
  - ufs (ffs), lfs, nfs, rfs, cdfs, etc.
  - Filesystems are administered independently.
- **Modern systems also include “logical” pseudo-filesystems in the naming tree, accessible through the file syscalls.**
  - procfs: the /proc filesystem allows access to process internals.
  - mfs: the memory file system is a memory-based scratch store.
- **Processes access filesystems through common system calls.**



# Virtual File System (VFS)

- **Sun Microsystems introduced the virtual file system interface in 1985 to accommodate diverse filesystem types cleanly.**
  - VFS allows diverse specific file systems to coexist in a file tree, isolating all FS-dependencies in pluggable filesystem modules.

- VFS was an internal kernel restructuring with no effect on the syscall interface.
- Incorporates object-oriented concepts like a generic procedural interface with multiple implementations.
- Based on abstract objects with dynamic method binding by type in C.
- Other abstract interfaces in the kernel: device drivers, file objects, executable files, memory objects.



# Inodes and vnodes

- **The operating system (OS) has three kinds of data structures for files:**

### File table entry

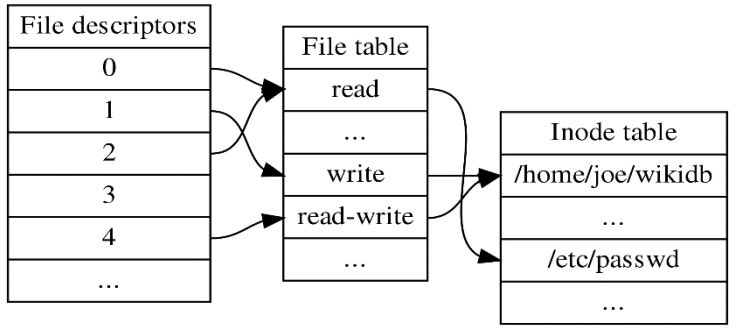
- For each file descriptor[1] for a user there is file table entry.
- It contains information like the current lseek pointer, and a pointer to a vnode.
- When you start your program, the OS has three file table entries one each for stdin, stdout, stderr.
- Each time open() is called, a new file table entry is created in the OS.

### vnode

- There is one of these for each physical file that has been opened.
- It contains a pointer to the file's inode, the file's size, etc.

### inode

- There is one of these for each file on disk.
- It contains all the information returned by stat().

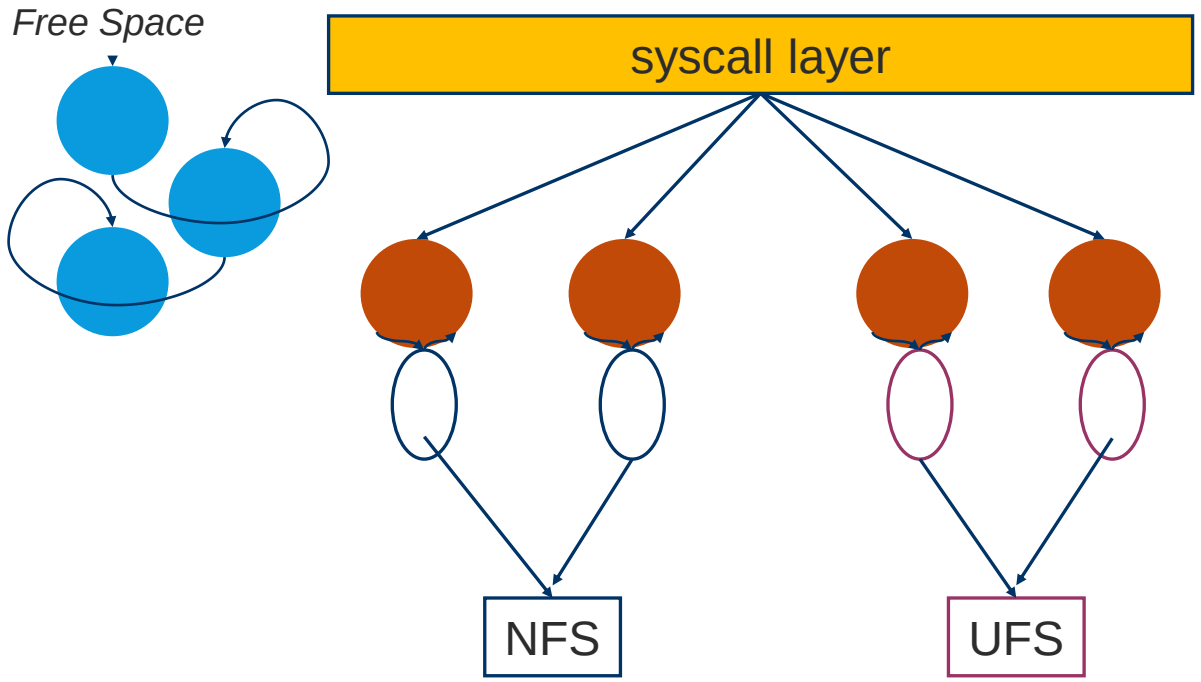



[1]A file descriptor (FD, less frequently fildes) is a process-unique identifier (handle) for a file or other input/output resource, such as a pipe or network socket.

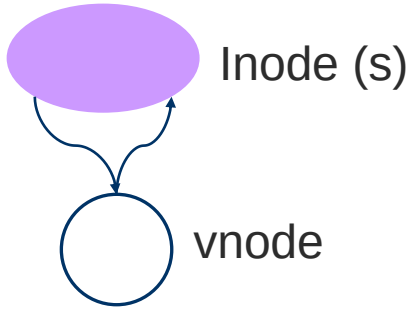
# Inodes and vnodes

- **In the VFS framework, every file or directory in active use is represented by a vnode object in kernel memory.**

  - Each specific file system maintains a cache of its resident vnodes.
  - Each vnode has a standard file attributes struct.
  - Generic vnode points at filesystem-specific struct (e.g., inode, rnode), seen only by the filesystem.
  - Vnode operations are macros that vector to filesystem-specific procedures.



# Vnode Operations and Attributes




**vnode attributes**

- Type
- Mode
- nlink
- ownerID
- ownerGroup
- fsID
- fileID
- fileSize
- accessTime
- modifyTime
- generationNo

**Generic Operations**

- Vop_getattr(vattr)
- vop_setattr (vattr)
- vhold()
- vholdrele()

**Directory Operations**

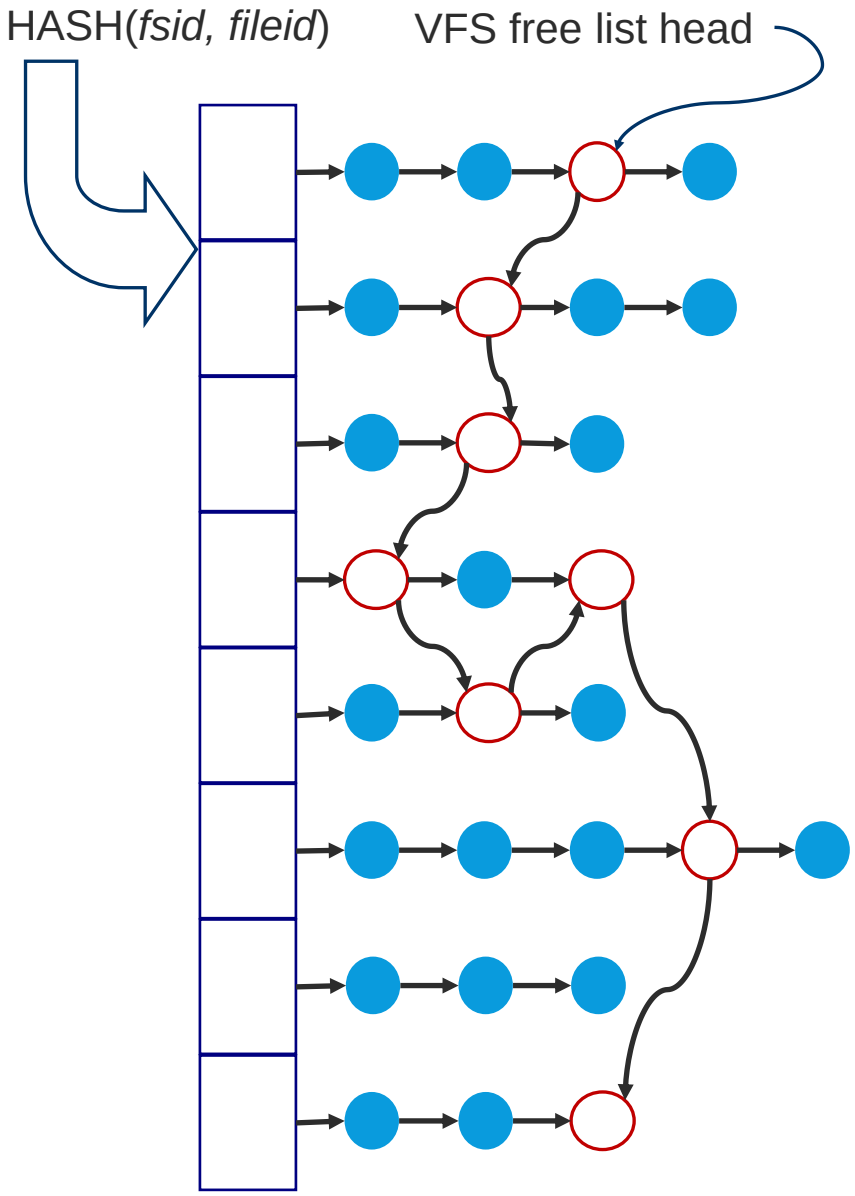
- vop_lookup (OUT vpp, name)
- vop_create (OUT vpp, name, vattr)
- vop_remove (vp, name)
- vop_link (vp, name)
- vop_rename (vp, name, tdvp, tvp, name)
- vop_mkdir (OUT vpp, name, vattr)
- vop_rmdir (vp, name)
- vop_symlink (OUT vpp, name, vattr, contents)
- vop_readdir (uio, cookie)
- vop_readlink (uio)

**Files Operations**

- vop_getpages (page**, count, offset)
- vop_putpages (page**, count, sync, offset)
- vop_fsync ()

# vnode/inode Cache

- Active vnodes are reference- counted by the structures that hold pointers to them.
  - system open file table
  - process current directory
  - file system mount points
  - etc.
- Each specific file system maintains its own hash of vnodes (BSD).
  - specific FS handles initialization
  - free list is maintained by VFS
- Methods
  - vget(vp): reclaim cached inactive vnode from VFS free list
  - vref(vp): increment reference count on an active vnode
  - vrele(vp): release reference count on a vnode
  - vgone(vp): vnode is no longer valid (file is removed)



# Pathname Traversal

- When a pathname is passed as an argument to a system call, the syscall layer must “convert it to a vnode”.
  - Pathname traversal is a sequence of vop_lookup calls to descend the tree to the named file or directory.
    - open(“/tmp/zot”)
    - vp = get vnode for / (rootdir)
    - vp->vop_lookup(&cvp, “tmp”);
    - vp = cvp;
    - vp->vop_lookup(&cvp, “zot”);
- Issues in pathname traversal
  - crossing mount points
  - obtaining root vnode (or current dir)
  - finding resident vnodes in memory
  - caching name->vnode translations
  - symbolic (soft) links
  - disk implementation of directories
  - locking/referencing to handle races
  - with name create and delete operations

# NFS Protocol

- NFS is a network protocol layered above TCP/IP.
  - Original implementations  use UDP transport for low overhead.
  - Maximum IP packet size was increased to match FS block size, to allow send/receive of entire file blocks.
  - Some implementations of NFS also use TCP as a transport.

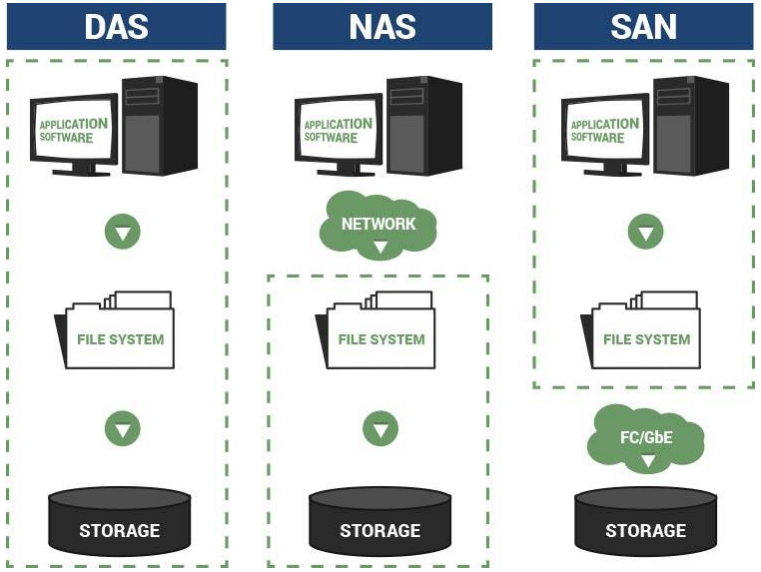
- The NFS protocol is a set of message formats and types.
  - Client issues a request message for a service operation.
  - Server performs requested operation and returns a reply message with status and (perhaps) requested data.

# Network Block Storage

- One approach to scalable storage is to attach raw block storage to a network.
  - abstraction: OS addresses storage by <volume, sector>.
    - Examples are iSCSI, Petal, FC: access through souped-up device driver
  - dedicated Storage Area Network or general-purpose network
    - Examples are FibreChannel vs. Ethernet
  - shared access with scalable bandwidth and capacity
  - volume-based administrative tools
    - backup, volume replication, remote sharing
- Called “raw” or “block”, “storage volumes” or just “SAN”.

# NAS and SAN

- **Network Attached Storage (NAS)**
  - In NAS, one or more dedicated file server or storage devices are made available in a LAN. Therefore, the transfer of data, particularly for backup, still takes place over the existing LAN.




- **Storage Area Network (SAN)**
  - SAN is a network which provides access to consolidated, block-level data storage. SANs are primarily used to access data storage devices like disk arrays and tape libraries from servers so that the devices appear to the operating system as direct-attached storage.
- **There has been some debate over whether NAS and SAN is good for enterprises.**
  - Network-Attached Storage has been the dominant approach to shared storage since NFS.
  - NAS == NFS or CIFS: named files over Ethernet/Internet.
  - Network Appliance Known as filers
- **FibreChannel (FC) SAN is considered to be fundamentally faster way to access shared storage.**
  - No indirection through a file server
  - Lower overhead on clients
  - Network is better/faster (if not cheaper) and dedicated/trusted
  - Brocade, HP, Emulex are some big players.

# NAS and SAN: What is better?

- FC is a high-end technology incorporating NIC enhancements to reduce host overhead but bogged down in interoperability problems.

- Ethernet is getting faster faster than FC such as gigabit, 10-gigabit, + smarter NICs, + smarter/faster switches

- The choice of network is fundamentally orthogonal to storage service design.
  - Well, almost: flow control, RDMA, user-level access

- The fundamental questions are really about abstractions. For example, shared raw volume vs. shared file volume vs. private disks

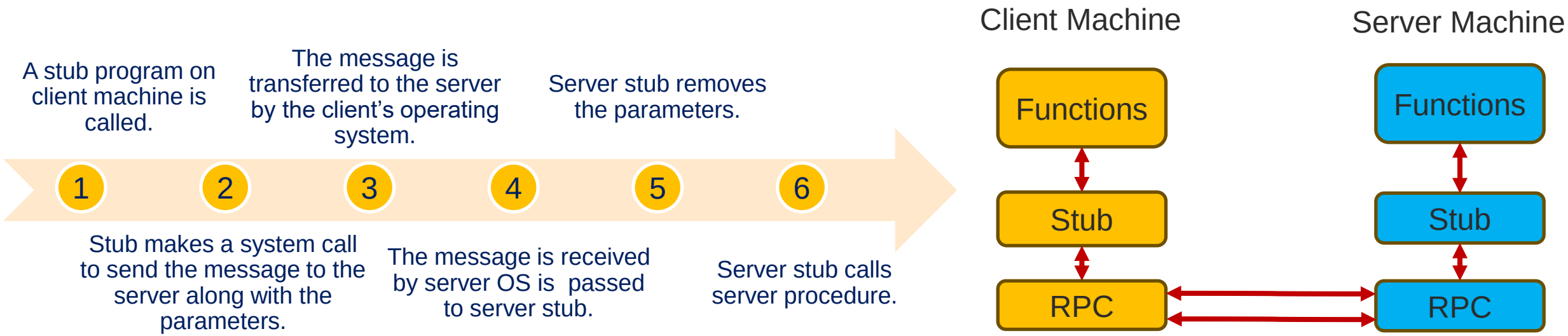
# Abstractions for storage

- Relational database (IBM and Oracle) such as tables, transactions, query language
- File system  such as hierarchical name space with ACLs
- Block storage such as SAN, Petal, RAID-in-a-box (e.g., EMC)
- Object storage where object == file, with a flat name space: NASD, DDS
- Persistent objects such as pointer structures, requires transactions: OODB, ObjectStore

# Remote Procedure Call (RPC)

- A remote procedure call is an inter-process communication technique used for client-server-based applications.
  - A client sends request message that the RPC translates and sends to the server. The request may be a procedure or a function call to a remote server.
  - When the server receives the request, it sends the required response back to the client. The client is blocked while the server is processing the call and only resumed execution after the server is finished.

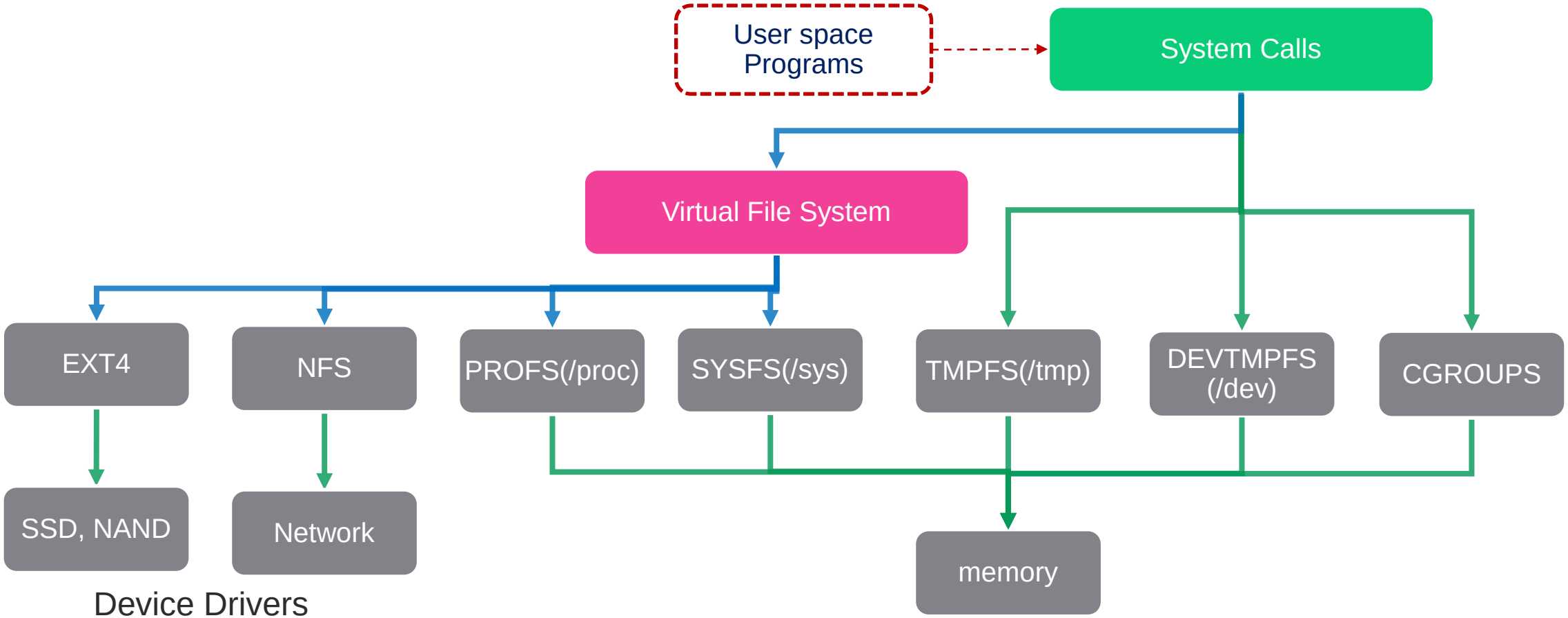
- The sequence of events involved in RPC are :-



# Virtual Filesystem

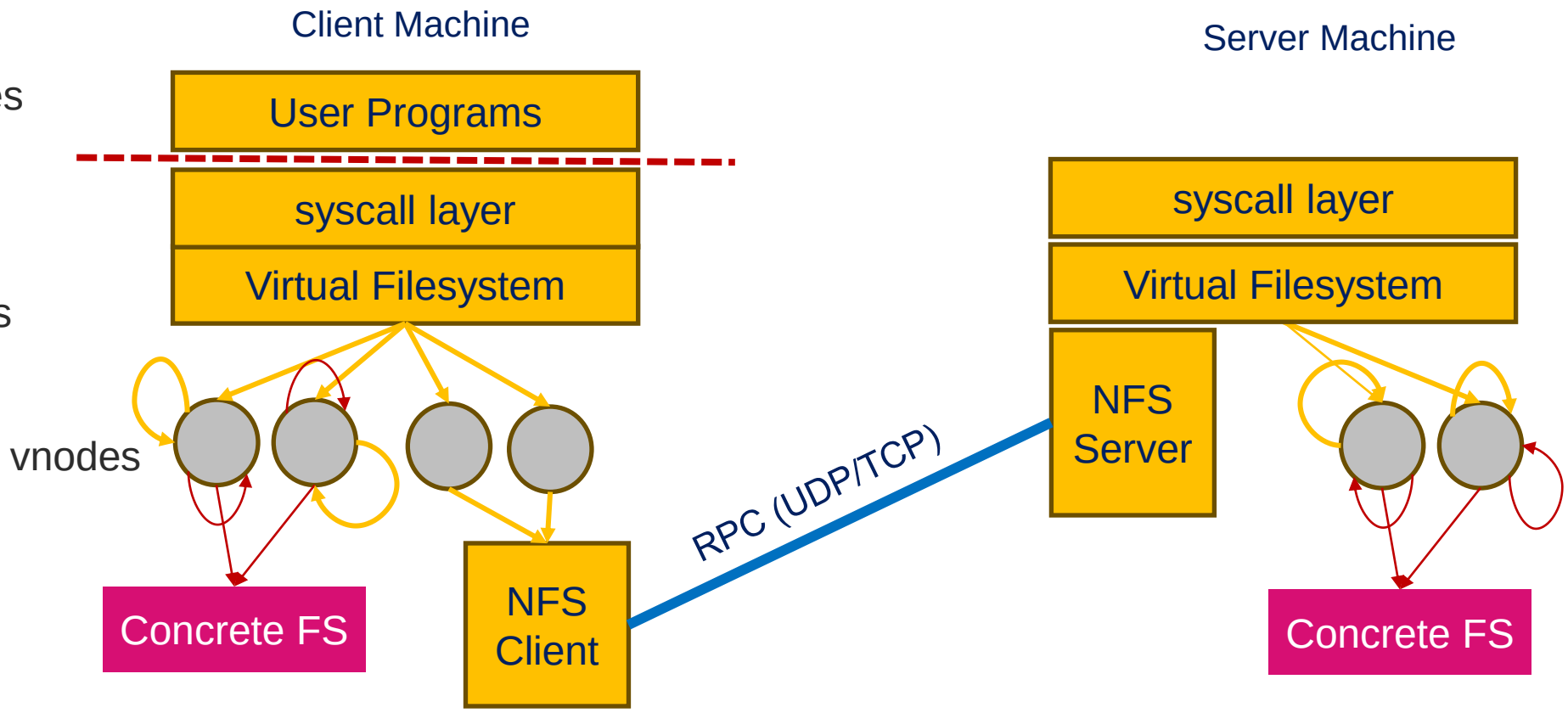
- A virtual file system (VFS) or virtual filesystem switch is an abstract layer on top of a more concrete file system.

  It allows client applications to access different types of concrete file systems in a uniform way.



# Network File System (NFS)

- VFS abstracts the concrete FS
  - VFS builds the vnodes corresponding to inodes of concrete FS
  - Users interreacts with VFS using system calls
  - NFS client and server implement the RPC
  - RPC communicates of TCP/UDP ports
  - NFS implements stubs
  - The nfsnode holds client state needed to interact with the server to operate on the file.



# NFS file reference

- In NFS, the reference is a file handle or fhandle, a token/ticket whose value is determined by the server.
  - Includes all information needed to identify the file/object on the server, and find it quickly.

| volume ID | inode # | generation # |
|---|---|---|

# Stateless Behavior of NFS

- **Stateful behavior vs Stateless Services**

- **A classical NFS server maintains no in-memory hard state.**
  - The only hard state is the stable file system image on disk.
  - no record of clients or open files
  - no implicit arguments to requests
    - no server-maintained file offsets: read and write requests must explicitly transmit the byte offset for each operation.
  - no write-back caching on the server
  - no record of recently processed requests

- **Stateless behavior of NFS makes failure recovery simple and efficient.**

# Stateless Behavior of NFS

## ➢ Recovery in Stateless NFS

### Server Restart

- No Need to rebuild in-memory state of server
- Client reestablishes TCP Connection
- Client retransmits pending requests.

### Connectionless

- Classical NDS uses which is connectionless protocol
- Server failure is transparent to the client; no connection to break or reestablish.
- A crashed server is indistinguishable from a slow server.

### Error Masking

- RPC masks network errors by retransmitting a request after an adaptive timeout.
- A dropped packet is indistinguishable from a crashed server.

## ➢ Although simple but Stateless behavior has problems

### Constrained interface

- Recovery-by-retransmission constrains the server interface.
- RPC/UDP has execute-at-least-once semantics (“send and pray”), which compromises performance and correctness.

### Update Operations

- Update operations are disk-limited.
- Updates must commit synchronously at the server.

### Inconsistency

- NFS cannot (quite) preserve local single-copy semantics.
- Files may be removed while they are open on the client.
- Server cannot help in client cache consistency.

# Stateless Behavior of NFS : Potential Remedies

## Retransmissions and Idempotency

- **Hope for the best and smooth over non-idempotent requests.**
  - map ENOENT and EEXIST to ESUCCESS.
- **Use TCP or some other transport protocol that produces reliable, in-order delivery.**
  - However, higher overhead...and we still need sessions.
- **Implement an execute-at-most once RPC transport.**
  - TCP-like features (sequence numbers)...and sessions.
- **Keep a *retransmission cache* on the server.**
  - Remember recent request IDs and their results, and just resend the result.

## Synchronous Writes

- **Delay the response until convenient for the server.**
  - NFS *write-gathering* optimizations for clustered writes (similar to *group commit* in databases).
  - Relies on write-behind from NFS I/O daemons (*iods*).
- **Throw hardware at it: non-volatile memory (NVRAM)**
  - Battery-backed RAM or UPS (uninterruptible power supply).
  - Use as an operation log (Network Appliance WAFL) or as a non-volatile disk write buffer (Legato).
- **Replicate server and buffer in memory (e.g., MIT Harp).**

## File Cache Consistency

- **Timestamp invalidation (NFS).**
  - Timestamp each cache entry, and periodically query the server: “has this file changed since time **t**?”; invalidate cache if stale.
- **Callback invalidation (AFS, Sprite, Spritely NFS).**
  - Request notification (callback) from the server if the file changes; invalidate cache and/or disable caching on callback.
- **Leases (NQ-NFS) Later: distributed shared memory**

# NFS V3

- **NFS V3 sidesteps the synchronous write problem by adding a new asynchronous write operation.**
  - Server may reply to client as soon as it accepts the write, before executing/committing it.
    - If the server fails, it may discard any subset of the accepted but uncommitted writes.
  - Client holds asynchronously written data in its cache and reissues the writes if the server fails and restarts.
    - When is it safe for the client to discard its buffered writes?
    - How can the client tell if the server has failed?
- **NFS V3 adds a new commit operation to go with async-write.**
  - Client may issue a commit for a file byte range at any time.
  - Server must execute all covered uncommitted writes before replying to the commit.
  - When the client receives the reply, it may safely discard any buffered writes covered by the commit.
  - Server returns a verifier with every reply to an async write or commit request.
  - The verifier is just an integer that is guaranteed to change if the server restarts, and to never change back.
  - What if the client crashes?

# NFSv4 Features

- Several new features have been added to NSF v4.

## State-fullness

- The usage information of an object by an NFSv4 client is maintained by the server.
- opening, locking, reading, and writing, carry state information that notify the server of the intentions on the object by the client.
- The server can return information to a client about other clients having intentions on the same object.
- The use of a persistent open on the server avoids some situations where an NFS version 2 or 3 client could become locked out of a file while writing to it.

## Byte-range locking and share modes

- It provides support for byte-range locking and share modes as part of the base protocol.
- Locking in NFSv4 is lease-based, which requires the NFSv4 client to maintain contact with the server to preserve open and lock state owned by the client.

## Compound request format

- An NFSv4 client can combine several simple operations (for example, LOOKUP, OPEN, and READ) into a single request to the server.
- The single request allows NFSv4 to perform a complex operation in one network exchange.

# NFSv4

| improved security | standardizes the representation of string data | Unification | RPC Over TCP |
|---|---|---|---|
| ➢ The NFSv4 protocol specifies improved security mechanisms over the others required by previous protocol versions.<br>➢ provides support for Kerberos 5 authentication and data protection in addition to the traditional AUTH_SYS security.<br>➢ The security API used by NFSv4 allows for easy addition of new security mechanisms in the future. | ➢ The NFSv4 protocol standardizes the representation of string data.<br>➢ All string data used by the protocol is represented in UTF-8 as it crosses the network.<br>➢ User and group information is passed between the client and server in string form, not as numeric values as in previous versions. | ➢ The NFSv4 protocol combines the separate component protocols of previous NFS versions into a single protocol specification.<br>➢ The single point of contact for the NFSv4 protocol allows for better compatibility with network fire walls. | ➢ NFS Version 4 requires support of RPC over streaming network transport protocols such as TCP.<br>➢ The NFSv4 support provided by IBM i uses TCP exclusively. |

# Installing NFS on Linux Server

- **Install the nfs-kernel-server software package.**
  - sudo apt install nfs-kernel-server
- **Start the NFS server**
  - sudo systemctl start nfs-kernel-server.service
- **Configure the directories to be exported by adding them to the /etc/exports file.**
  - /srv *(ro,sync,subtree_check)
  - /cshare *.vusna.com(rw,sync,no_subtree_check)
  - /scratch *(rw,async,no_subtree_check,no_root_squash)
- **Apply the new config via**
  - sudo exportfs -a
- **Check status or restart service**
  - sudo systemctl restart nfs-kernel-server

- **rw**
  - It gives the client computer both read and write access to the volume.
- **sync**
  - It forces NFS to write changes to disk before replying. This results in a more stable and consistent environment but reduces the speed of file operations.
- **no_subtree_check**
  - It prevents subtree checking, which is a process where the host must check whether the file is actually still available in the exported tree for every request. This can cause many problems when a file is renamed while the client has it opened. In almost all cases, it is better to disable subtree checking.
- **no_root_squash**
  - By default, NFS translates requests from a root user remotely into a non-privileged user on the server. This was intended as security feature to prevent a root account on the client from using the file system of the host as root. no_root_squash disables this behavior for certain shares.

# NFS Client Configuration

- **To enable NFS support on a client system**
  - sudo apt install nfs-common

- **Use the mount command to mount a shared NFS directory from another machine**
  - sudo mkdir /opt/cshares
  - sudo mount cshares.vusna.com:/srv /opt/cshares

  The mount point directory /opt/example must exist. There should be no files or subdirectories in the /opt/example directory, else they will become inaccessible until the nfs filesystem is unmounted.

- **The general syntax for the line in /etc/fstab file is as follows:**
  - cshares.vusna.com:/srv /opt/cshares nfs rsize=8192,wsize=8192,timeo=14,intr

IT601 – System and Network Administration

# SMB Service

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Server Message Block (SMB)

- Server Message Block (SMB) is a client-server protocol that provides access to resources such as files, printers and serial interfaces, and facilitates communication between network processes.
- SMB clients can communicate with any software that is configured to receive SMB requests over TCP/IP or NetBIOS.
- Operates as an application-level network protocol
- Provides an authenticated Inter-process communication mechanism.
- Most usage of SMB involves computers running Microsoft Windows:
- With SMB, you can mount a shared file folder directly on a local Windows or MacOS machine, or in a cloud virtual machine. Modern versions of SMB provide security features such as AES-based data encryption.
- SMB was formerly known as CIFS (Common Internet File System)—this is an old version of the SMB protocol which was decommissioned because it was inefficient and had severe security flaws.
- The SMB protocol operates in “request-response” mode—several messages are sent between the client and the server to establish a connection.

# History

- Barry Feigenbaum originally invented SMB at IBM
  - Turn DOS "Interrupt 33" (21h) local file-access into a networked file-system
- Microsoft made considerable modifications to the version used most commonly
  - Merged the SMB protocol with the LAN Manager
    - Developing with 3Com (circa 1990)
  - Continued to add features to the protocol
    - Windows for Workgroups (circa 1992)
  - Later versions of Windows.
- Original design of SMB envisaged it running on top of the NetBIOS and NetBEUI APIs
  - Typically implemented with NBF, NetBIOS over IPX/SPX, NBT
- SMB can also run directly on the TCP/IP protocols Introduced with Windows 2000.
- About when Sun Microsystems announced WebNFS Microsoft launched an initiative in 1996 to rename SMB to Common Internet File System (CIFS)
  - Added more features, including
  - Support for symbolic links, hard links, larger file sizes-and an attempt at supporting direct connections without all the NetBIOS trimmings
  - Largely experimental effort that required further refinement

# SMB Versions

- **Commonly used versions of SMB are**

| CIFS | SMB 2.0 | SMB 3.0 |
|---|---|---|
| • Common Internet File System<br>• Also Known as SMB1<br>• a protocol that was extremely chatty and slowed down WANs due to the extra load it created. It also suffered from major security vulnerabilities. | • improved on CIFS and reduced chattiness by reducing the number of protocol commands from hundreds to under 20 and added support for symbolic links.<br>• Also resolved critical security issues. | • an improved version of the protocol which provided features like SMB Direct, SMB Transparent Failover, and important security features including AES encryption. |

- **Older versions of SMB**
  - Older versions of SMB used legacy protocols like IPE or NetBEUI.
  - Modern SMB software and devices commonly communicate directly over TCP/IP, or if this is not supported, via NetBIOS over TCP/IP.
  - Clients and servers can implement different versions of SMB and negotiate versions and capabilities before connecting.

# Supported OS

- Operating systems that support SMB include:
  - Microsoft Windows—all versions since Windows 95
  - Linux kernel
  - MacOS—starting from MacOS X Lion

- Free and open source implementations of SMB include:
  - FreeBSD—based on an SMB client implementation called smbfs
  - Samba—a server that implements the SMB protocol and Microsoft extensions

# SMB Direct

- Newer versions of windows servers include a feature called SMB Direct, which supports Remote Direct Memory Access (RDMA) network adapters.
  - RDMA-compatible network cards provide high performance with very low latency and low use of CPU resources.
- To use SMB Direct, the following conditions must be met:
  - Connection is between two or more computers are running Windows Server 2012 R2, Windows Server 2012 or later
  - One or more of the computers are equipped with RDMA-compatible network adapters
- When using SMB Direct in a failover scenario, ensure that the cluster network provides sufficient performance for SMB Direct, and that all cluster nodes have Receive Side Scaling (RSS) and RDMA network adapters.
- SMB Multichannel checks the network adapter capabilities of connection partners. You must enable SMB Multichannel to use SMB Direct.

# SMB Security Threats

## EternalBlue

- In 2017, a serious vulnerability called EternalBlue was found in SMB Version 1 (SMBv1).
- The vulnerability allowed an attacker to install malware on any computer running SMB1, without any action required by the user.
- Microsoft released an emergency patch (MS17-010) for this vulnerability that covered all supported Windows versions.

## WannaCry

- WannaCry was an attack that leveraged the EternalBlue vulnerability.
- It spread very quickly, destroying compromised systems.
- If SMB1 is enabled on a system, WannaCry can use it without any user intervention, install ransomware payloads, and then scan and infect other SMB1 compatible systems connected to the infected system.
- WannaCry caused significant damage for governments, institutions and companies from the medical, automotive, communications, transportation and other industries. Microsoft took an unprecedented action and provided fixes for end-of-life versions of Windows, including Windows XP.

## Nyetya

- Nyetya was originally conceived as a supply chain attack, and was also distributed via EternalBlue.
- It also took advantage of another SMB vulnerability called EternalRomance, which was very effective in old Windows versions. Nyetya appeared to be Ransomware, but in fact it was wipeware.
- It displayed a Ransomware message, but users couldn't pay, and all data on infected systems was lost.

## Others

- There are additional scenarios in which attackers leverage the SMB protocol, even without relying on a vulnerability.
- Threats like Bad Rabbit, Olympic Destroyer and SamSam used various methods to gain access to a network, and once inside, used SMB to gain access to sensitive systems.
- In other cases, attackers conducted brute force attacks on SMB shares, trying a large number of passwords until they gained access to sensitive data.

# SMB Security Features

- As a response to SMB security vulnerabilities and widespread attacks, Microsoft introduced several important security features.

## SMB Encryption

- Provides end-to-end encryption of all data transmitted over SMB, preventing interception of communications on unsecured networks.
- SMB encryption does not require IPsec or WAN acceleration to operate.
- It can be configured on a specific file share or a full file server.
- SMB encryption is an important measure for protecting sensitive data and preventing man in the middle attack.

## Secure Dialect Negotiation

- SMB 3.0 can detect attacks that attempt to downgrade the protocol from 3.0 to 2.0, or remove essential security capabilities.
- When a client or server detects such an attack, the connection is terminated and a security event is recorded in the event log.
- However, secure language negotiation cannot detect or prevent a downgrade to SMB 1.0.
- it is essential to disable SMB 1.0 server, which has critical security flaws, on any legacy system that still supports it.

## New Signing Algorithm

- SMB 3.0 uses modern cryptographic algorithms for signing, in particular AES-CMAC and AES-CCM.
- These modern algorithms can significantly accelerate encryption on modern CPUs. SMB 2.0 also supports encryption, but using the older HMAC-SHA256 algorithm.

# Versions and implementations

- The following list explicitly refers to "SMB" as including an SMB client or an SMB server
  - Including the various protocols that extend SMB
  - For simplicity and conciseness and vagueness, however, the list omits mention of the extent or completeness of the reimplementation or porting status for any of these implementations
  - Lumps them all together simply as "SMB"

# Versions and implementations

- Samba, a free re-implementation of the SMB protocol and of the Microsoft extensions to it, includes an SMB server and a command-line SMB client.
- Samba TNG: a fork of Samba.
- The Linux kernel includes two SMB client implementations that use the Linux VFS, providing access to files on an SMB server through the standard file system API: smbfs and cifs.
- FreeBSD includes an SMB client implementation called smbfs that uses its VFS.
- Solaris has a project called CIFS client for Solaris, based on the Mac OS X smbfs.
- Sun Microsystems added in-kernel CIFS support to Solaris in October 2007
- FreeNAS, a dedicated small-sized NAS server, runs FreeBSD for Network-attached storage (NAS) services, and supports protocols including CIFS/Samba
- Network Appliance has an SMB server implementation
- JCIFS offers an implementation of SMB in Java

# IT601 – System and Network Administration

# FTP Service

Arif Husen

**Department of Computer Science and Information Technology,**

**Virtual University of Pakistan**

# File Transfer Protocol (FTP)

- **FTP allows user to install and transfer files or folders from one system to the other.**
  - It also works in a similar way as HTTP for transferring Web pages from a server to a user browser.
  - File Transfer protocol is mostly used for downloading a file from a server and then is uploaded as a Web page file.
- **The term FTP is owned by Apache**
- **Required FTP software to be installed on  both machines involved in transferring file**
- **Different software implementations are available, some are paid and some are free to use.**

# History of File Transfer Protocol(FTP)

- In April of 1971 the first FTP standard was created which was RFC 114. It was invented before TCP/IP was even invented.

- Objectives
  - to promote sharing of files
  - to shield a user from variations in file storage systems among hosts
  - to transfer data reliably and efficiently.

- Made for Interoperability
  - Many public organizations support FTP Servers
  - Allows transfer across the OS, Systems and Different architectures

# Procedure to used

- In order to Setup an FTP Service one must do the following:
  - Install FTP software (Server and Client)
  - Select file or files which need to be transferred
  - Send file or files by computer networks
  - Final step: File or files start to transfer

# ADVANTAGES OF FTP

- FTP can be used numerous amounts of time {Unlimited}
- FTP can be used any time
- Files can be transferred without any hassle
- No limit as to how many files or folders can be transferred
- Any type of file or folder can be transferred
- Easy to install on machine

# Disadvantages of File Transfer Protocol

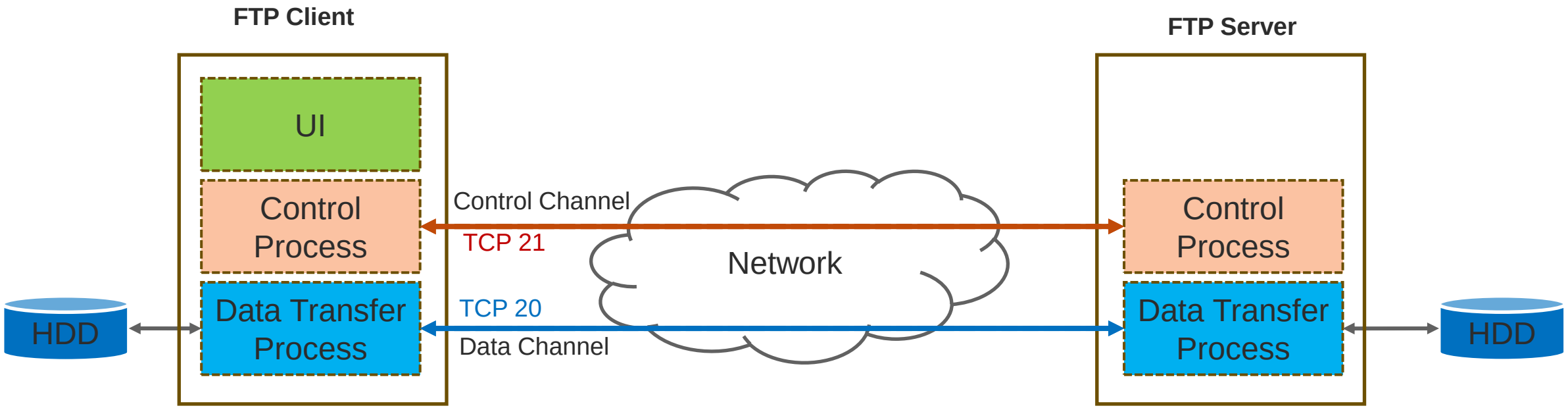
- Mostly used on Unix and Linux computer systems
- Not compatible with every system and lacks support
- ASCII mode and Binary Mode are used and differ from the way they send data.

# The problems FTP can solve

- File Transfer Protocol (FTP) is the standard mechanism provided by TCP/IP for copying a file from one host to another.
- Although transferring files from one system to another seems simple and straightforward, some problems must be dealt with first.
  - The systems may use different file name conventions
  - The systems may have different ways to represent text and data.
  - The systems may have different directory structures.
- All of these problems have been solved by FTP in a very simple and elegant approach.

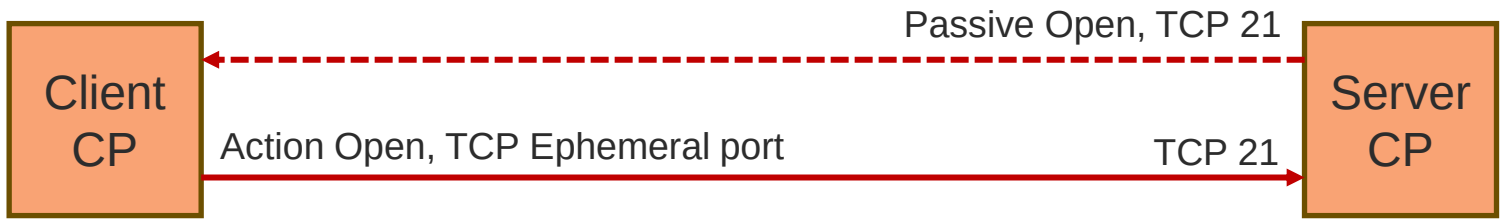
# FTP Connections

- FTP uses the services of TCP.
  - It needs two TCP connections.
    - The well-known port 21 is used for the control connection.
    - The well-known port 20 for the data connection.



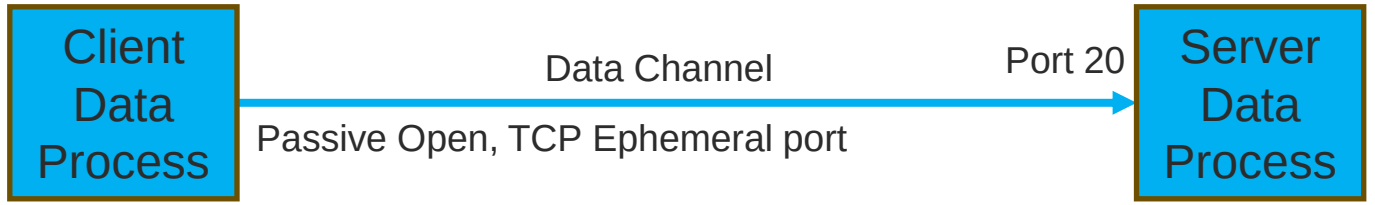
# FTP Control Connection Steps

- Opening of Connection takes place in two steps
  - Server Control Process Opens a passive connection
  - Client Control Process Opens an active connection

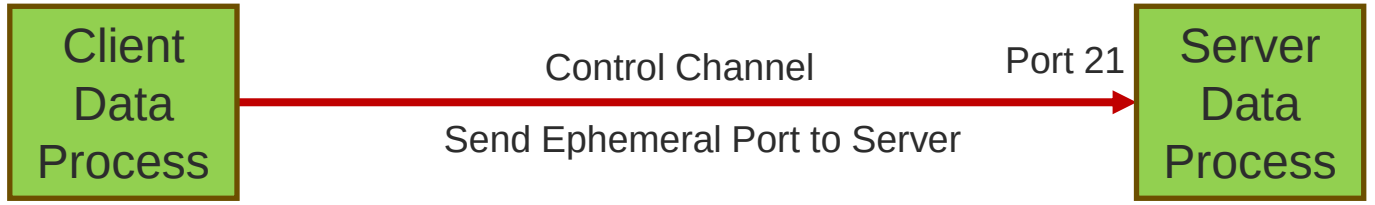


# FTP Control Connection Steps

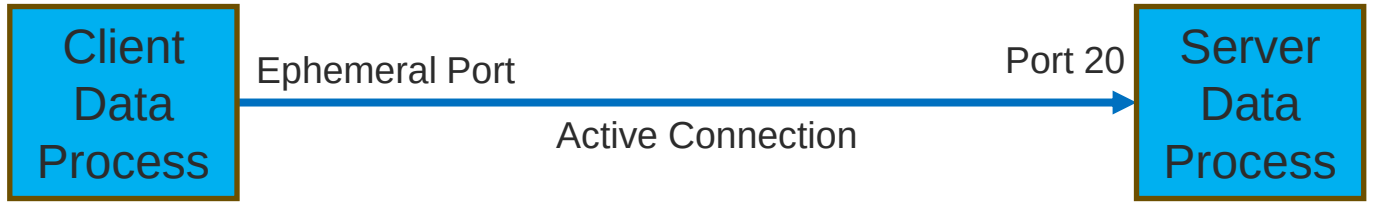
- Client opens a passive data channel




- Client send an ephemeral port number via control channel

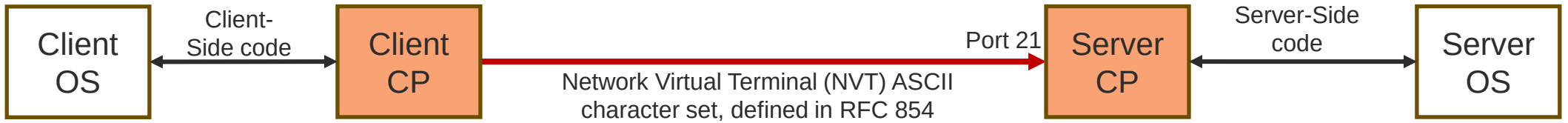



- Server opens an active data channel from port 20 to ephemeral port on client data process

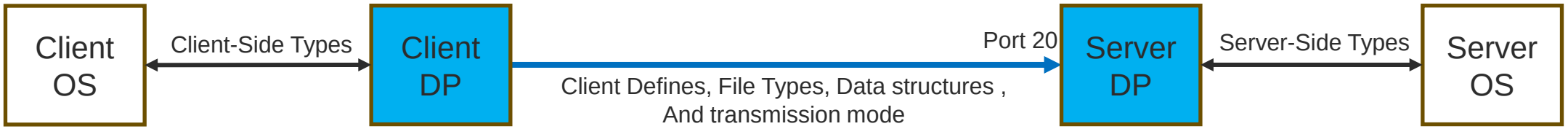


# Using Data and Control Channels

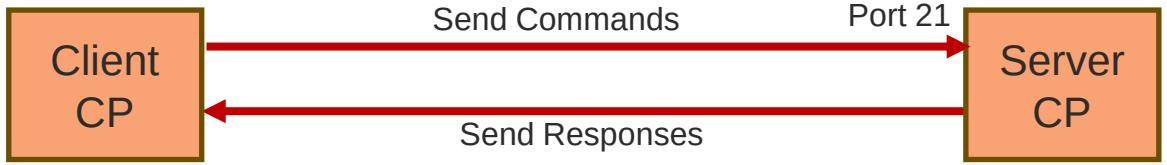
- **Control communications takes place using NVT ASCII**




- **Client defines the transmission mode along with file types and related data structure**




- **Client Sends commands and Server replies with responses**



# Connection Commands

- Connection commands are used to maintainer or disconnect session between client and server

**USER**

Used to specify user-id

**PASS**

Used to Specify Password

**ACCT**

Account Information

**REIN**

Used to Reinitialize

**QUIT**

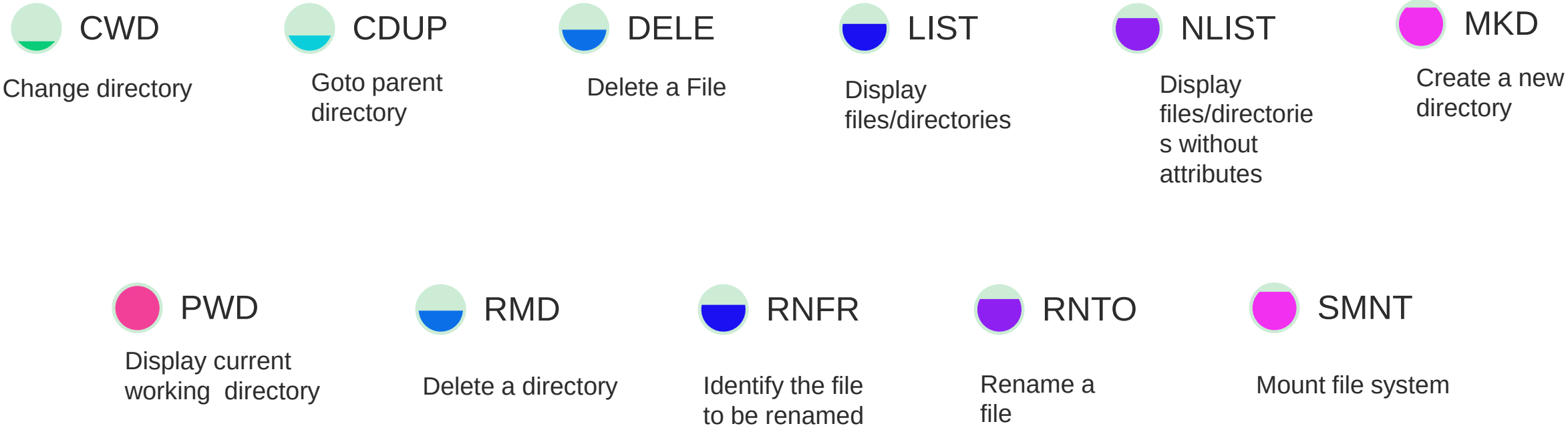
Logout of the system

**ABOR**

Abort last command

# Data Exchange Related Commands

- Commands used to manipulate data related operations



# Data type, Port and organization commands

- These commands are used to specify the data type and their structure

## Type

- Specify type of file
  - A – ASCII
  - E – EBCDIC
  - I – Image
  - N – Nonprint
  - T - Telnet

## STRU

- Specify organization of data
  - F – File
  - R – Record
  - P - Page

## Mode

- Specify Mode of Transmission
  - S – Stream
  - B – Block
  - C - Compressed

## PORT

- Specify a digit port number

## PASV

- Sever to select a port

# Data Transfer Related Commands

- Followings commands are used for data transfer related functions

**RETR**
Retrieve a file from server

**STOR**
Store a file to server

**APPE**
Same as STOR but appends data if file exists

**STOU**
Same a STOR, but uses unique file name in directory

**ALLO**
Allocated Storage at server

**REST**
Position File marker at given data point

**STAT**
Gets the status of files

- Other Commands

**HELP**
Retrieve about server

**NOOP**
Check if server is alive

**SITE**
Site Specific commands

**SYST**
Retrieve information and Server OS

# Responses

| | |
|---|---|
| **Initial Reply** | |
| 120 | Service will be ready shortly |
| 125 | Data Channel Open, DT will start shortly |
| 150 | File Statis OK, DC will open shortly |
| **Completion Replies** | |
| 200 | Command OK |
| 211 | System Status or Help Reply |
| 212 | Directory Status |
| 213 | File Status |
| 214 | Help Message |
| 215 | Naming the system type |
| 220 | Service Ready |
| 221 | Service Closing |
| 225 | DC Open |
| 226 | Closed DC |
| 227 | Entering passive mode; server sends its IP address and port number |
| 230 | User Login OK |
| 250 | Request file action OK |
| **Immediate Replies** | |
| 331 | Username OK, Password Required |
| 332 | Need account for logging |
| 350 | File action is pending, more information required |

| | |
|---|---|
| **Transient Negative Replies** | |
| 425 | Cannot open DC |
| 426 | Channel Closed; Transfer aborted |
| 450 | File Action not taken; file not available |
| 451 | Action aborted; local error |
| 452 | Action aborted; insufficient storage |
| **Negative Completion Replies** | |
| 500 | Syntax Error: recognized command |
| 501 | Syntax Error in parameters or arguments |
| 502 | Command not implemented |
| 503 | Bad sequence of commands |
| 504 | Command Parameter not implemented |
| 530 | User is not logged in |
| 532 | Need account for storing file |
| 550 | Action is not done; file not available |
| 552 | Requested Action aborted; exceeded storage quota |
| 553 | Request action no taken; file name not allowed |

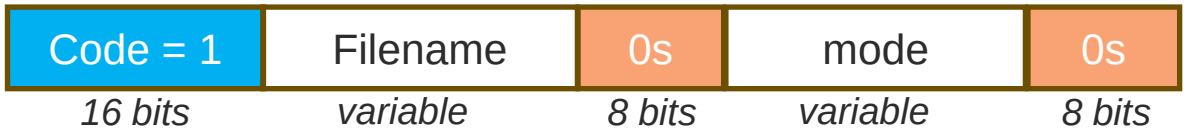
# TFTP

- **There are occasions when we need to simply copy a file without the need for all of the features of the FTP protocol.**
  - For example, when a diskless workstation or a router is booted, we need to download the bootstrap and configuration files.
  - Here we do not need all of the sophistication provided in FTP. We just need a protocol that quickly copies the files.

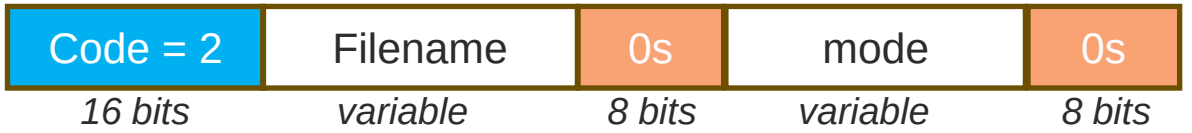
- TFTP uses the services of UDP on the well-known port 69.

- Message Categories
  - RRQ
  - WRQ
  - DATA
  - ACK
  - Error

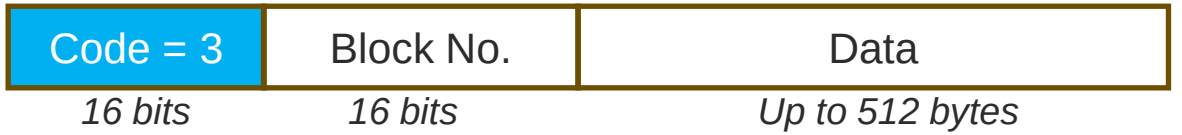
*RRQ Message*



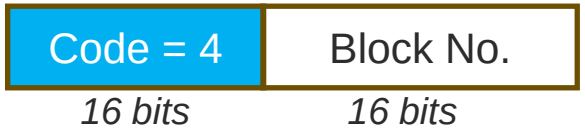

*WRQ Message*




*Data Format*




*ACK Message*



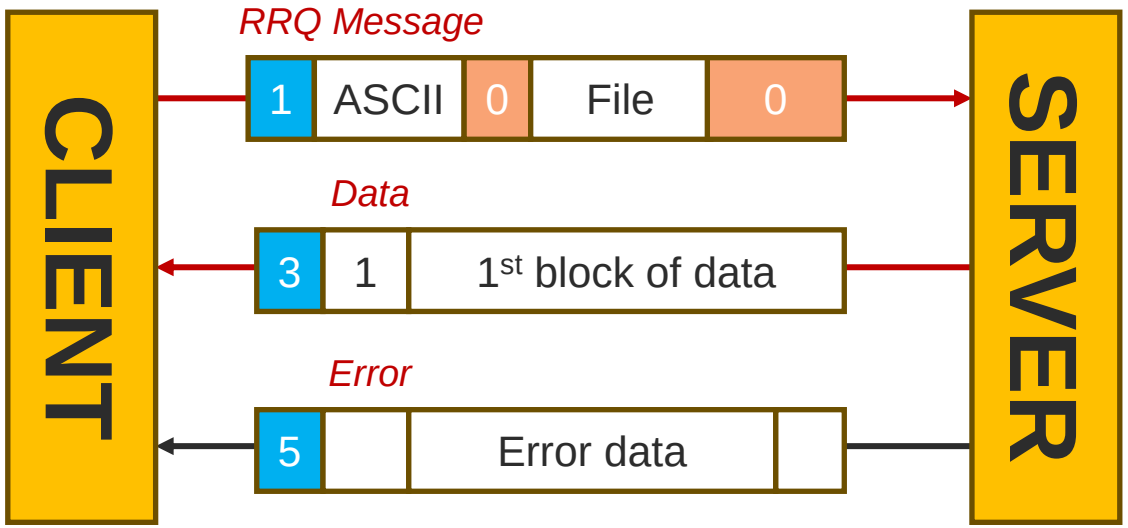

*Error Message*

# TFTP Error Messages

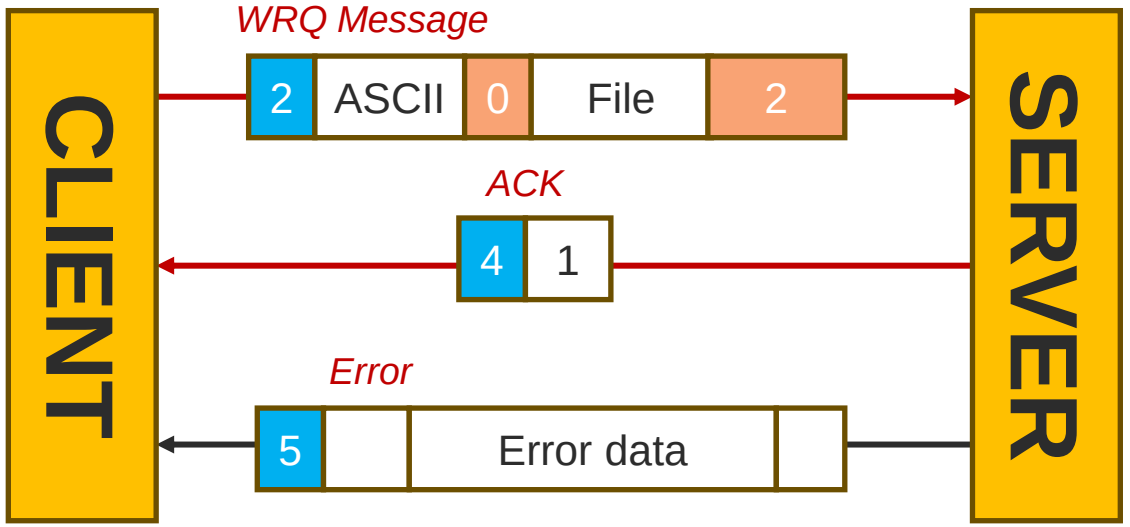
| | |
|---|---|
| 0 | No defined |
| 1 | File Not found |
| 2 | Access Violation |
| 3 | Disk Full or Allocation Exceeded quota |
| 4 | Illegal Operation |
| 5 | Unknown port number |
| 6 | File Already Exists |
| 7 | No Such User |

# Connection Procedure

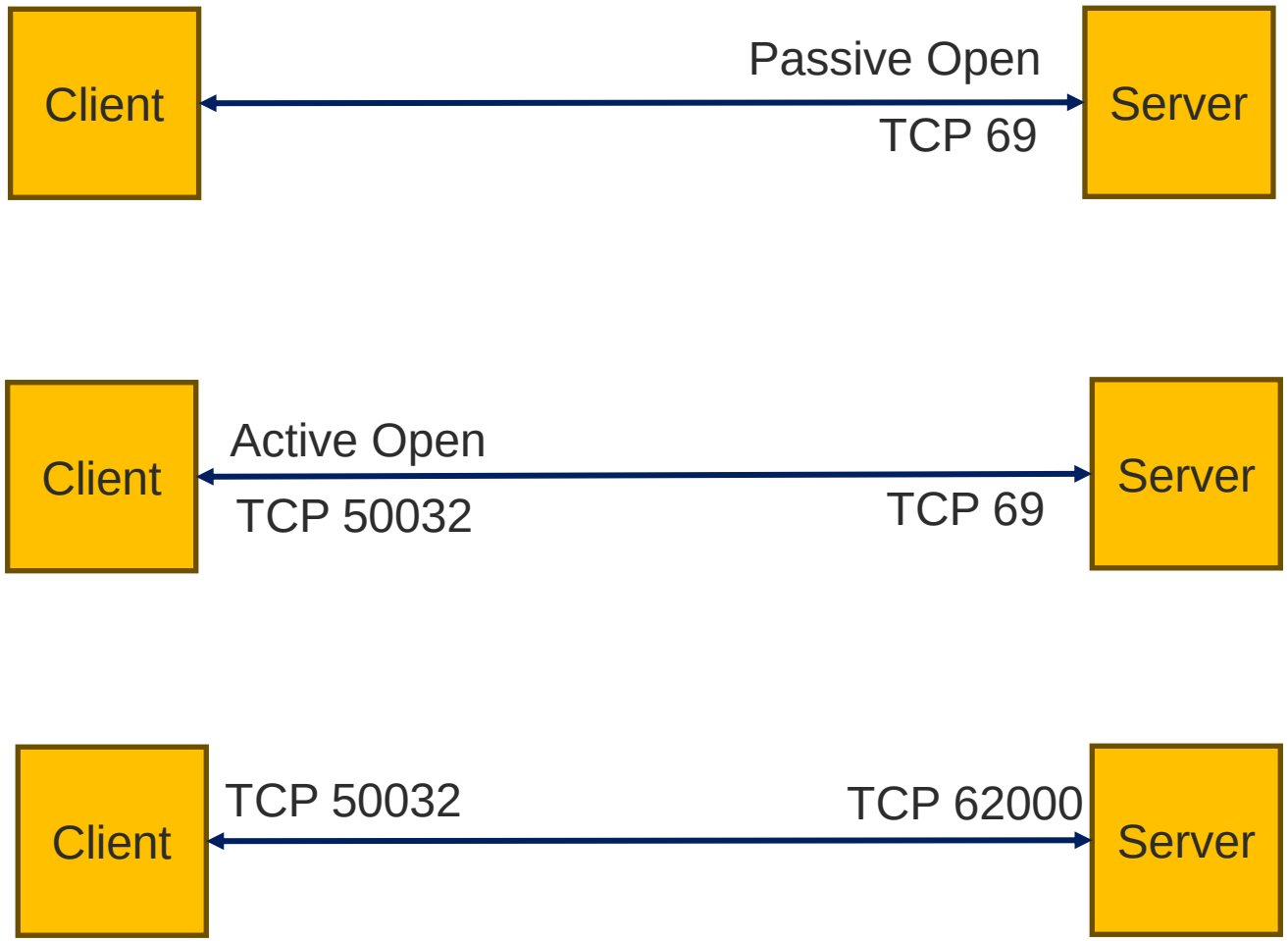
- Reading File from server




- Storing File on server



# UDP port numbers used by TFTP



IT601 – System and Network Administration

# File Transfer and Sharing Services

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# File Sharing and Transfer Services

- Following services shall be covered in this module.
  - NFS
  - SMB/CIFS
  - FTP/TFTP/SFTP Services
  - SCP Services

IT601 – System and Network Administration

# NFS Service

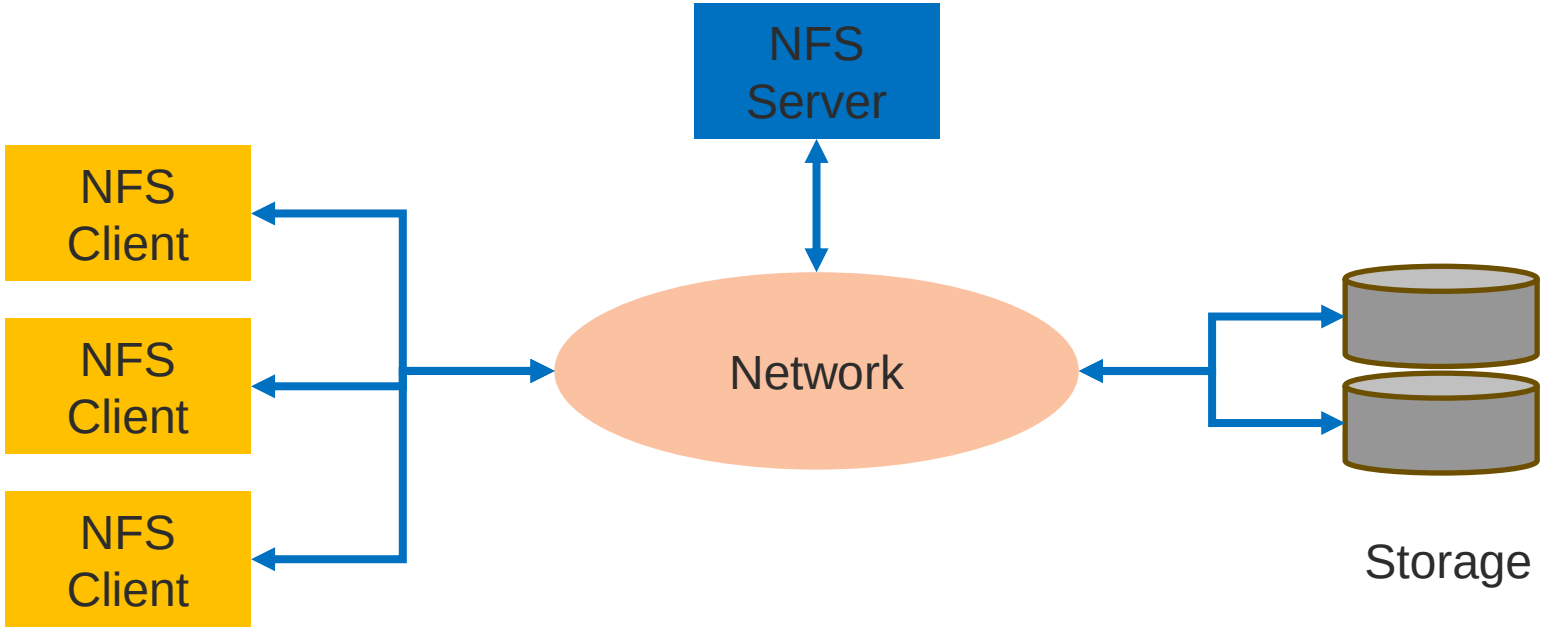
Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Introduction

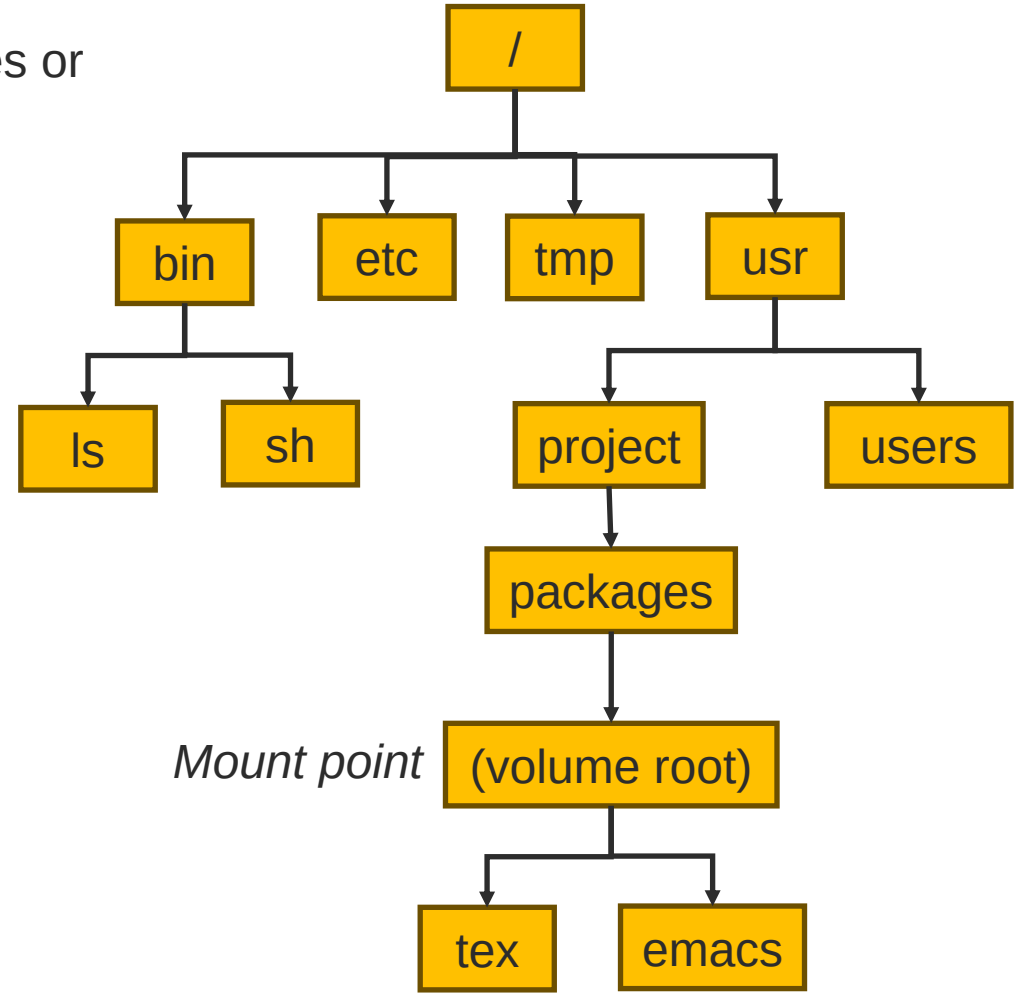
- Network File System (NFS) is a distributed file system protocol developed by Sun Microsystems (Sun) in 1984.
  - It allows a user on a client computer to access files over a computer network much like local storage is accessed.
  - It uses the Open Network Computing Remote Procedure Call (ONC RPC) system.
  - NFS is an open IETF standard defined in a Request for Comments (RFC), allowing anyone to implement the protocol.



# Review of Linux File System

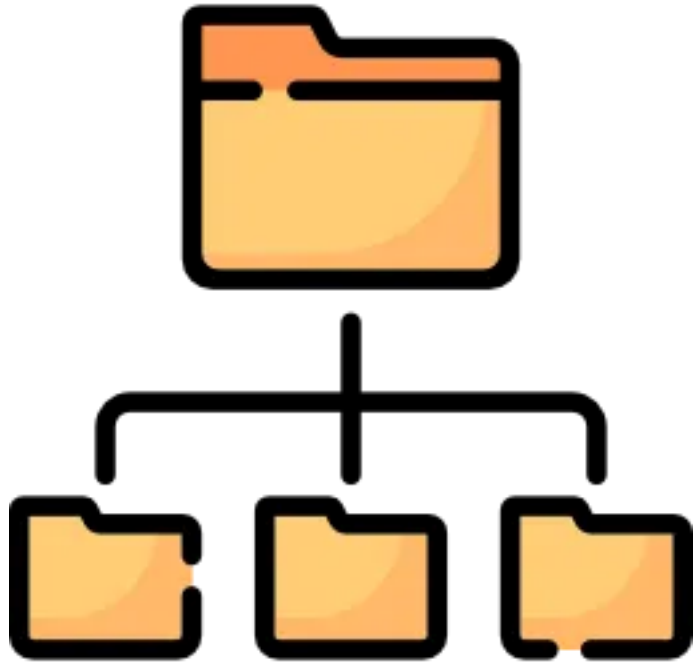
- **A volume is a set of directories and files; a host's file tree is the set of directories and files visible to processes on a given host.**
  - File trees are built by grafting volumes from different volumes or from network servers.
  - The graft operation is the privileged mount system call, and each volume is a filesystem.

- **Mounting volumes**
  - mount (dirs, volume)
    - dirs : directory pathname
    - volume: device specifier or network volume
    - volume root contents become visible at pathname dirs



# Filesystems

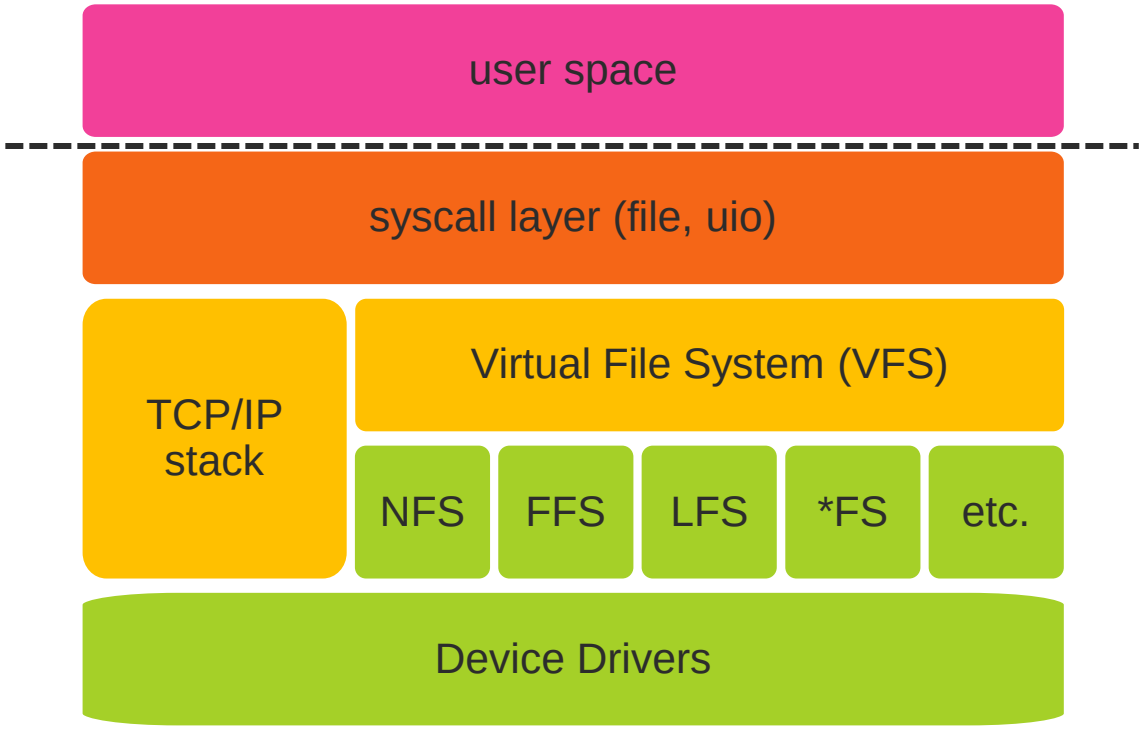
- **Each file volume (filesystem) has a type, determined by its disk layout or the network protocol used to access it.**
  - ufs (ffs), lfs, nfs, rfs, cdfs, etc.
  - Filesystems are administered independently.
- **Modern systems also include “logical” pseudo-filesystems in the naming tree, accessible through the file syscalls.**
  - procfs: the /proc filesystem allows access to process internals.
  - mfs: the memory file system is a memory-based scratch store.
- **Processes access filesystems through common system calls.**



# Virtual File System (VFS)

- **Sun Microsystems introduced the virtual file system interface in 1985 to accommodate diverse filesystem types cleanly.**
  - VFS allows diverse specific file systems to coexist in a file tree, isolating all FS-dependencies in pluggable filesystem modules.

- VFS was an internal kernel restructuring with no effect on the syscall interface.
- Incorporates object-oriented concepts like a generic procedural interface with multiple implementations.
- Based on abstract objects with dynamic method binding by type in C.
- Other abstract interfaces in the kernel: device drivers, file objects, executable files, memory objects.



# Inodes and vnodes

- **The operating system (OS) has three kinds of data structures for files:**

### File table entry

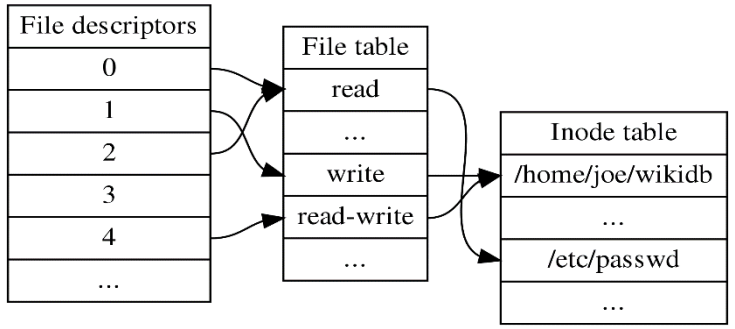
- For each file descriptor[1] for a user there is file table entry.
- It contains information like the current lseek pointer, and a pointer to a vnode.
- When you start your program, the OS has three file table entries one each for stdin, stdout, stderr.
- Each time open() is called, a new file table entry is created in the OS.

### vnode

- There is one of these for each physical file that has been opened.
- It contains a pointer to the file's inode, the file's size, etc.

### inode

- There is one of these for each file on disk.
- It contains all the information returned by stat().

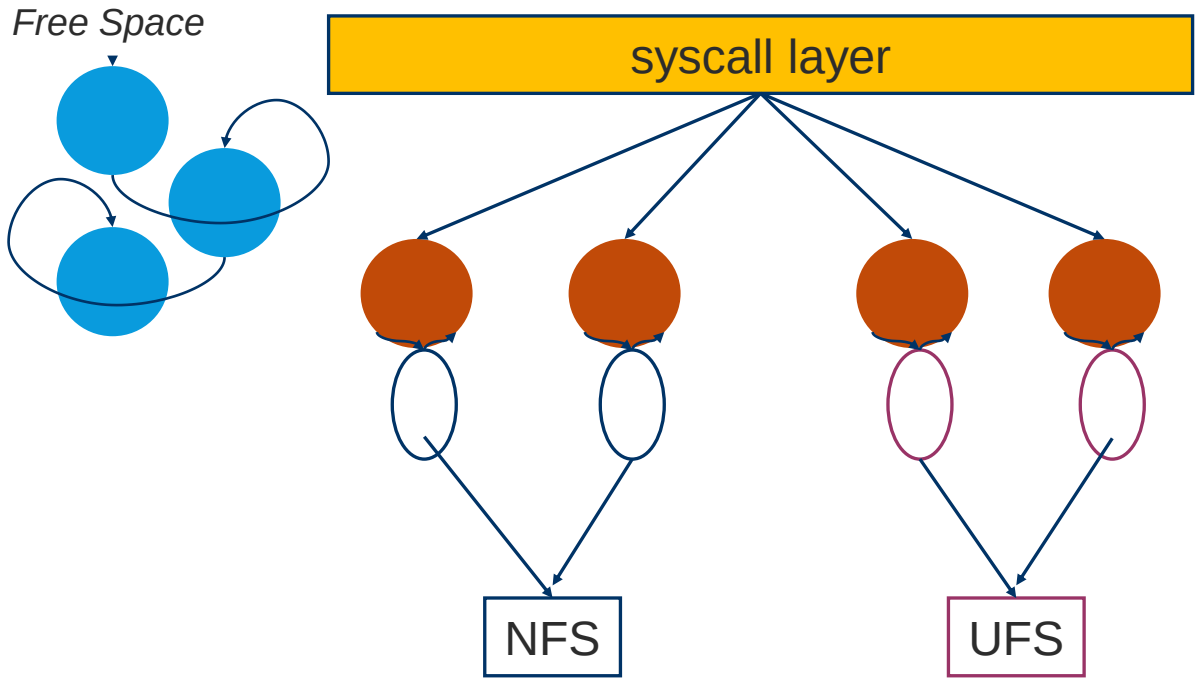



[1]A file descriptor (FD, less frequently fildes) is a process-unique identifier (handle) for a file or other input/output resource, such as a pipe or network socket.

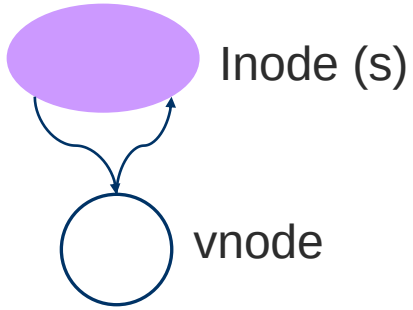
# Inodes and vnodes

- **In the VFS framework, every file or directory in active use is represented by a vnode object in kernel memory.**

  - Each specific file system maintains a cache of its resident vnodes.
  - Each vnode has a standard file attributes struct.
  - Generic vnode points at filesystem-specific struct (e.g., inode, rnode), seen only by the filesystem.
  - Vnode operations are macros that vector to filesystem-specific procedures.



# Vnode Operations and Attributes




**vnode attributes**

- Type
- Mode
- nlink
- ownerID
- ownerGroup
- fsID
- fileID
- fileSize
- accessTime
- modifyTime
- generationNo

**Generic Operations**

- Vop_getattr(vattr)
- vop_setattr (vattr)
- vhold()
- vholdrele()

**Directory Operations**

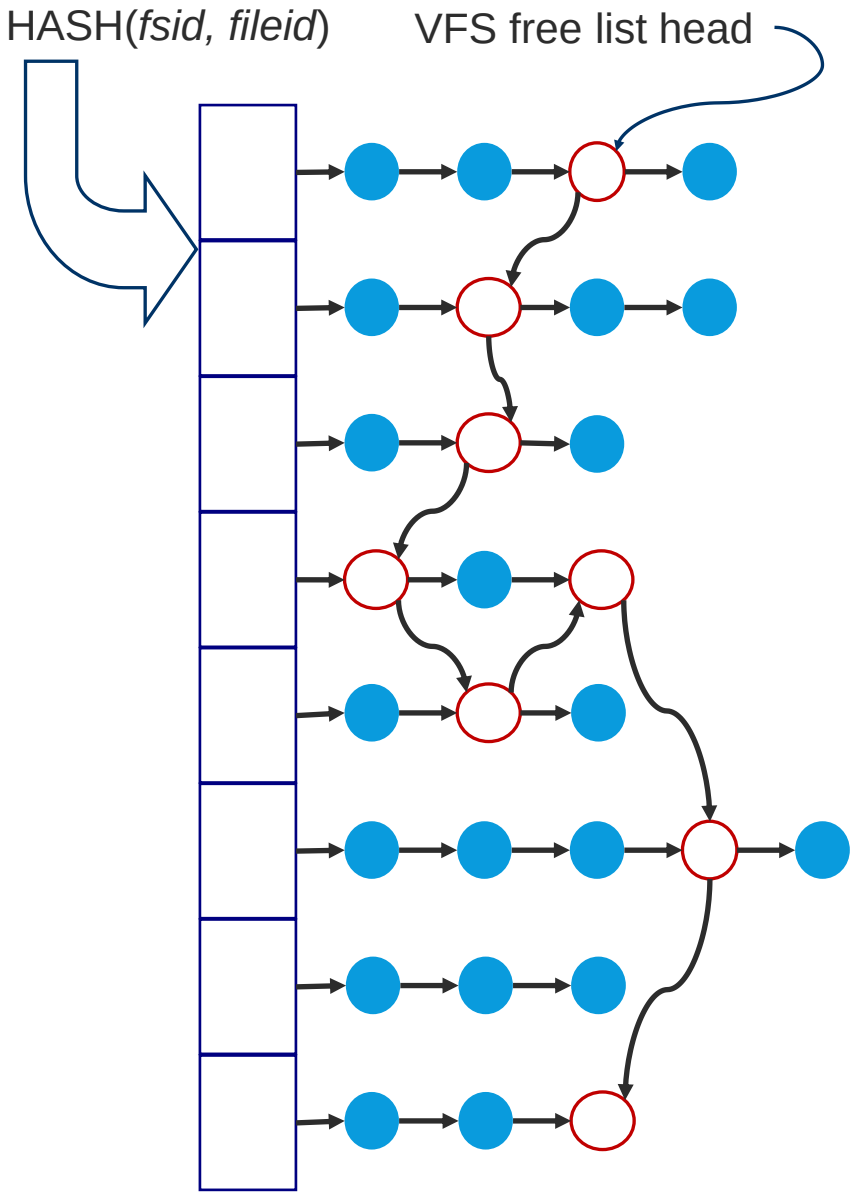
- vop_lookup (OUT vpp, name)
- vop_create (OUT vpp, name, vattr)
- vop_remove (vp, name)
- vop_link (vp, name)
- vop_rename (vp, name, tdvp, tvp, name)
- vop_mkdir (OUT vpp, name, vattr)
- vop_rmdir (vp, name)
- vop_symlink (OUT vpp, name, vattr, contents)
- vop_readdir (uio, cookie)
- vop_readlink (uio)

**Files Operations**

- vop_getpages (page**, count, offset)
- vop_putpages (page**, count, sync, offset)
- vop_fsync ()

# vnode/inode Cache

- Active vnodes are reference- counted by the structures that hold pointers to them.
  - system open file table
  - process current directory
  - file system mount points
  - etc.
- Each specific file system maintains its own hash of vnodes (BSD).
  - specific FS handles initialization
  - free list is maintained by VFS
- Methods
  - vget(vp): reclaim cached inactive vnode from VFS free list
  - vref(vp): increment reference count on an active vnode
  - vrele(vp): release reference count on a vnode
  - vgone(vp): vnode is no longer valid (file is removed)



# Pathname Traversal

- When a pathname is passed as an argument to a system call, the syscall layer must “convert it to a vnode”.
  - Pathname traversal is a sequence of vop_lookup calls to descend the tree to the named file or directory.
    - open(“/tmp/zot”)
    - vp = get vnode for / (rootdir)
    - vp->vop_lookup(&cvp, “tmp”);
    - vp = cvp;
    - vp->vop_lookup(&cvp, “zot”);
- Issues in pathname traversal
  - crossing mount points
  - obtaining root vnode (or current dir)
  - finding resident vnodes in memory
  - caching name->vnode translations
  - symbolic (soft) links
  - disk implementation of directories
  - locking/referencing to handle races
  - with name create and delete operations

# NFS Protocol

- NFS is a network protocol layered above TCP/IP.
  - Original implementations  use UDP transport for low overhead.
  - Maximum IP packet size was increased to match FS block size, to allow send/receive of entire file blocks.
  - Some implementations of NFS also use TCP as a transport.

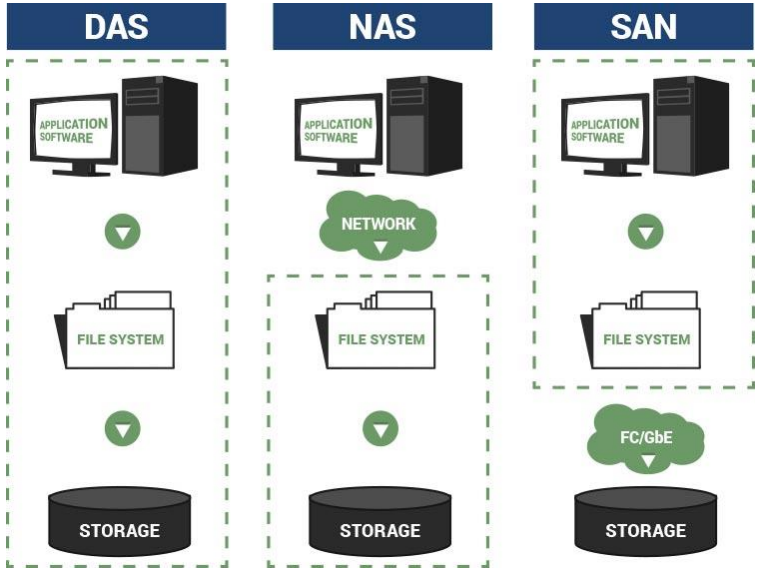
- The NFS protocol is a set of message formats and types.
  - Client issues a request message for a service operation.
  - Server performs requested operation and returns a reply message with status and (perhaps) requested data.

# Network Block Storage

- One approach to scalable storage is to attach raw block storage to a network.
  - abstraction: OS addresses storage by <volume, sector>.
    - Examples are iSCSI, Petal, FC: access through souped-up device driver
  - dedicated Storage Area Network or general-purpose network
    - Examples are FibreChannel vs. Ethernet
  - shared access with scalable bandwidth and capacity
  - volume-based administrative tools
    - backup, volume replication, remote sharing
- Called “raw” or “block”, “storage volumes” or just “SAN”.

# NAS and SAN

- **Network Attached Storage (NAS)**
  - In NAS, one or more dedicated file server or storage devices are made available in a LAN. Therefore, the transfer of data, particularly for backup, still takes place over the existing LAN.




- **Storage Area Network (SAN)**
  - SAN is a network which provides access to consolidated, block-level data storage. SANs are primarily used to access data storage devices like disk arrays and tape libraries from servers so that the devices appear to the operating system as direct-attached storage.
- **There has been some debate over whether NAS and SAN is good for enterprises.**
  - Network-Attached Storage has been the dominant approach to shared storage since NFS.
  - NAS == NFS or CIFS: named files over Ethernet/Internet.
  - Network Appliance Known as filers
- **FibreChannel (FC) SAN is considered to be fundamentally faster way to access shared storage.**
  - No indirection through a file server
  - Lower overhead on clients
  - Network is better/faster (if not cheaper) and dedicated/trusted
  - Brocade, HP, Emulex are some big players.

# NAS and SAN: What is better?

- FC is a high-end technology incorporating NIC enhancements to reduce host overhead but bogged down in interoperability problems.

- Ethernet is getting faster faster than FC such as gigabit, 10-gigabit, + smarter NICs, + smarter/faster switches

- The choice of network is fundamentally orthogonal to storage service design.
  - Well, almost: flow control, RDMA, user-level access

- The fundamental questions are really about abstractions. For example, shared raw volume vs. shared file volume vs. private disks

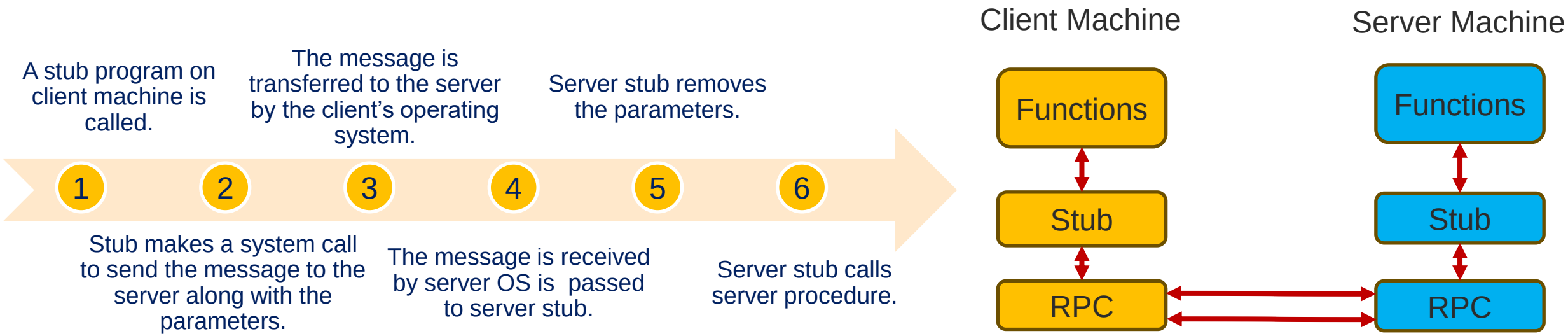
# Abstractions for storage

- Relational database (IBM and Oracle) such as tables, transactions, query language
- File system  such as hierarchical name space with ACLs
- Block storage such as SAN, Petal, RAID-in-a-box (e.g., EMC)
- Object storage where object == file, with a flat name space: NASD, DDS
- Persistent objects such as pointer structures, requires transactions: OODB, ObjectStore

# Remote Procedure Call (RPC)

- A remote procedure call is an inter-process communication technique used for client-server-based applications.
  - A client sends request message that the RPC translates and sends to the server. The request may be a procedure or a function call to a remote server.
  - When the server receives the request, it sends the required response back to the client. The client is blocked while the server is processing the call and only resumed execution after the server is finished.

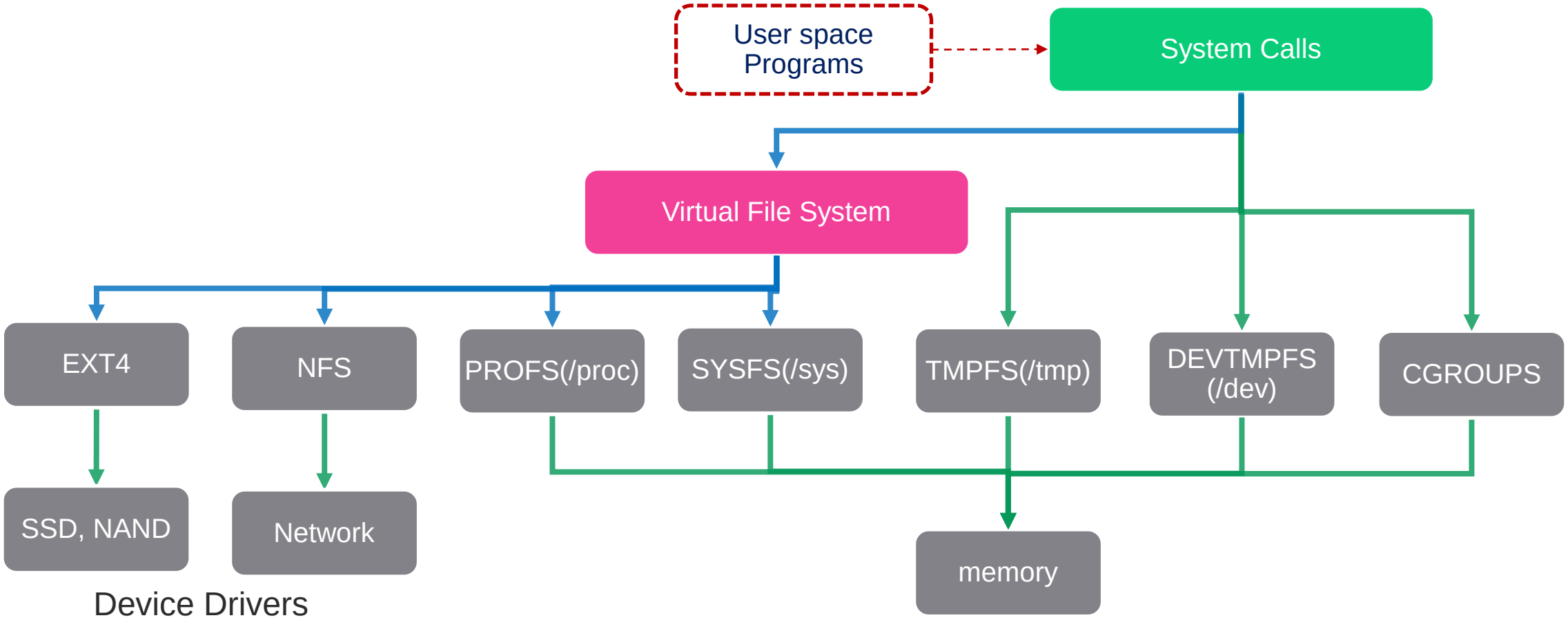
- The sequence of events involved in RPC are :-



# Virtual Filesystem

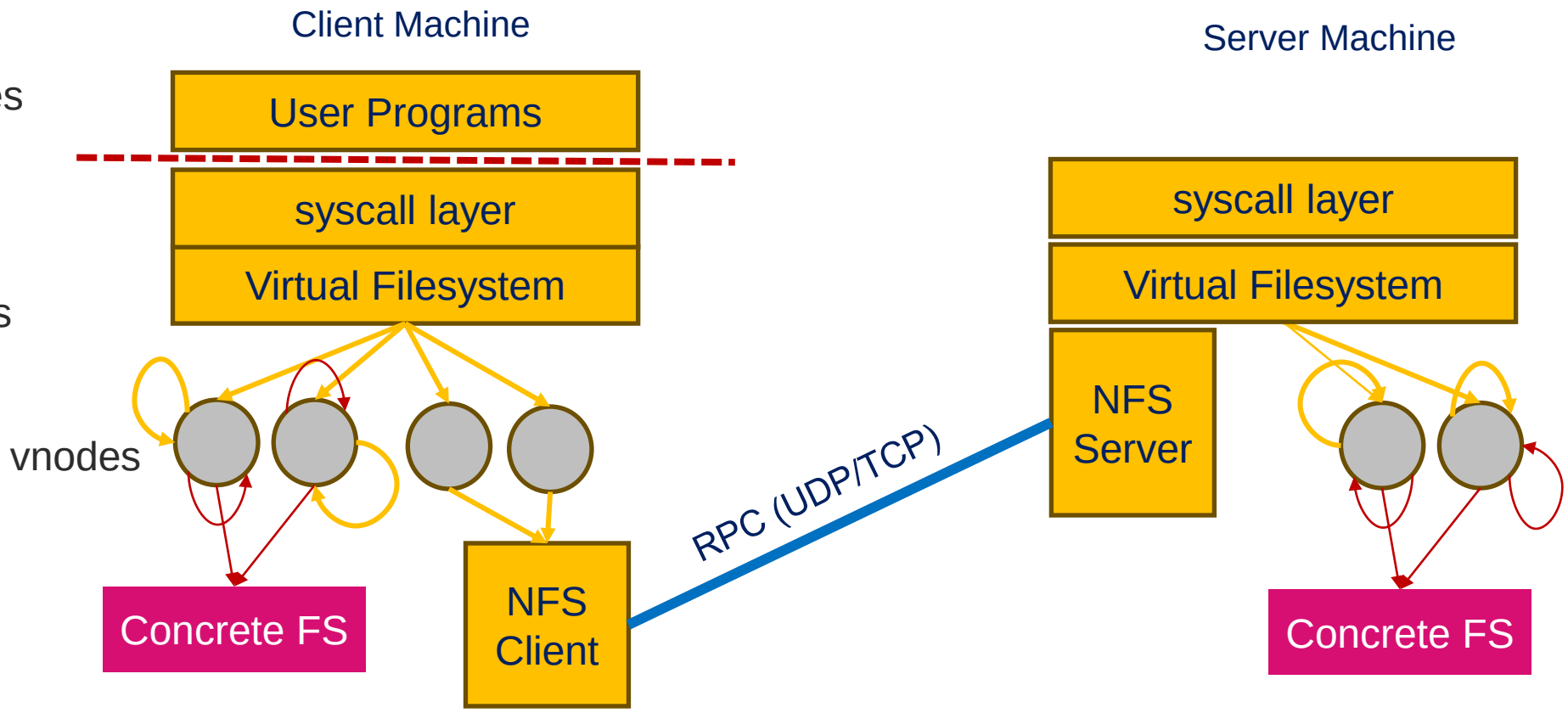
- A virtual file system (VFS) or virtual filesystem switch is an abstract layer on top of a more concrete file system.

  It allows client applications to access different types of concrete file systems in a uniform way.



# Network File System (NFS)

- VFS abstracts the concrete FS
  - VFS builds the vnodes corresponding to inodes of concrete FS
  - Users interreacts with VFS using system calls
  - NFS client and server implement the RPC
  - RPC communicates of TCP/UDP ports
  - NFS implements stubs
  - The nfsnode holds client state needed to interact with the server to operate on the file.



# NFS file reference

- In NFS, the reference is a file handle or fhandle, a token/ticket whose value is determined by the server.
  - Includes all information needed to identify the file/object on the server, and find it quickly.

| volume ID | inode # | generation # |
|---|---|---|

# Stateless Behavior of NFS

- **Stateful behavior vs Stateless Services**

- **A classical NFS server maintains no in-memory hard state.**
  - The only hard state is the stable file system image on disk.
  - no record of clients or open files
  - no implicit arguments to requests
    - no server-maintained file offsets: read and write requests must explicitly transmit the byte offset for each operation.
  - no write-back caching on the server
  - no record of recently processed requests

- **Stateless behavior of NFS makes failure recovery simple and efficient.**

# Stateless Behavior of NFS

## ➢ Recovery in Stateless NFS

### Server Restart

- No Need to rebuild in-memory state of server
- Client reestablishes TCP Connection
- Client retransmits pending requests.

### Connectionless

- Classical NDS uses which is connectionless protocol
- Server failure is transparent to the client; no connection to break or reestablish.
- A crashed server is indistinguishable from a slow server.

### Error Masking

- RPC masks network errors by retransmitting a request after an adaptive timeout.
- A dropped packet is indistinguishable from a crashed server.

## ➢ Although simple but Stateless behavior has problems

### Constrained interface

- Recovery-by-retransmission constrains the server interface.
- RPC/UDP has execute-at-least-once semantics (“send and pray”), which compromises performance and correctness.

### Update Operations

- Update operations are disk-limited.
- Updates must commit synchronously at the server.

### Inconsistency

- NFS cannot (quite) preserve local single-copy semantics.
- Files may be removed while they are open on the client.
- Server cannot help in client cache consistency.

# Stateless Behavior of NFS : Potential Remedies

## Retransmissions and Idempotency

- **Hope for the best and smooth over non-idempotent requests.**
  - map ENOENT and EEXIST to ESUCCESS.
- **Use TCP or some other transport protocol that produces reliable, in-order delivery.**
  - However, higher overhead...and we still need sessions.
- **Implement an execute-at-most once RPC transport.**
  - TCP-like features (sequence numbers)...and sessions.
- **Keep a *retransmission cache* on the server.**
  - Remember recent request IDs and their results, and just resend the result.

## Synchronous Writes

- **Delay the response until convenient for the server.**
  - NFS *write-gathering* optimizations for clustered writes (similar to *group commit* in databases).
  - Relies on write-behind from NFS I/O daemons (*iods*).
- **Throw hardware at it: non-volatile memory (NVRAM)**
  - Battery-backed RAM or UPS (uninterruptible power supply).
  - Use as an operation log (Network Appliance WAFL) or as a non-volatile disk write buffer (Legato).
- **Replicate server and buffer in memory (e.g., MIT Harp).**

## File Cache Consistency

- **Timestamp invalidation (NFS).**
  - Timestamp each cache entry, and periodically query the server: “has this file changed since time **t**?”; invalidate cache if stale.
- **Callback invalidation (AFS, Sprite, Spritely NFS).**
  - Request notification (callback) from the server if the file changes; invalidate cache and/or disable caching on callback.
- **Leases (NQ-NFS) Later: distributed shared memory**

# NFS V3

- **NFS V3 sidesteps the synchronous write problem by adding a new asynchronous write operation.**
  - Server may reply to client as soon as it accepts the write, before executing/committing it.
    - If the server fails, it may discard any subset of the accepted but uncommitted writes.
  - Client holds asynchronously written data in its cache and reissues the writes if the server fails and restarts.
    - When is it safe for the client to discard its buffered writes?
    - How can the client tell if the server has failed?
- **NFS V3 adds a new commit operation to go with async-write.**
  - Client may issue a commit for a file byte range at any time.
  - Server must execute all covered uncommitted writes before replying to the commit.
  - When the client receives the reply, it may safely discard any buffered writes covered by the commit.
  - Server returns a verifier with every reply to an async write or commit request.
  - The verifier is just an integer that is guaranteed to change if the server restarts, and to never change back.
  - What if the client crashes?

# NFSv4 Features

- Several new features have been added to NSF v4.

## State-fullness

- The usage information of an object by an NFSv4 client is maintained by the server.
- opening, locking, reading, and writing, carry state information that notify the server of the intentions on the object by the client.
- The server can return information to a client about other clients having intentions on the same object.
- The use of a persistent open on the server avoids some situations where an NFS version 2 or 3 client could become locked out of a file while writing to it.

## Byte-range locking and share modes

- It provides support for byte-range locking and share modes as part of the base protocol.
- Locking in NFSv4 is lease-based, which requires the NFSv4 client to maintain contact with the server to preserve open and lock state owned by the client.

## Compound request format

- An NFSv4 client can combine several simple operations (for example, LOOKUP, OPEN, and READ) into a single request to the server.
- The single request allows NFSv4 to perform a complex operation in one network exchange.

# NFSv4

| improved security | standardizes the representation of string data | Unification | RPC Over TCP |
|---|---|---|---|
| ➢ The NFSv4 protocol specifies improved security mechanisms over the others required by previous protocol versions.<br>➢ provides support for Kerberos 5 authentication and data protection in addition to the traditional AUTH_SYS security.<br>➢ The security API used by NFSv4 allows for easy addition of new security mechanisms in the future. | ➢ The NFSv4 protocol standardizes the representation of string data.<br>➢ All string data used by the protocol is represented in UTF-8 as it crosses the network.<br>➢ User and group information is passed between the client and server in string form, not as numeric values as in previous versions. | ➢ The NFSv4 protocol combines the separate component protocols of previous NFS versions into a single protocol specification.<br>➢ The single point of contact for the NFSv4 protocol allows for better compatibility with network fire walls. | ➢ NFS Version 4 requires support of RPC over streaming network transport protocols such as TCP.<br>➢ The NFSv4 support provided by IBM i uses TCP exclusively. |

# IT601 – System and Network Administration

# Networks

Arif Husen

**Department of Computer Science and Information Technology,**

**Virtual University of Pakistan**

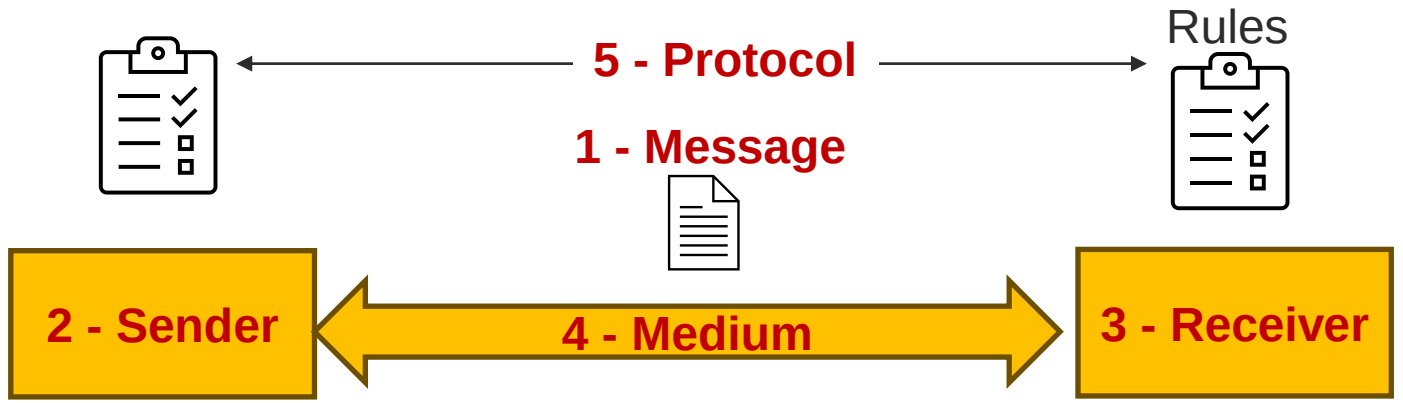
# Data communications

- Data communications are used to exchange data between two devices via some form of transmission medium such as a wire cable or wireless.

1. Delivery → Correct destination
2. Accuracy → Accurate data
3. Timelines → Real-time transmission
4. Jitter → Uneven delay

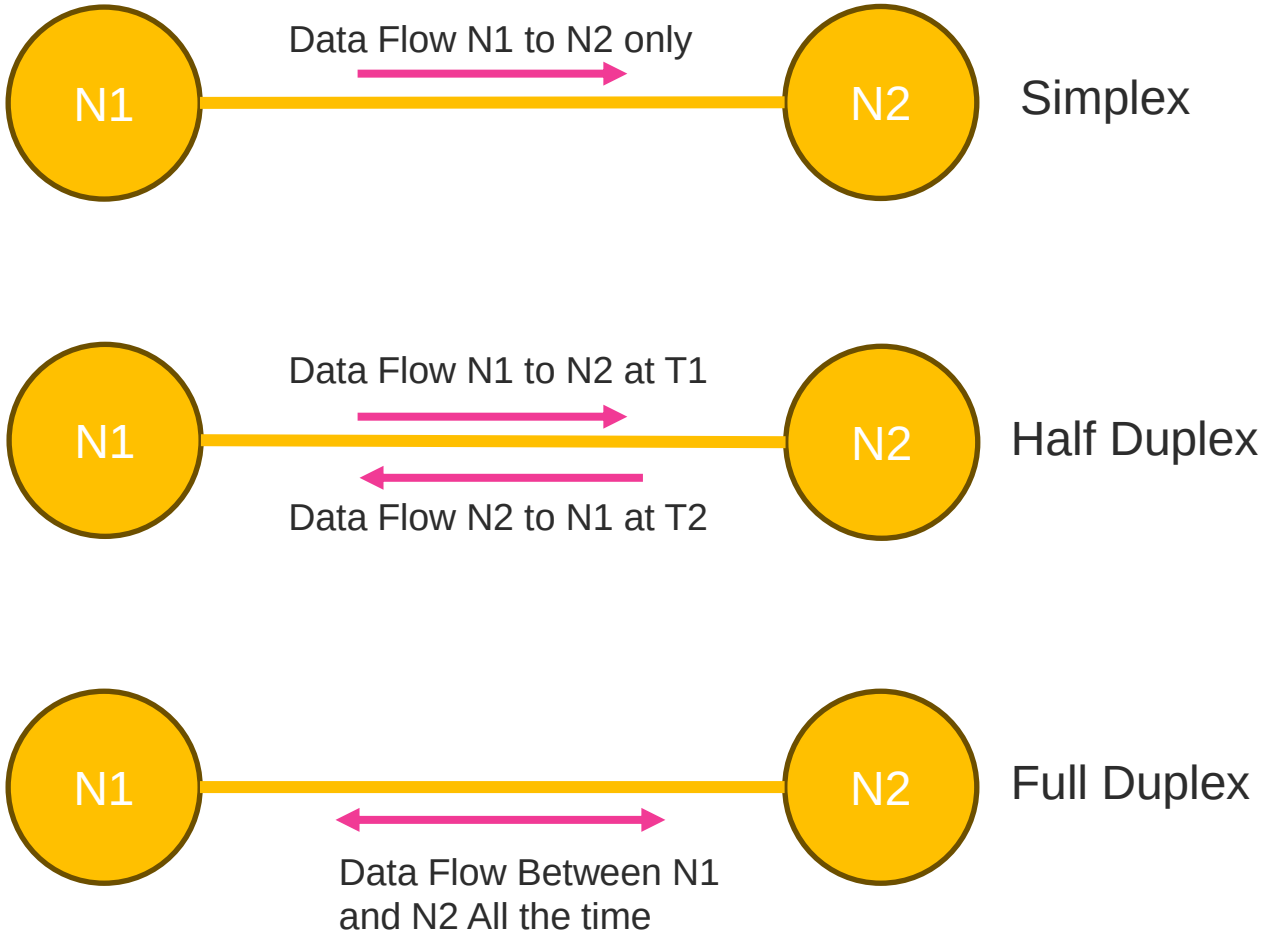
# Five components of data communication

- Message
- Sender
- Receiver
- Medium
- Protocol



# Data types and flows

- **Data Types**
  - Text
  - Numbers
  - Images
  - Audio
  - Video
- **Data flow**
  - Simplex
  - Half-duplex
  - Full-duplex

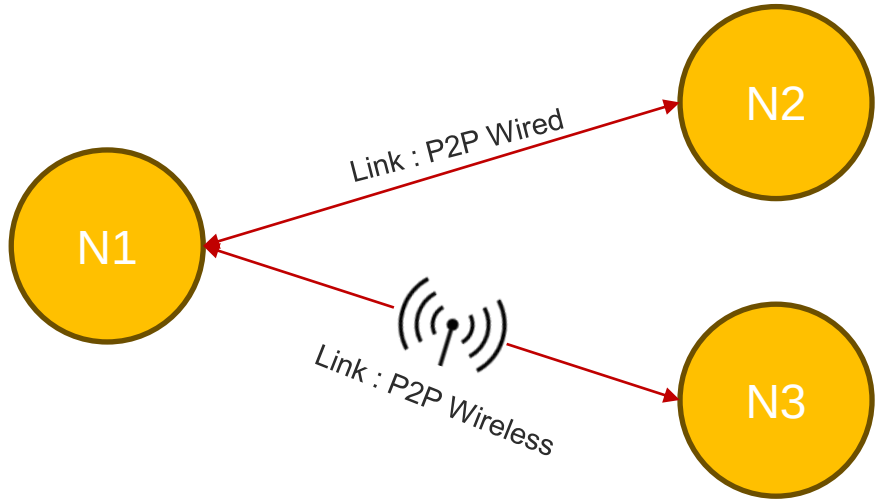


# 1-2 NETWORKS

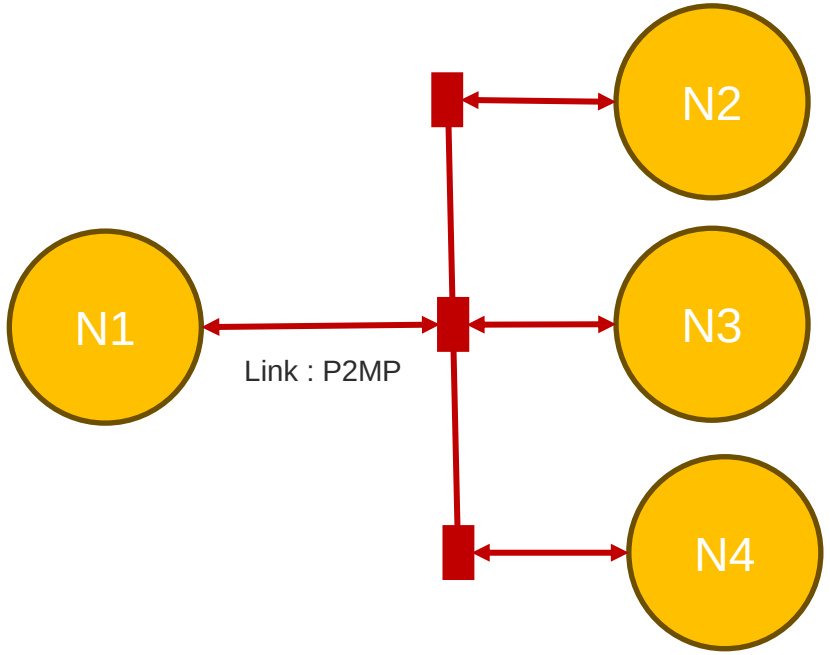
- A network is a set of devices (nodes) connected by communication links.
  - A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.

# Types of connections

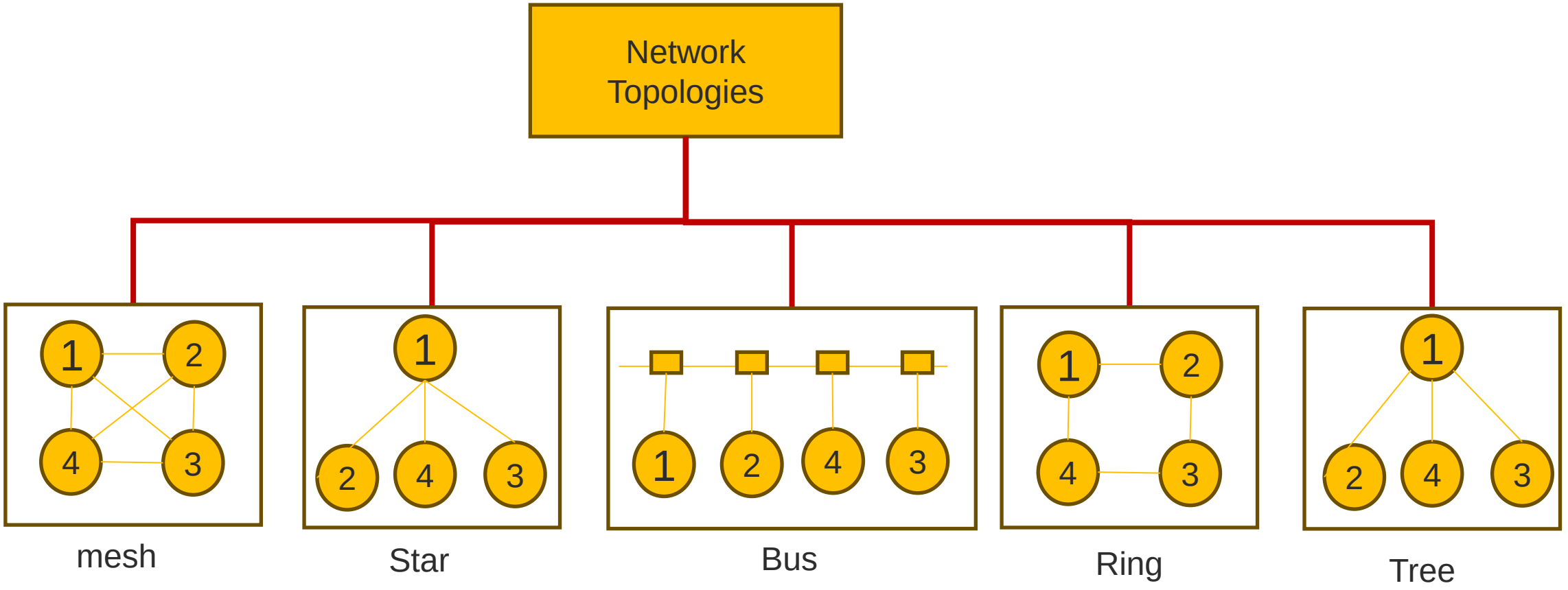
- Point to point
  - A dedicated link is provided between two devices




- Multipoint
  - More than two specific devices share a single link



# Physical Topologies

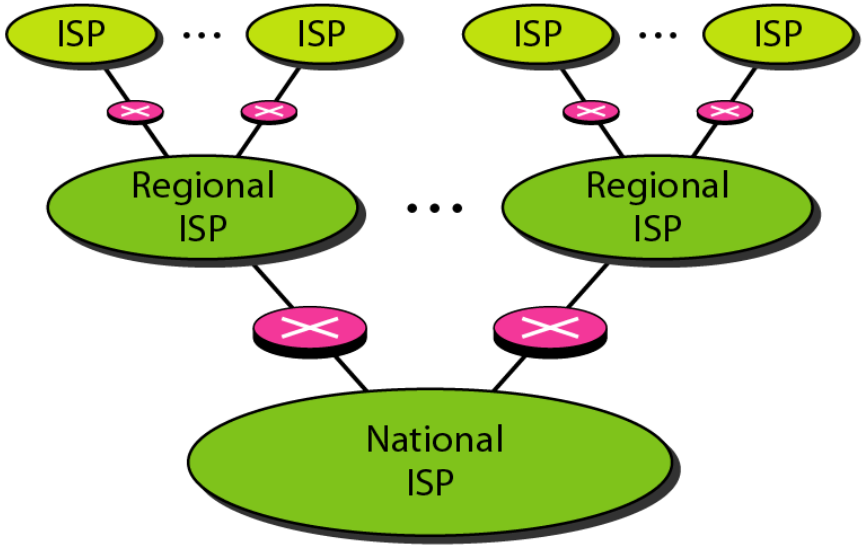


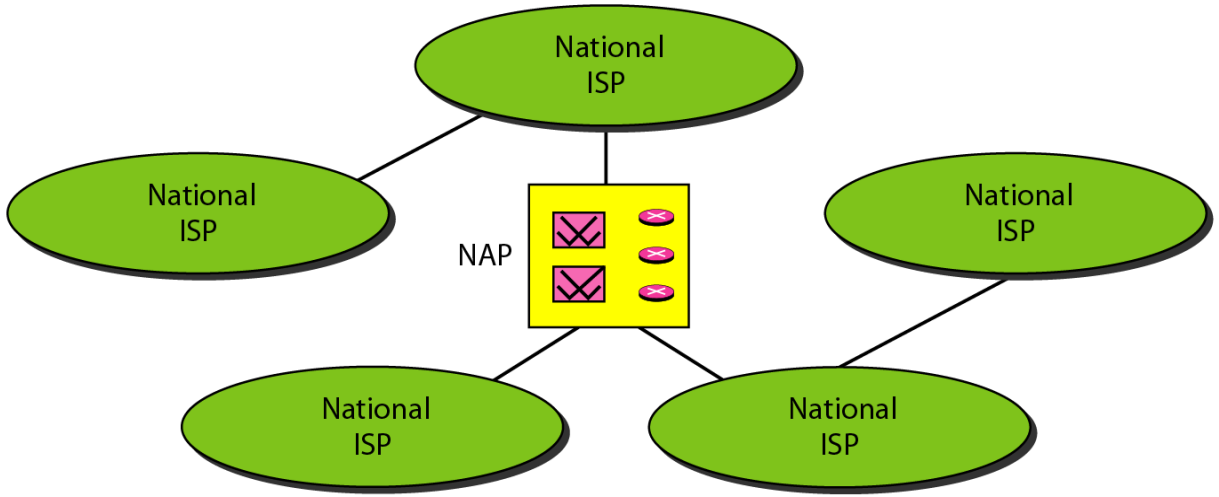
# Categories of Networks

- Local Area Network (LAN)
- Wireless Local Area Network (WLAN)
- Metropolitan Area Network (MAN)
- Wide Area Network (WAN)
- Virtual Private Networks
  - L3 VPN
  - L2 VPN

# 1-3 THE INTERNET

- The Internet has changed many aspects of our daily lives.
- It has affected the way we do business as well as the way we spend our leisure time.
- The Internet is a communication system that has brought a wealth of information to our fingertips and organized it for our use.



# 1-4 PROTOCOLS AND STANDARDS

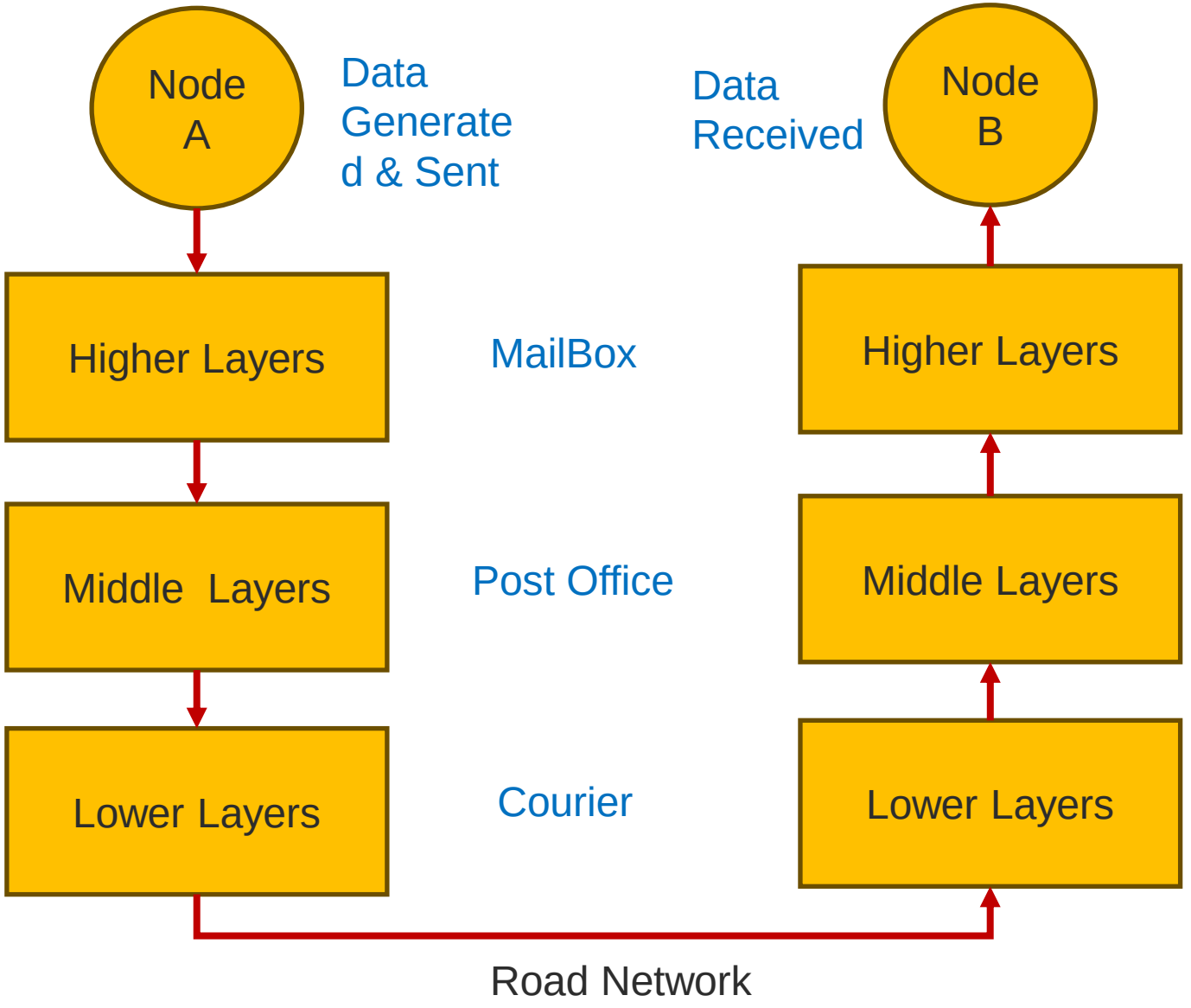
- Protocol is synonymous with rule. Standards are agreed-upon rules.

  - Protocols
    - Syntax → format of the data
    - Semantics → meaning of each section
    - Timing → when data should be sent and how fast.
  - Standards
    - De facto → by fact (not approved as a standard)
    - De jure → by Law (approved)
  - Standards Organizations
    - International Organization for Standardization (ISO)
    - International Telecommunication Union - Telecommunication Standards (ITU-T)
    - American National Standards Institute (ANSI)
    - Institute of Electrical and Electronics Engineers (IEEE)
    - Electronic Industries Association (EIA)

# LAYERED TASKS

- A network model is a layered architecture
  - Task broken into subtasks
  - Implemented separately in layers in stack
  - Functions need in both systems
  - Peer layers communicate

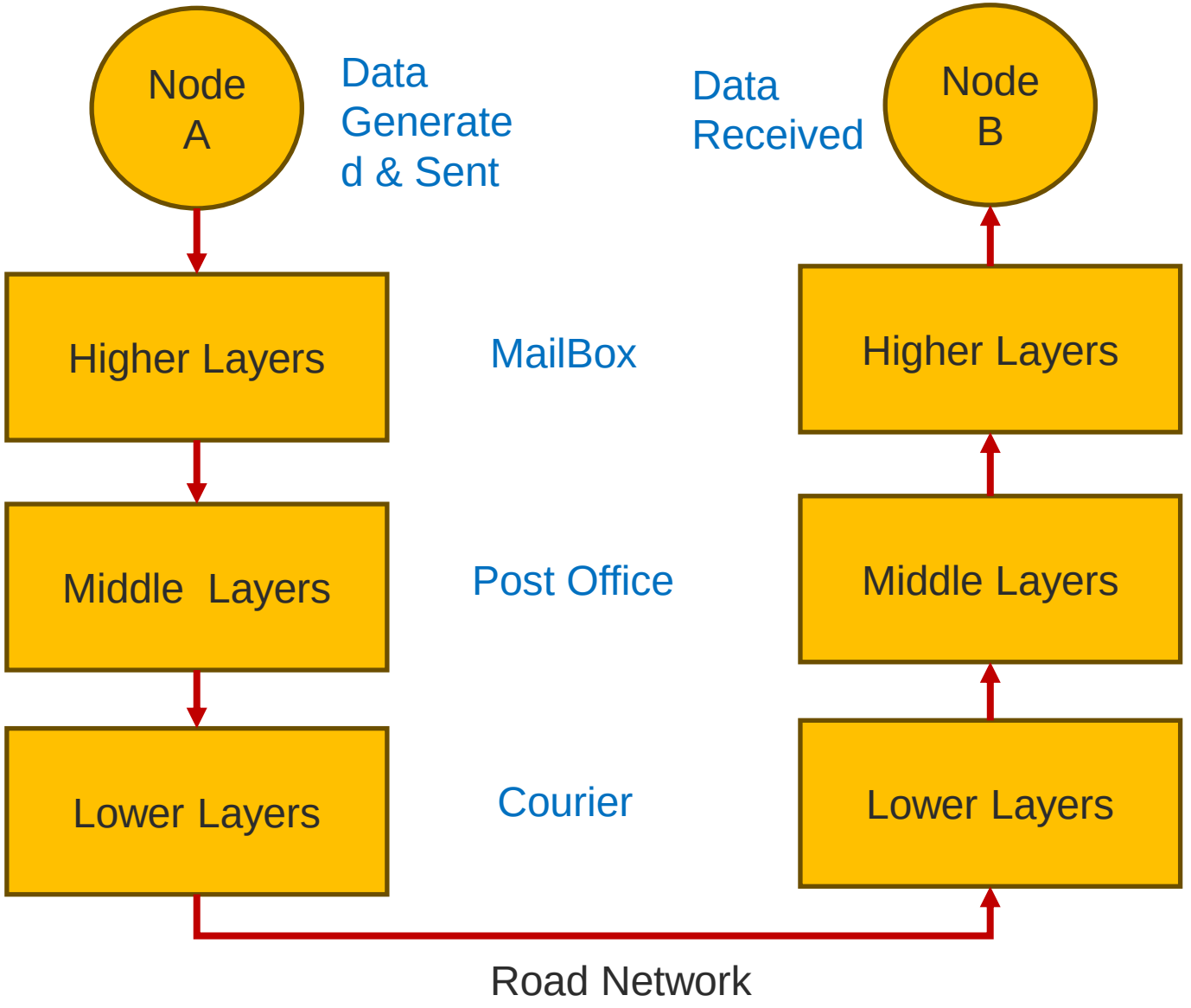
- Protocol:
  - A set of rules that governs data communication
  - It represents an agreement between the communicating devices



# LAYERED TASKS

- A network model is a layered architecture
  - Task broken into subtasks
  - Implemented separately in layers in stack
  - Functions need in both systems
  - Peer layers communicate

- Protocol:
  - A set of rules that governs data communication
  - It represents an agreement between the communicating devices



# IT601 – System and Network Administration

# OSI Model

Arif Husen

**Department of Computer Science and Information Technology,**

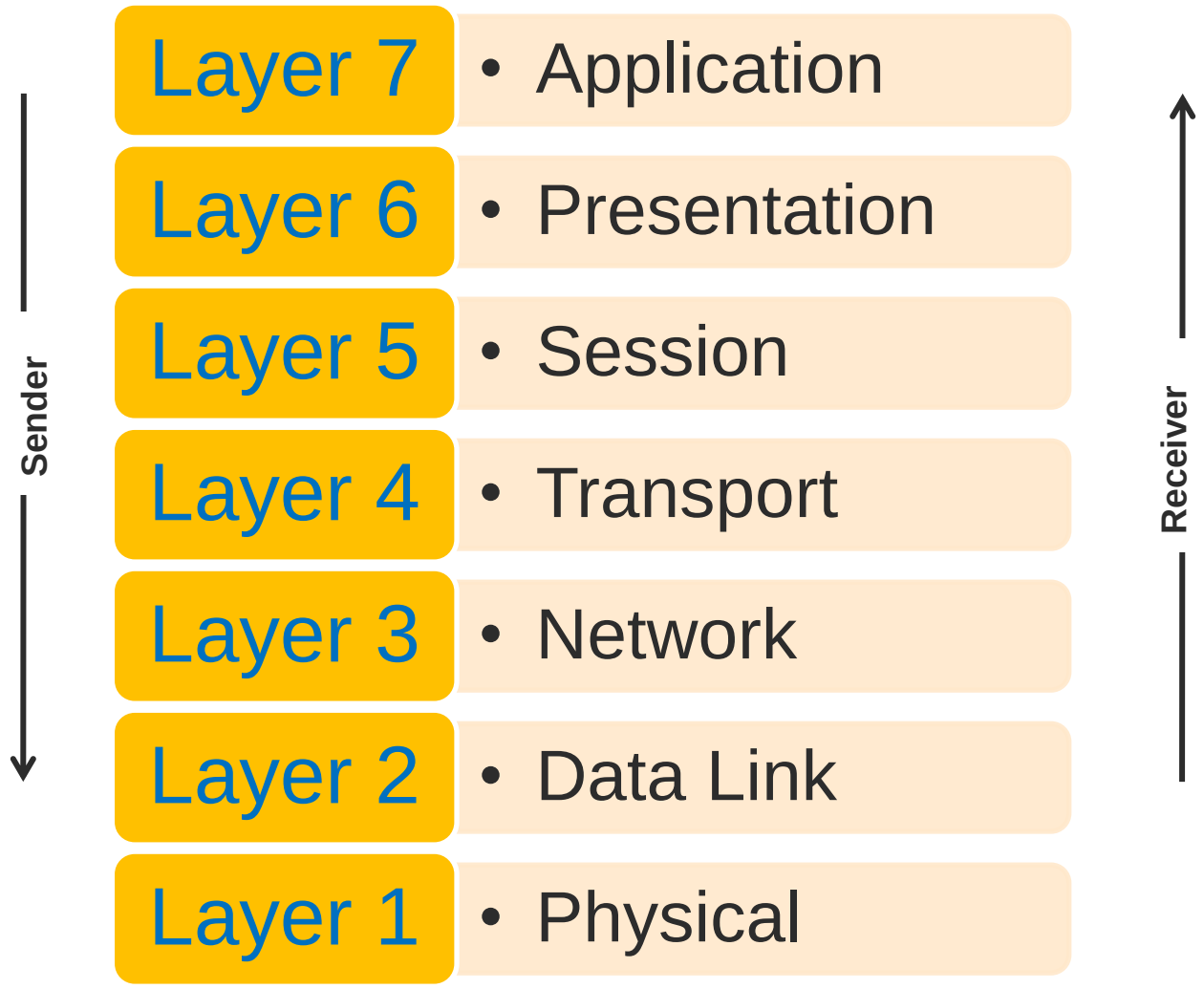
**Virtual University of Pakistan**

# THE OSI MODEL

- Established in 1947, the International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement on international standards.

- An ISO is the Open Systems Interconnection (OSI) model is the standard that covers all aspects of network communications from ISO. It was first introduced in the late 1970s.

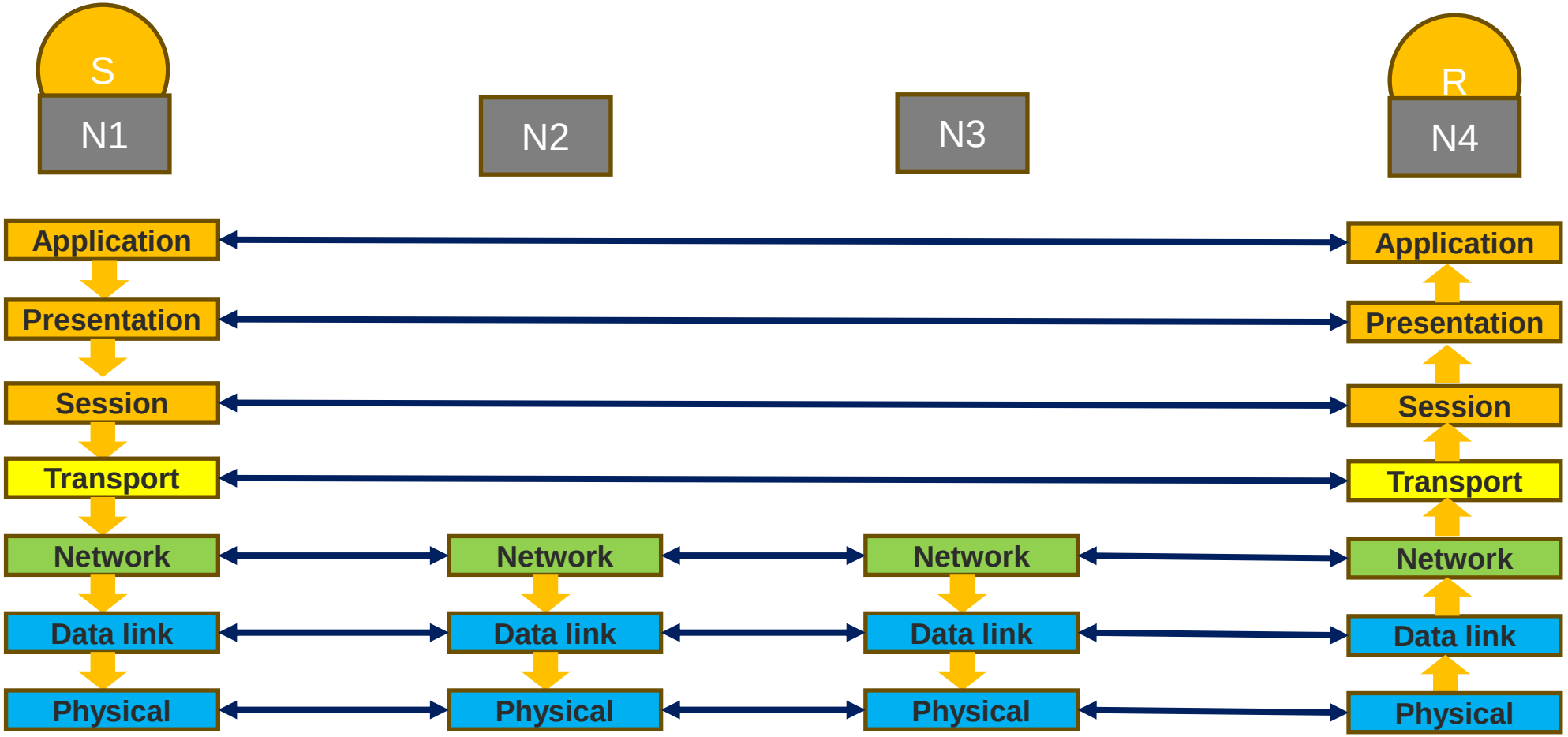
- Layered Architecture
  - A layered model
  - Each layer performs a subset of the required communication functions
  - Each layer relies on the next lower layer to perform more primitive functions
  - Each layer provides services to the next higher layer
  - Changes in one layer should not require changes in other layers
  - The processes on each machine at a given layer are called peer-to-peer process



# PEER – TO – PEER PROCESS

- Communication must move downward through the layers on the sending device, over the communication channel, and upward to the receiving device
- Each layer in the sending device adds its own information to the message it receives from the layer just above it and passes the whole package to the layer just below it
- At the receiving device, the message is unwrapped layer by layer, with each process receiving and removing the data meant for it
- The passing of the data and network information down through the layers of the sending device and backup through the layers of the receiving device is made possible by *interface* between each pair of adjacent layers
- Interface defines what information and services a layer must provide for the layer above it.

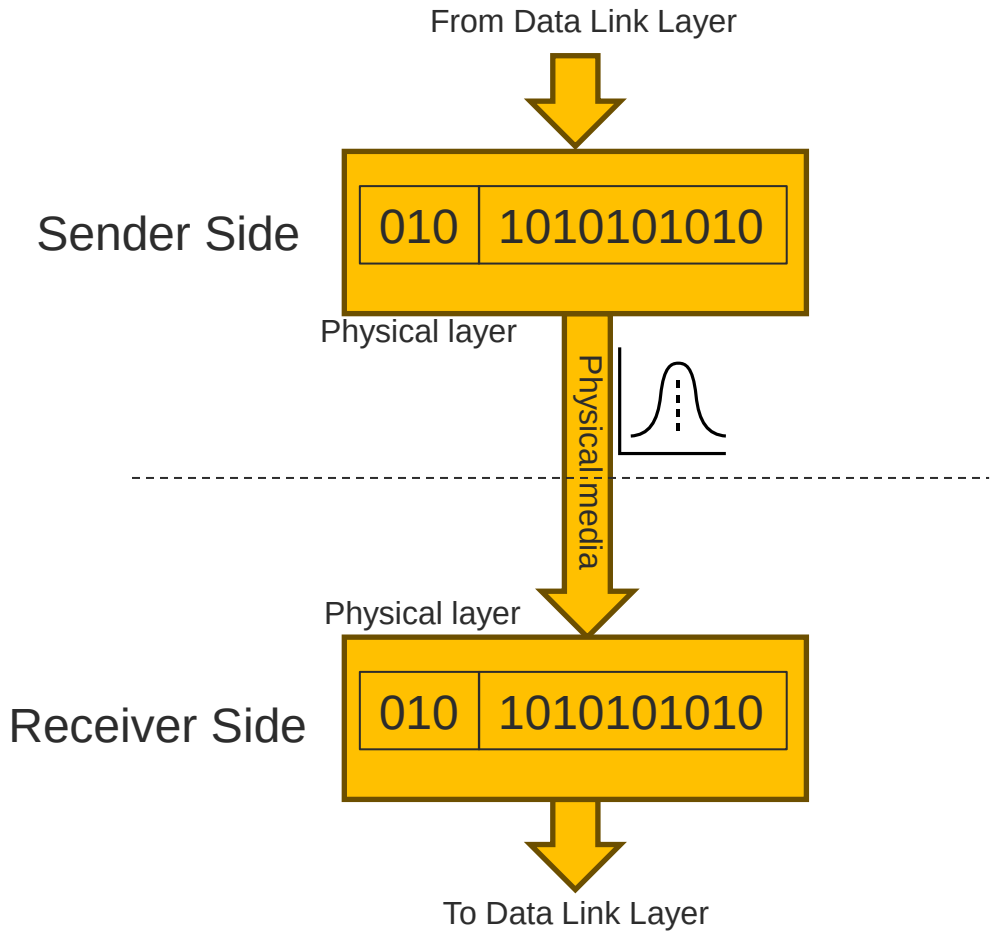
# The interaction between layers in the OSI model



# Physical Layer

- **The physical layer is responsible for movements of individual bits from one hop (node) to the next.**

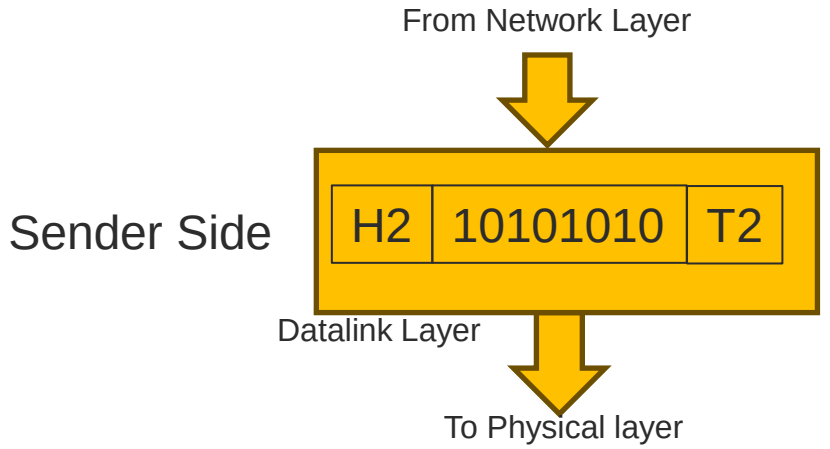
- Function
  - Physical characteristics of interfaces and media
  - Representation of bits
  - Data rate
  - Synchronization of bits
  - Line configuration (point-to-point or multipoint)
  - Physical topology (mesh, star, ring or bus)
  - Transmission mode ( simplex, half-duplex or duplex)

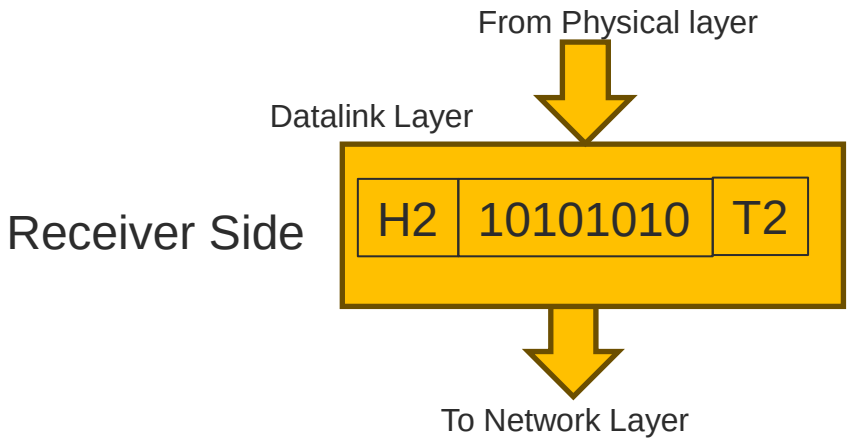


# Data Link Layer

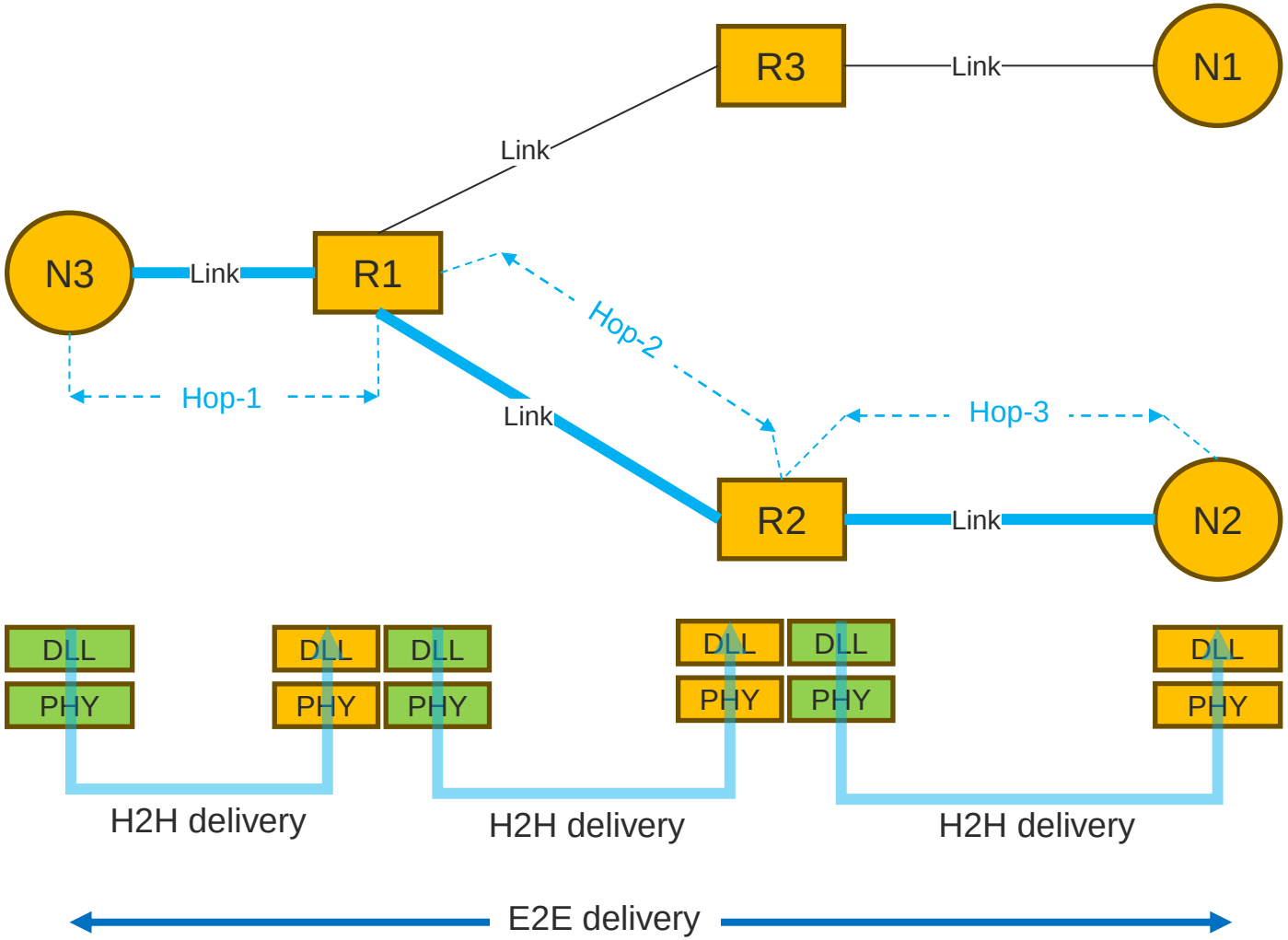
- **The data link layer is responsible for moving frames from one hop (node) to the next.**

  - Functions
    - Framing
    - Physical addressing
    - Flow control
    - Error control
    - Access control



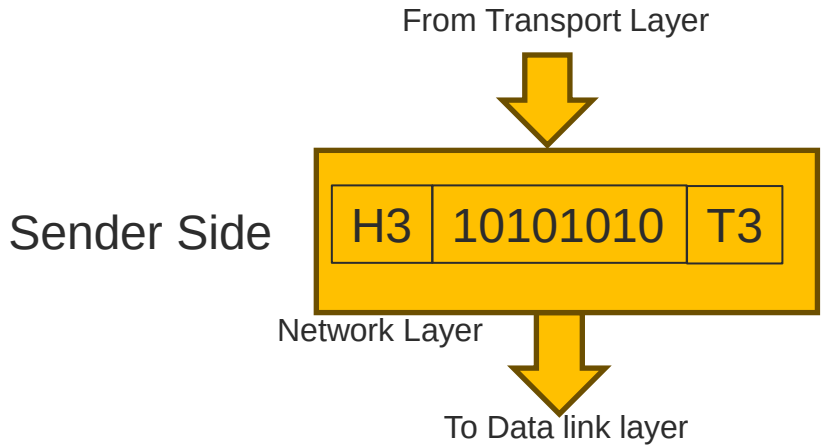



# Hop-to-hop delivery

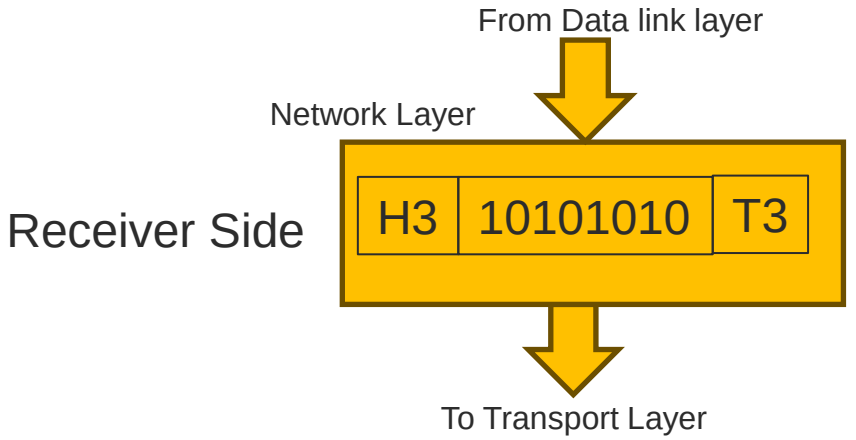


# Network Layer

**The network layer is responsible for the delivery of individual packets from the source host to the destination host.**




- Source-to-destination delivery
- Responsible from the delivery of packets from the original source to the final destination
- Functions
  - Logical addressing
  - routing

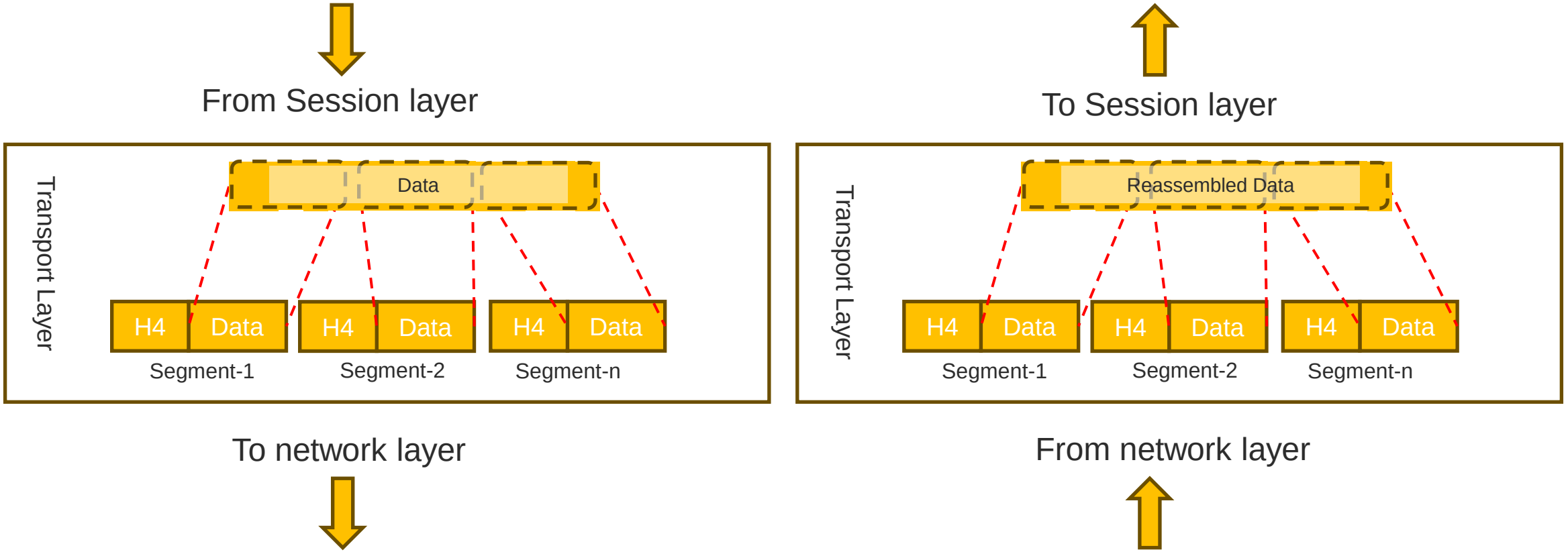


# Transport Layer

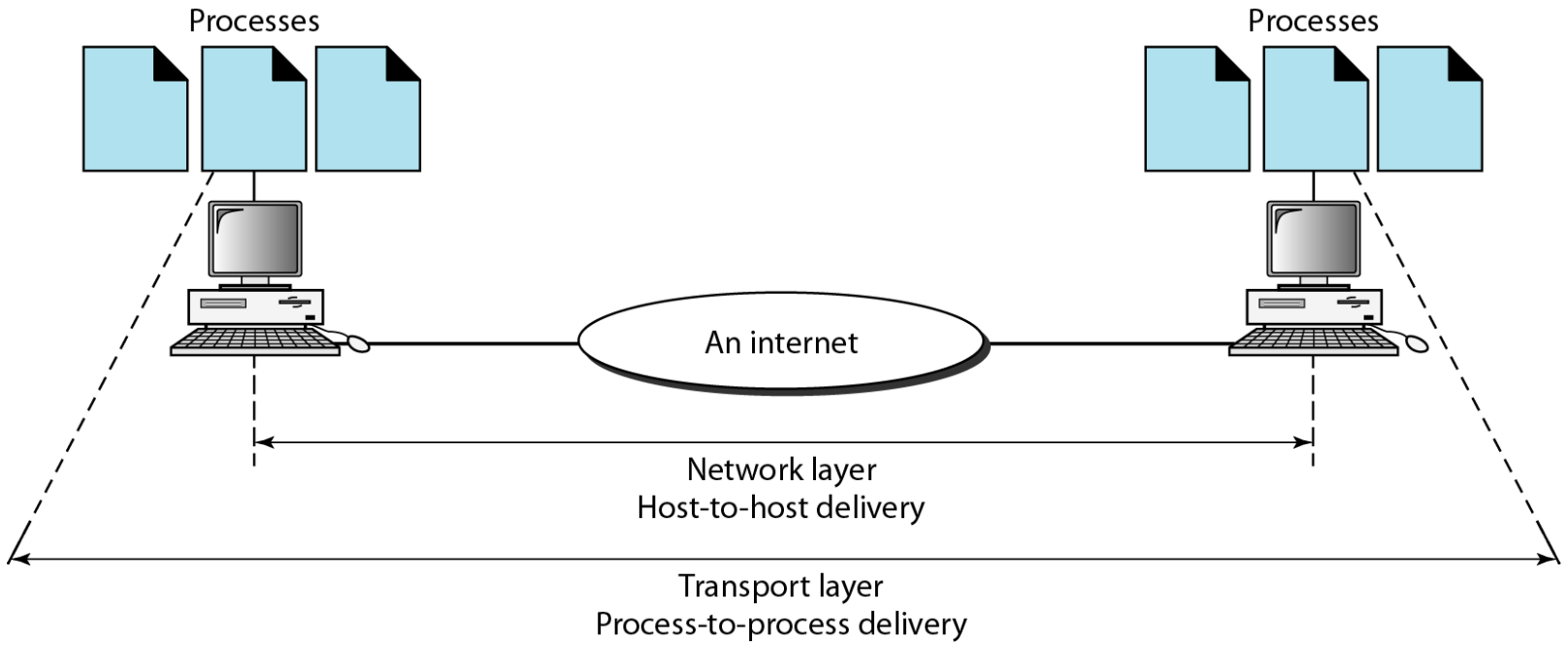
- **The transport layer is responsible for the delivery of a message from one process to another.**
  - Process-to- process delivery
  - Functions
    - Port addressing
    - Segmentation and reassembly
    - Connection control ( Connection-oriented or connection-less)
    - Flow control
    - Error control

# Transport layer

- Segmentation and reassembly



# Reliable process-to-process delivery of a message

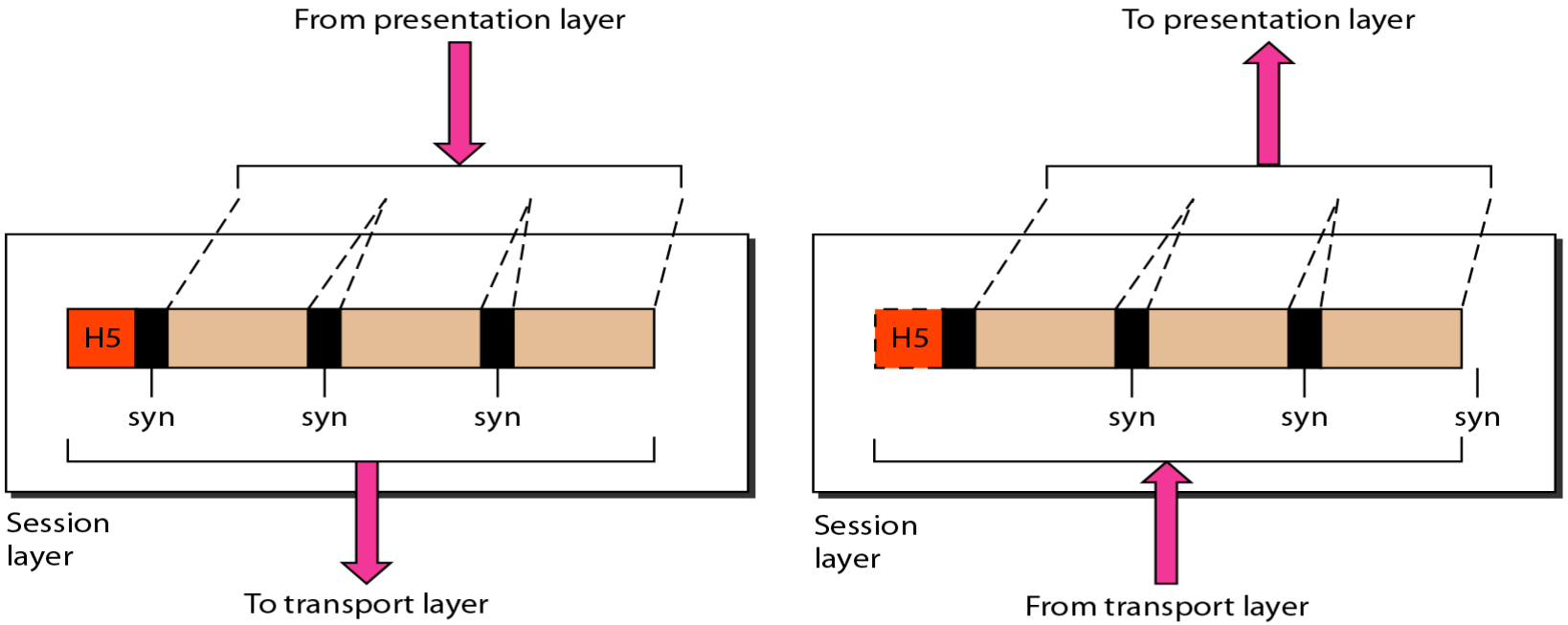


# Session Layer

- **The session layer is responsible for dialog control and synchronization.**

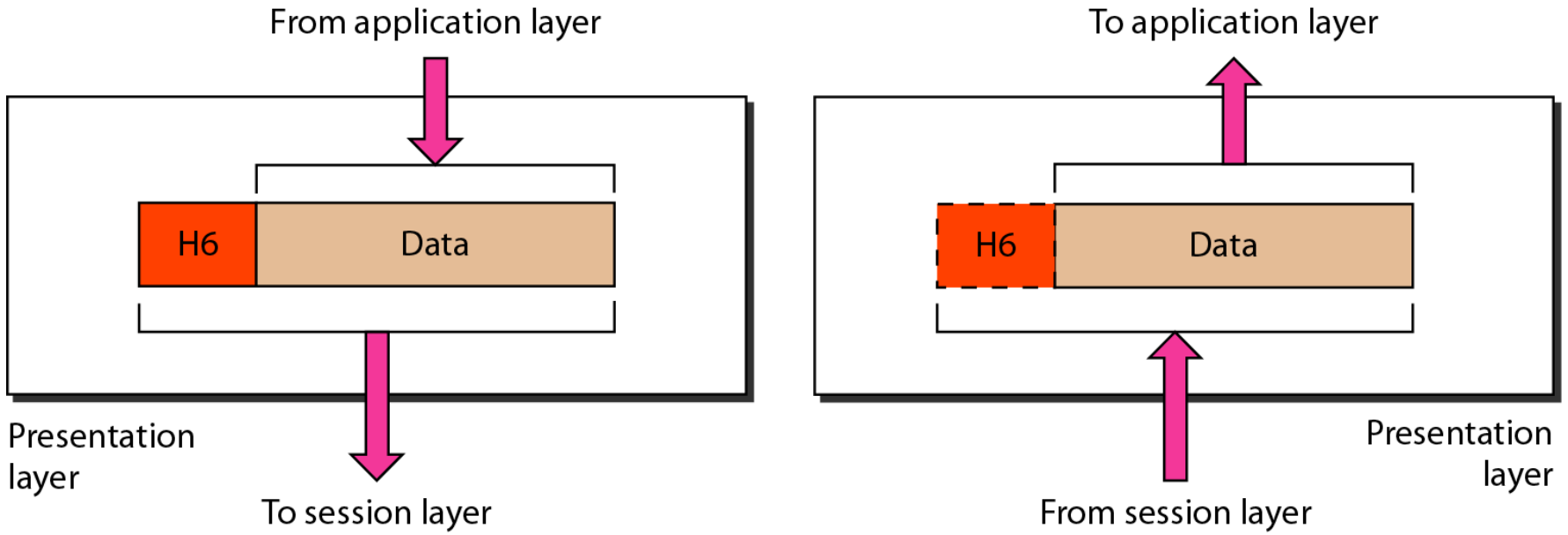
- **It establishes, maintains and synchronize the interaction between communicating system**
- **Function**
  - **Dialog control**
  - **Synchronization (checkpoints)**

## Synchronization



# Presentation Layer

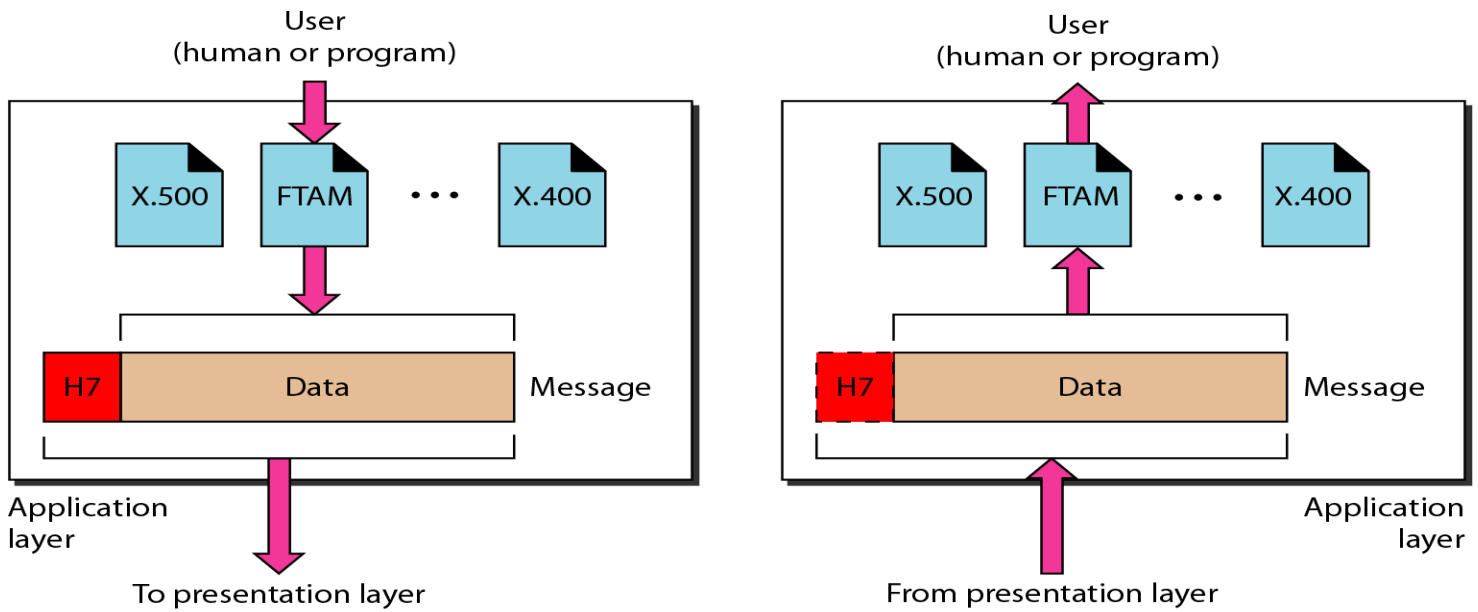
- **The presentation layer is responsible for translation, compression, and encryption.**
  - **Concerned with the syntax and semantics of the information exchanged between two system**
  - **Functions**
    - **Translation** ( EBCDIC-coded text file → ASCII-coded file)
    - **Encryption and Decryption**
    - **Compression**



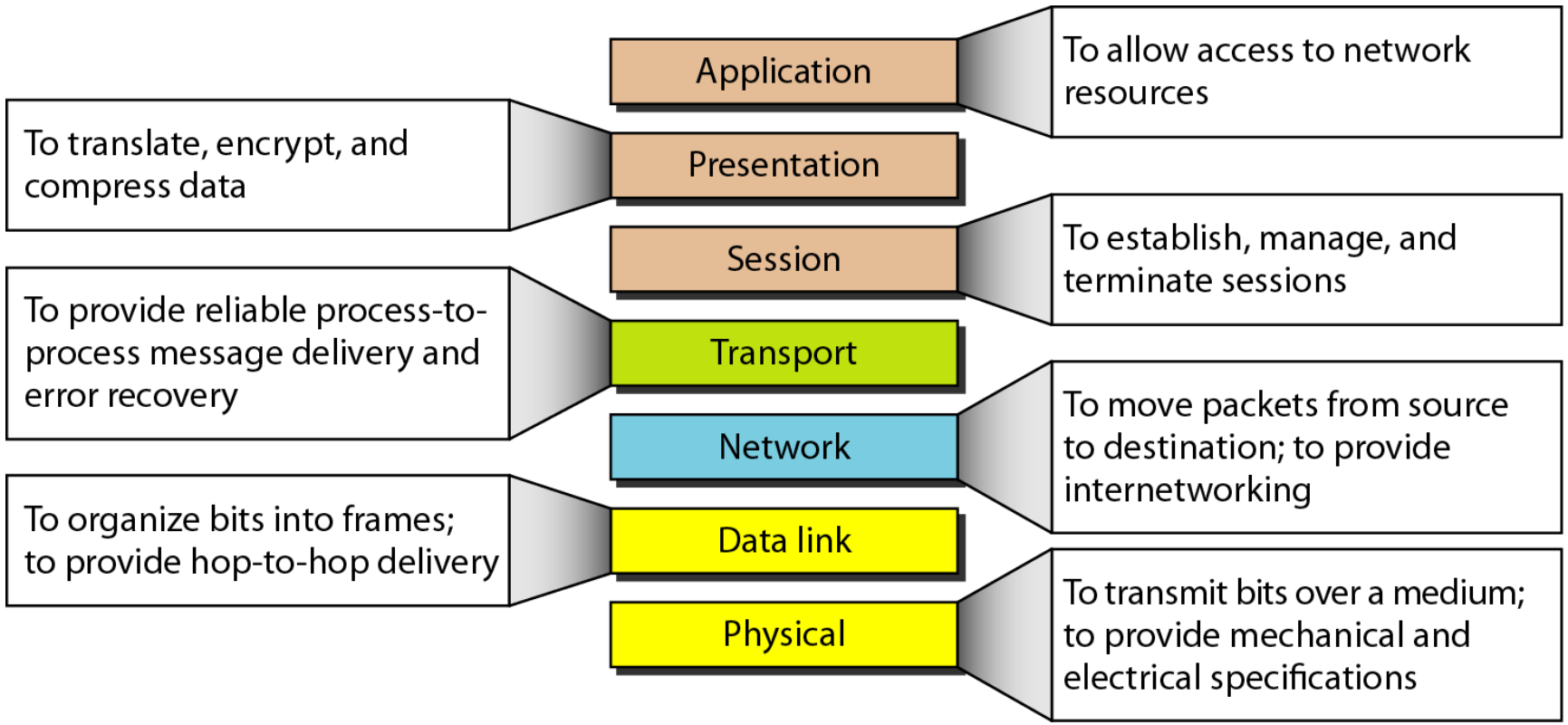
# Application Layer

**The application layer is responsible for providing services to the user.**

- Functions
  - Network virtual terminal (Remote log-in)
  - File transfer and access
  - Mail services
  - Directory services (Distributed Database)
  - Accessing the World Wide Web



# Summary of layers



# Summary of layers

Sender

| OSI Model | | | |
|---|---|---|---|
| | **Data unit** | **Layer** | **Function** |
| **User support layers** | Data | 7. Application | Network process to application |
| | | 6. Presentation | Data representation and encryption |
| | | 5. Session | Inter-host communication |
| User⇔ Network | Segment | 4. Transport | End-to-end connections and reliability |
| **Network support layers** | Packet | 3. Network | Path determination and logical addressing |
| | Frame | 2. Data Link | Physical addressing |
| | Bit | 1. Physical | Media, signal and binary transmission |

Receiver

# IT601 – System and Network Administration

# TCP/IP Model

Arif Husen

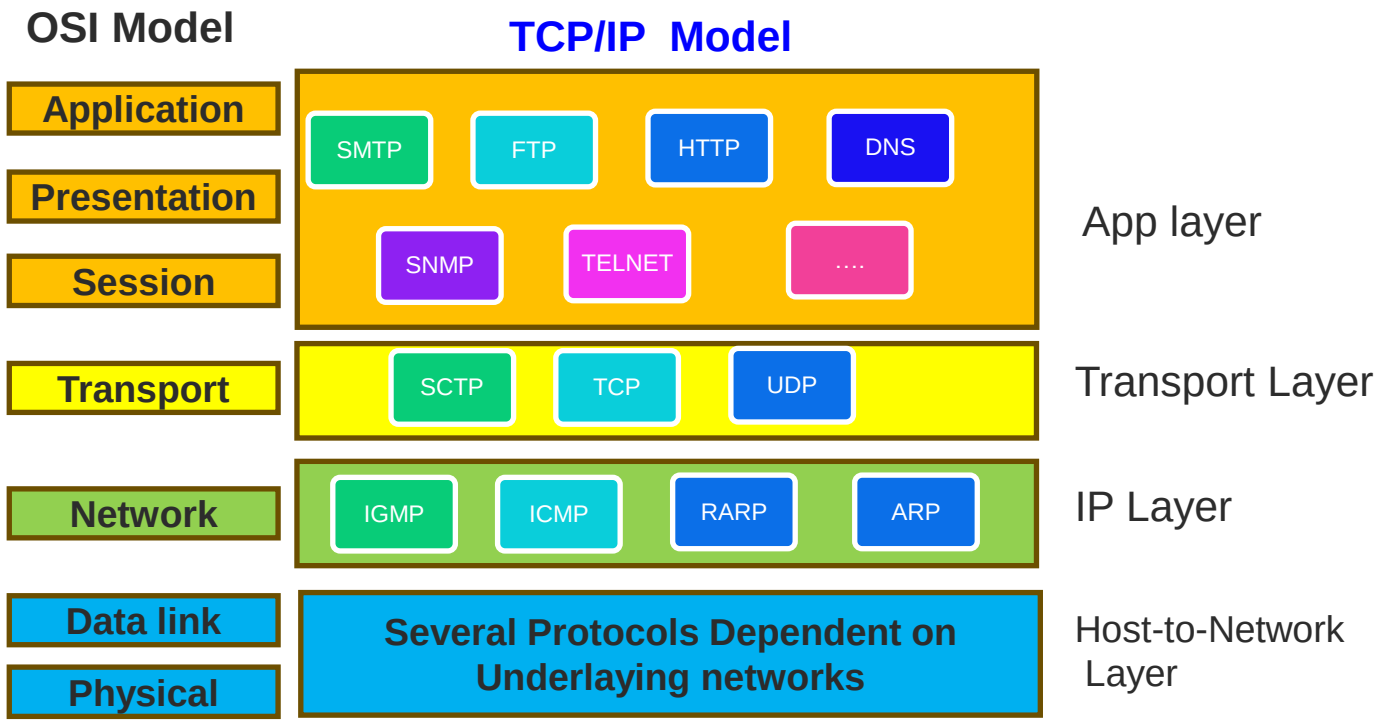
**Department of Computer Science and Information Technology,**

**Virtual University of Pakistan**

# TCP/IP PROTOCOL SUITE

- The layers in the TCP/IP protocol suite do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having four layers:
  - User Datagram Protocol (UDP)
  - Transmission Control Protocol (TCP)
  - Stream Control Transmission Protocol (SCTP)
- However, when TCP/IP is compared to OSI, we can say that the TCP/IP protocol suite is made of five layers: physical, data link, network, transport, and application.

- **Internet Layer**
  - TCP/IP support the IP ( unreliable).
  - IP is a host-to-host protocol.
  - Supporting protocols:
    - Address Resolution Protocol (ARP)
    - Reverse Address Resolution Protocol (RARP)
    - Internet Control Massage Protocol (ICMP)
    - Internet Group Massage Protocol (IGMP)
- **Transport Layer**
  - Process-to-process protocol.
    - host-to-network
    - Internet
    - Transport
    - application.



# *Addressing*

# 1-6 ADDRESSING

- Four levels of addresses are used in an internet employing the TCP/IP protocols:

## Physical

- Used on Physical and Data link Layer
- Examples are MAC Address…

## Logical

- Used on Network Layer
- Examples are IP and IPX addresses

## Port

- Used on Transport Layer
- Examples are TCP/UDP Addresses

## Specific

- Used on Application Layer
- Examples are Socket Addresses etc.

# Physical Address

- Physical addresses are imprinted on the NIC. Most local-area networks (Ethernet) use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon.

**Example:**

**07:01:02:01:2C:4B**

**A 6-byte (12 hexadecimal digits) physical address.**

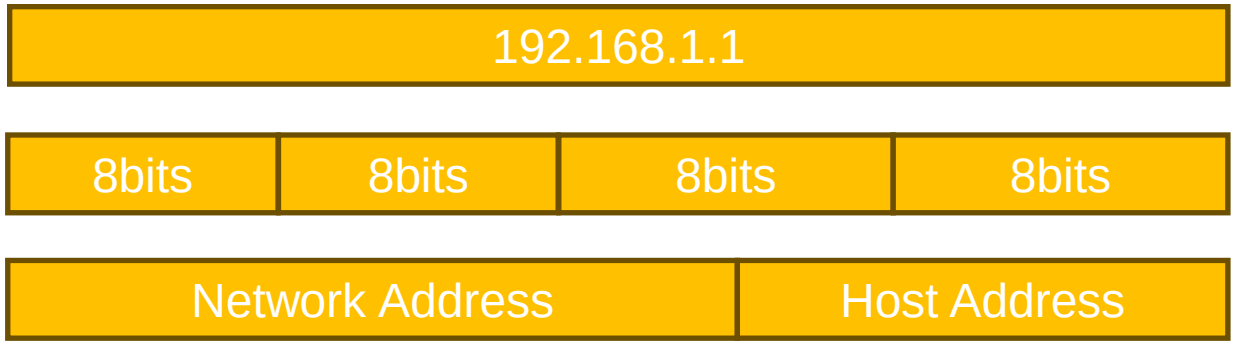
- known also as the MAC address
- Is the address of a node as defined by its LAN or WAN
- It is included in the frame used by data link layer

- The physical addresses in the datagram may change from hop to hop.

# Logical Address

- IP addresses are necessary for universal communications that are independent of physical network.
- No two host address on the internet can have the same IP address
- IP addresses in the Internet are 32-bit address that uniquely define a host.
- The physical addresses will change from hop to hop, but the logical addresses usually remain the same.
- IPv4 Addresses


- Dotted decimal notation



# IP Addressing

- Network Mask is used to identify the network and host addresses


| 11000000 | 10101000 | 00000001 | 00000001 | | 11111111 | 11111111 | 11111111 | 00000000 |
|---|---|---|---|---|---|---|---|---|
| 192 | 168 | 1 | 1 | | 255 | 255 | 255 | 0 |

192.168.1.1/24

192.168.1.1 mask 255.255.255.0

# Classes of Network Addresses

| Class | 1st octet | Network Mask |
|---|---|---|
| Class A Network | 0 - 127 | /8 |
| Class B Network | 128 - 191 | /16 |
| Class C Network | 192 - 223 | /24 |
| Class D Network | 224 - 239 | - |
| Class E Network | 240 - 255 | - |

| | | |
|---|---|---|
| Special IP address | 0.0.0.0/8 | addresses used to communicate with the local network |
| | 127.0.0.0/8 | Loopback Addresses |
| | 169.254.0.0/16 | Link-local Addresses |

# Public and Private Addresses

| Private IP Address | Public IP Address |
| --- | --- |
| • Used with LAN<br>• Not Recognized over internet<br>• Assigned by Sas<br>• No Cost | • Used in Public Network<br>• Recognized on Internet<br>• Assigned by IANA<br>• Unique Globally<br>• Not Free |
| • Class A : 10.0.0.0 to 10.255.255.255<br>• Class B : 172.16.0.0 to 172.31.255.255<br>• Class C : 192.168.0.0 to 192.168.255.255 | • Class A: 1.0.0.0 To 9.255.55.255<br>  • 11.0.0.0 To 126.255.255.255<br>• Class B : 128.0.0.0 To 172.15.255.255<br>  • 172.32.0.0 To 191.255.255.255<br>• Class C : 192.0.0.0 To 192.167.255.255<br>  • 192.169.0.0 To 223.255.255.255 |

# Unicast Vs Multicast Addresses

- A unicast address represents a single device in the network.
- A multicast address represents a group of devices in the network.
- A broadcast address represents all devices in the network.

| IP multicast address range | Description | Routable |
|---|---|---|
| 224.0.0.0 to 224.0.0.255 | Local subnetwork | No |
| 224.0.1.0 to 224.0.1.255 | Internetwork control | Yes |
| 224.0.2.0 to 224.0.255.255 | AD-HOC block 1 | Yes |
| 224.1.0.0 to 224.1.255.255 | Reserved | |
| 224.3.0.0 to 224.4.255.255 | AD-HOC block 2 | Yes |
| 225.0.0.0 to 231.255.255.255 | Reserved | |
| 232.0.0.0 to 232.255.255.255 | Source-specific multicast | Yes |
| 233.0.0.0 to 233.251.255.255 | GLOP addressing | Yes |
| 233.252.0.0 to 233.255.255.255 | AD-HOC block 3 | Yes |
| 234.0.0.0 to 234.255.255.255 | Unicast-prefix-based | Yes |
| 235.0.0.0 to 238.255.255.255 | Reserved | |
| 239.0.0.0 to 239.255.255.255 | Administratively scoped | Yes |

# Fixed and Variable Length Subnet Masking

- **In subnetting, a large network is logically or physically divided into multiple small networks or "subnets."**
  - subnetting a large network is to address network congestion and its negative impact on speed and productivity.
  - Subnetting also improves efficiency due to the way an address space is utilized in a small network.
  - The divisions between subnets allow organizations to enforce access controls, which improves network security, and helps contain security incidents.

- **Fixed vs. Variable Length Masking**
  - For subnetting an IP address for a network, one of two approaches can be used: VLSM or Fixed Length Subnet Mask (FLSM). These methods differ in three keyways:
    - FLSM creates subnets of the same size and an equal number of host identifiers, while VLSM creates subnets with varying sizes with a variable number of hosts.
    - FLSM is a better choice for private IP addresses, while VLSM is more suitable for public IP addresses.
    - FLSM tends to use more IP addresses than are necessary, which leads to wastage. In VLSM, wastage is minimum because it uses a given IP address range more efficiently.

# IPv6 Addresses

- **128 bits (or 16 bytes) long: four times as long as its predecessor.**
- **2^128 : about 340 billion billion billion billion different addresses**
- **Colon hexadecimal notation:**
  - addresses are written using 32 hexadecimal digits.
  - digits are arranged into 8 groups of four to improve the readability.
  - Groups are separated by colons

    2001:0718:1c01:0016:020d:56ff:fe77:52a3

- **IPv6 Zero Suppression**
  - To simplify the representation of IPv6 addresses, a contiguous sequence of 16-bit blocks set to 0 in the colon hexadecimal format can be compressed to "::", known as double-colon.
    - link-local address : FE80:0:0:0:2AA:FF:FE9A:4CA2 -> FE80::2AA:FF:FE9A:4CA2.
    - multicast address : FF02:0:0:0:0:0:0:2 -> FF02::2
    - loopback address : 0:0:0:0:0:0:0:1 -> ::1
  - Zero compression can only be used once in a given address.

# IPv6 Prefixes

- **The prefix is the part of the address that indicates the bits that have fixed values or are the bits of the subnet prefix.**

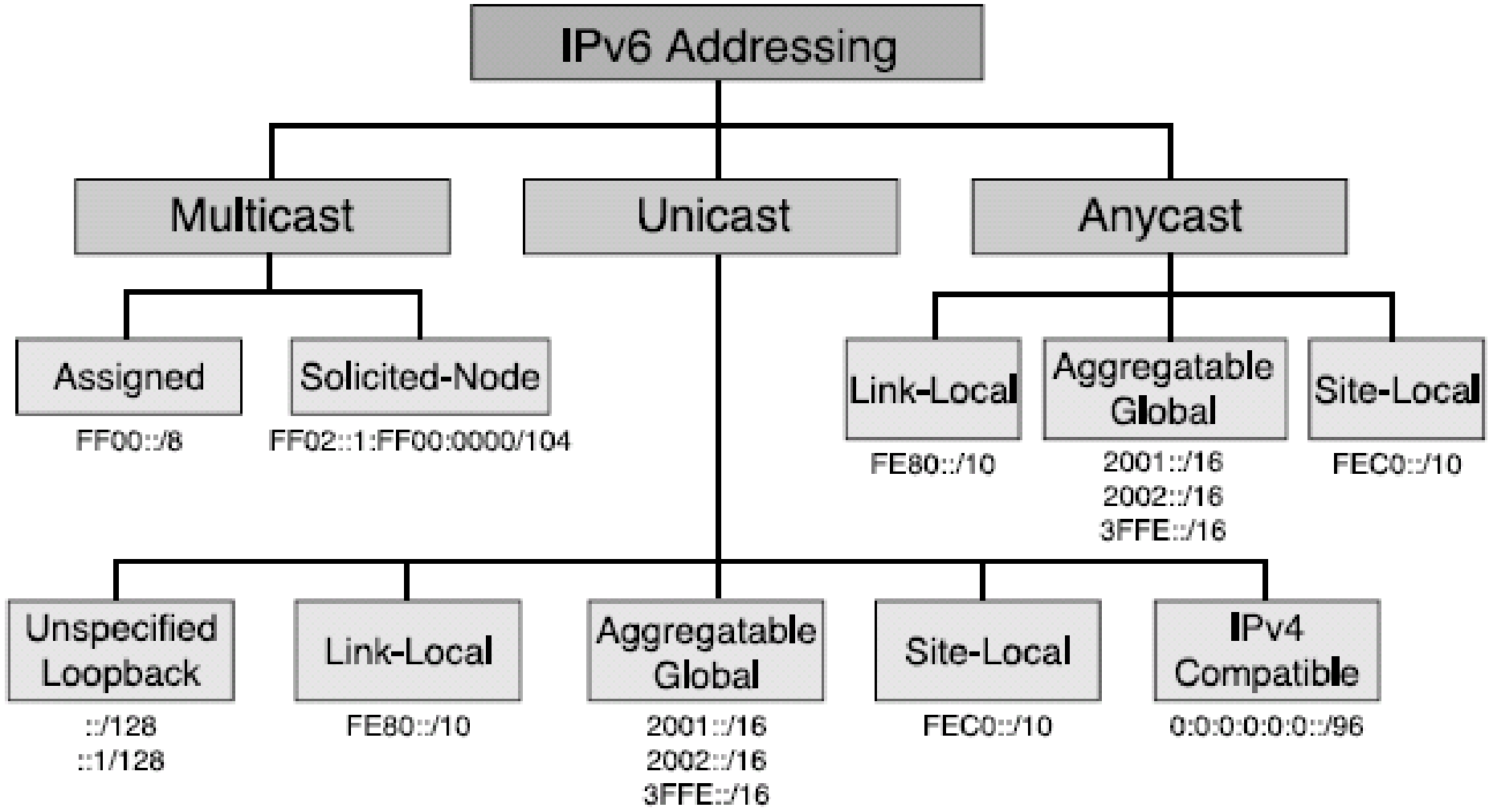
- **An IPv6 prefix is written in address/prefix-length notation.**
  - For example, 21DA:D3::/48 and 21DA:D3:0:2F3B::/64 are IPv6 address prefixes.

- **IPv6 Address Categories**

  - IPv6 have three types of addresses

    - Unicast : Transmission from source to single host
    - Multicast : Transmission from source to group (of hosts) address
    - Anycast : Transmission from source to single address of host where host has multiple ipv6 addresses

# IPv6 Address Types



# Port addresses

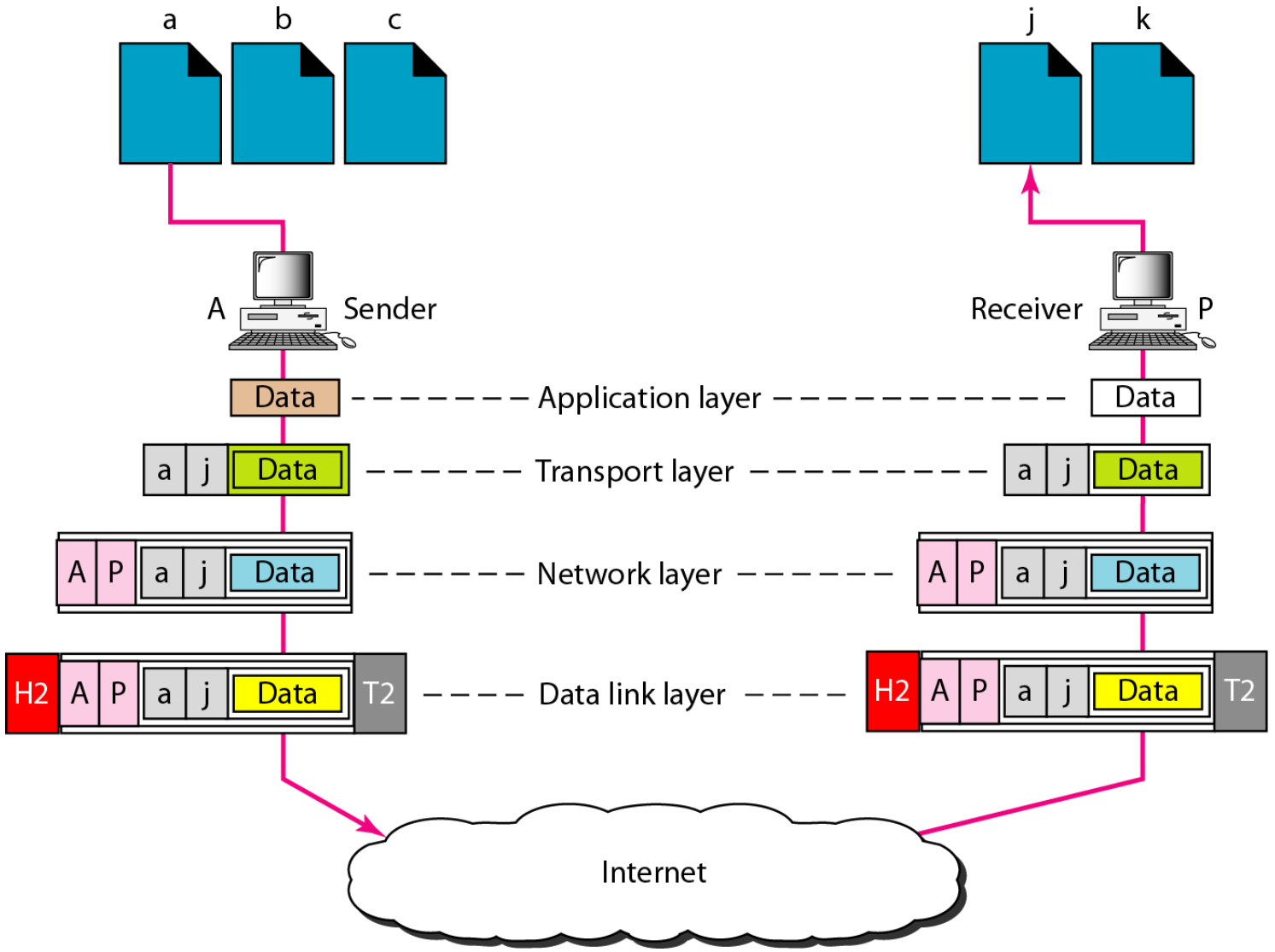
Port address is a 16-bit address represented by one decimal number ranged from (0-65535) to choose a process among multiple processes on the destination host.

- Destination port number is needed for delivery.
- Source port number is needed for receiving a reply as an acknowledgments.

In TCP/IP , a 16-bit port address represented as one single number. Example:  753

The physical addresses change from hop to hop, but the logical and port addresses usually remain the same.

# Port addresses



**IT601 – System and Network Administration**

# Collision and Broadcast Domains

Arif Husen

**Department of Computer Science and Information Technology,**

**Virtual University of Pakistan**

# Collision and Broadcast Domains

- **Collision Domains**

  In a group of devices or nodes, only one node can  transmit at a given time due shared medium.

- **Broadcast Domains**

  - In data link layers perspectives, The collection of nodes in  a given broadcast domain.

    Examples are :
    - VLANs
    - Isolated Networks

  - In network layer perspectives, The collection of nodes having IP address from a distinct network address specified with mask.

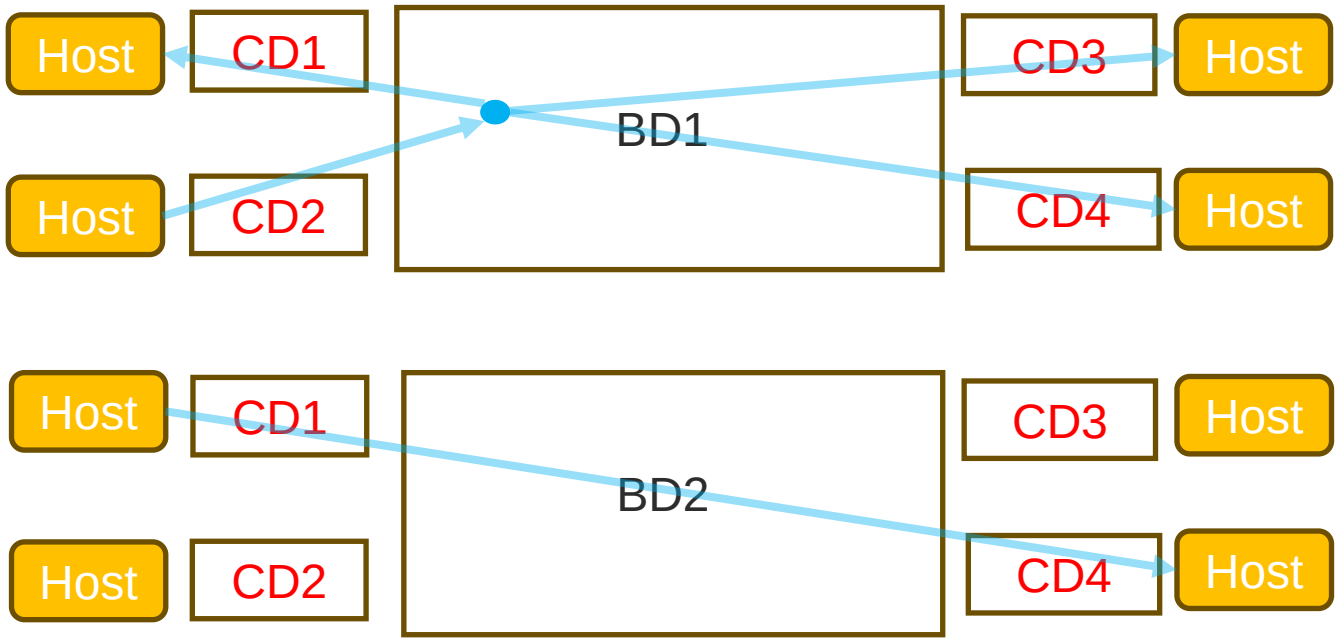
    - 192.168.1.0/24

# Broadcast and Collision Domains

- **Hub, Bridge, Switch and Router**

- **Broadcast Domain**
  - All hosts can listen

- **Collision Domain**
  - Only one host can send at t time.

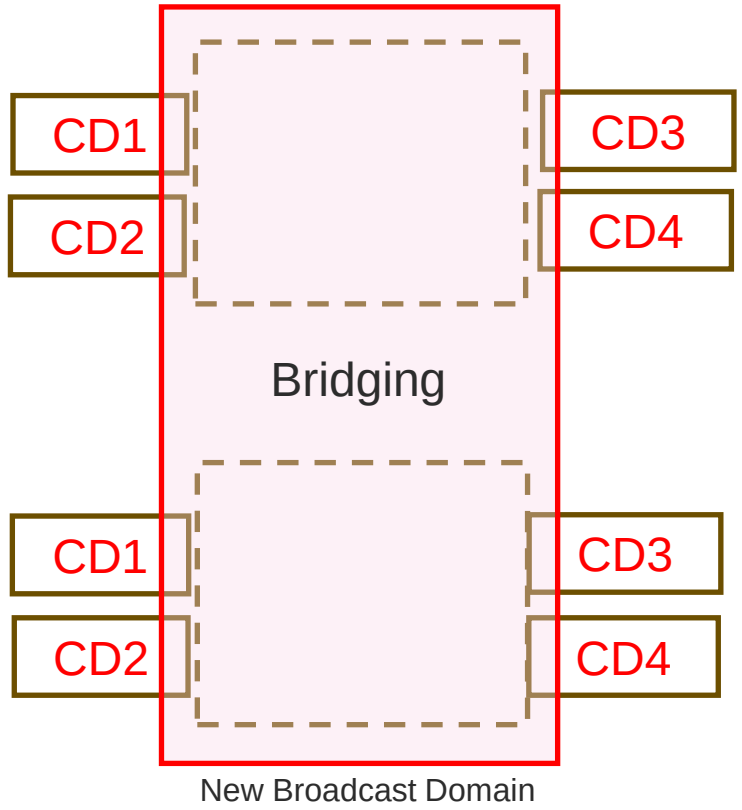


# Bridging

- **Bridging**

Joining of the separate networks in a way that both becomes one. Generally, works at  data link layer

- Existing BDs Merged
- No Effect on CDs
- Operates at Datalink Layer

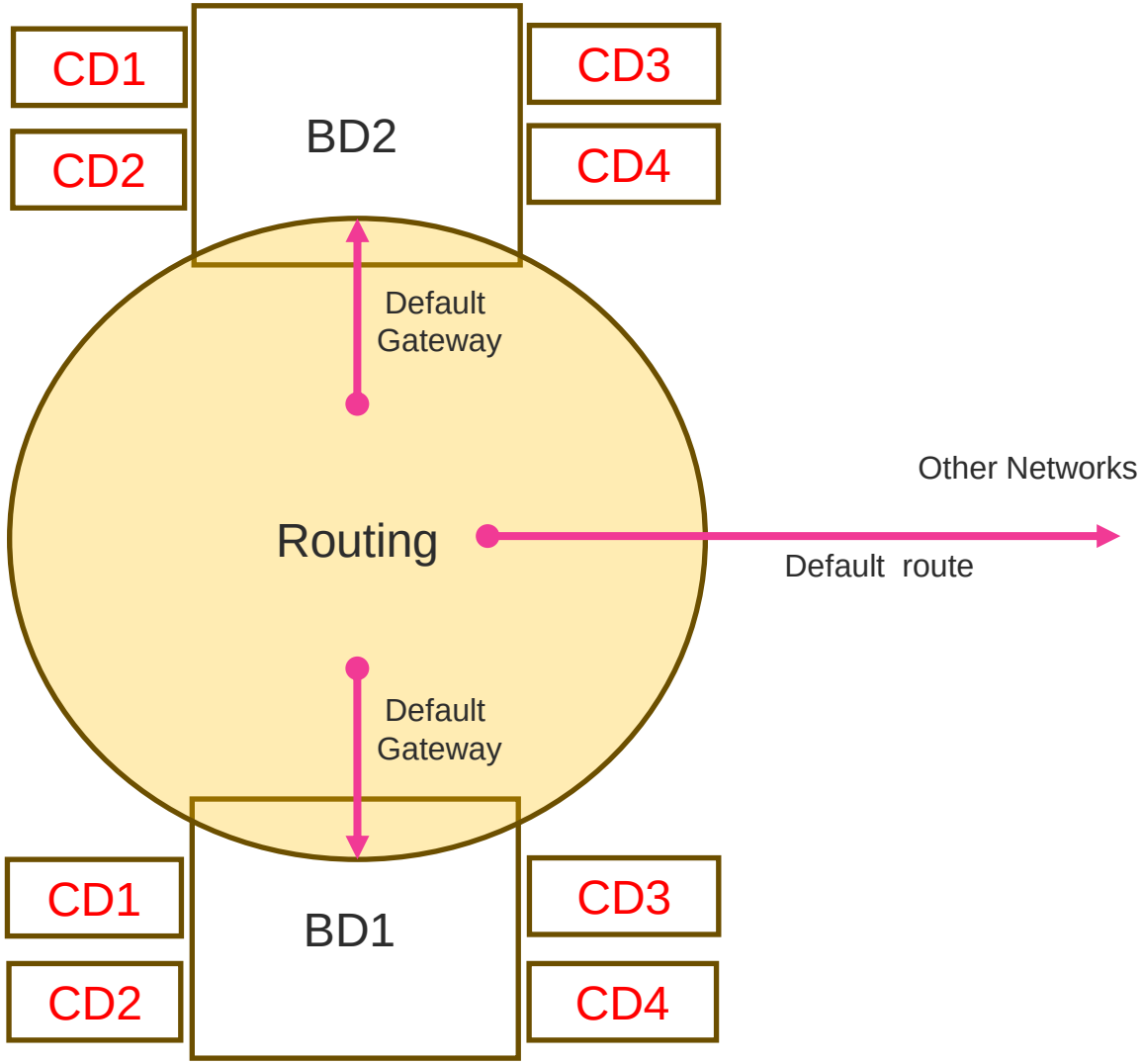



- **Switching**

Joining of the separate networks in a way that both becomes one.

# Routing

- **Routing**
  - Joining of networks without loosing their separation
  - Communications between disconnected networks
  - Operates at Network Layer
  - Default route, Default gateway, minimal cost link, shortest path, topology



**IT601 – System and Network Administration**

# Virtual Private Networking

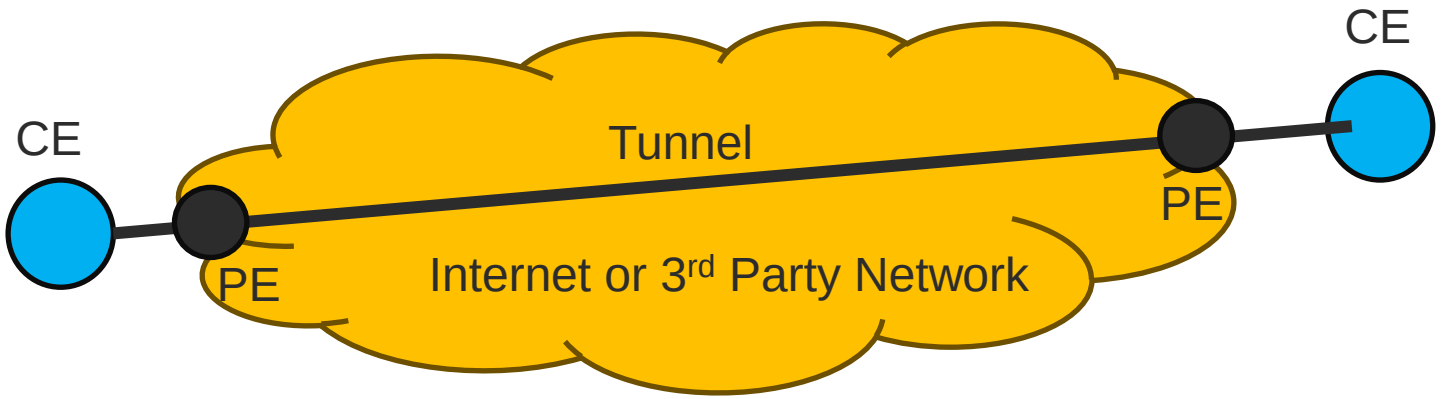
Arif Husen

**Department of Computer Science and Information Technology,**

**Virtual University of Pakistan**

# Virtual Private Networking

- **Layer 2 VPNs**
  - Extends networks at  datalink layer
  - Can Carry  STP, VLAN and  other L2 Control packets
  - More Secure than L3VPN , More Control



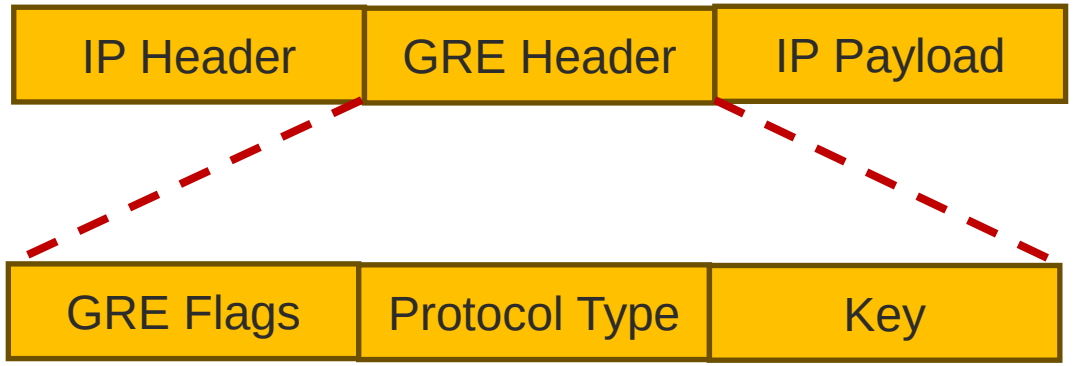

- **Layer 3 VPNs**
  - Extends networks at network layer
  - Can Carry  STP, VLAN and  other L2 Control packets
  - Encryption or Without Encryption
  - MP2MP

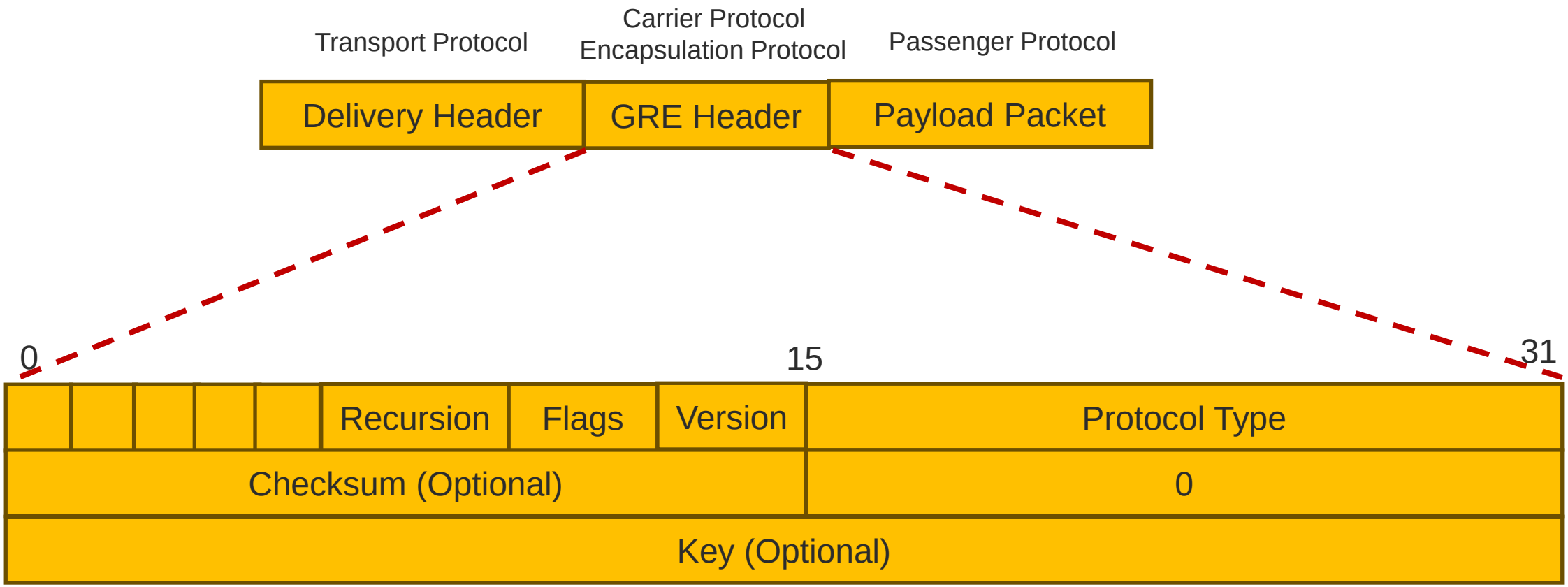
# Tunneling Protocols

- Generic Routing Encapsulation
- Internet Protocol Security
- Ip-in-IP
- SSH
- Point-to-Point Tunneling Protocol
- Secure Socket Tunneling Protocol
- Layer 2 Tunneling Protocol
- Virtual Extensible Local Area Network

# Generic Routing Encapsulation (GRE)

- Generic Routing Encapsulation is a method of encapsulation of IP packets in a GRE header that hides the original IP packet.
  - A new header named delivery header is added above the GRE header which contains the new source and destination address.
  - GRE header act as a new IP header with a Delivery header containing a new source and destination address.
  - Only routers between which GRE is configured can decrypt and encrypt the GRE header.
  - The original IP packet enters a router, travels in encrypted form, and emerges out of another GRE-configured router as the original IP packet as they have traveled through a tunnel.

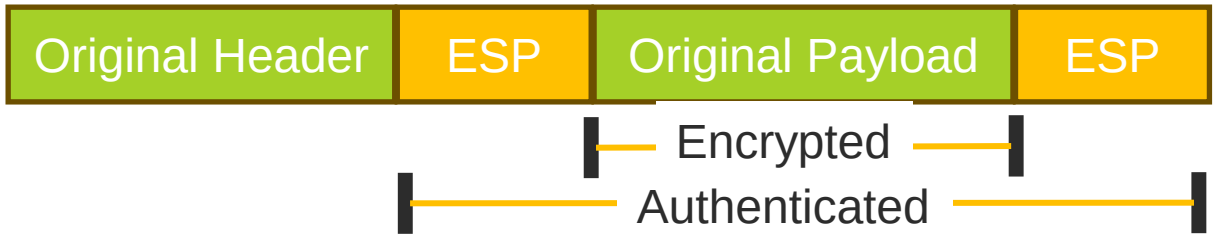

Transport Protocol
Carrier Protocol
Encapsulation Protocol
Passenger Protocol
Delivery Header
GRE Header
Payload Packet
0
15
31
Recursion
Flags
Version
Protocol Type
Checksum (Optional)
0
Key (Optional)

# IP security

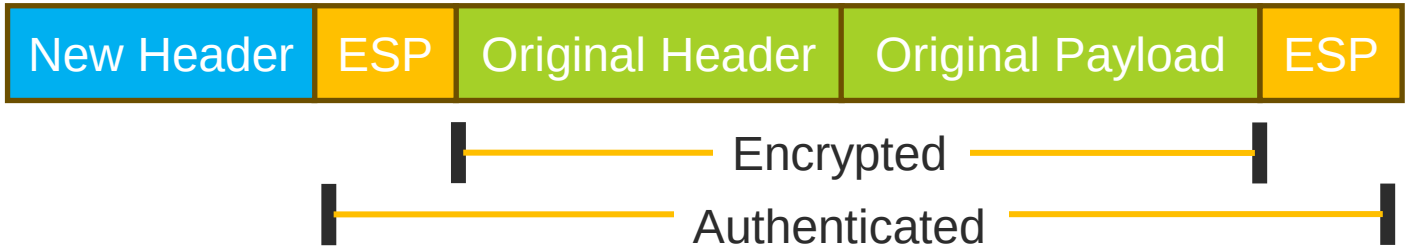
- IP security (IPSec) is an Internet Engineering Task Force (IETF) standard suite of protocols between 2 communication points across the IP network that provide data authentication, integrity, and confidentiality. It also defines the encrypted, decrypted, and authenticated packets.
- The protocols needed for secure key exchange and key management are defined in it.
- The AH (Authentication Header) protocol provides a mechanism for authentication only. AH provides data integrity, data origin authentication, and an optional replay protection service.
- The ESP (Ecapsulating Security Payload) protocol provides data confidentiality (encryption) and authentication (data integrity, data origin authentication, and replay protection). ESP can be used with confidentiality only, authentication only, or both confidentiality and authentication.

IP Packet W/o ESP


IP Packet With ESP Transport

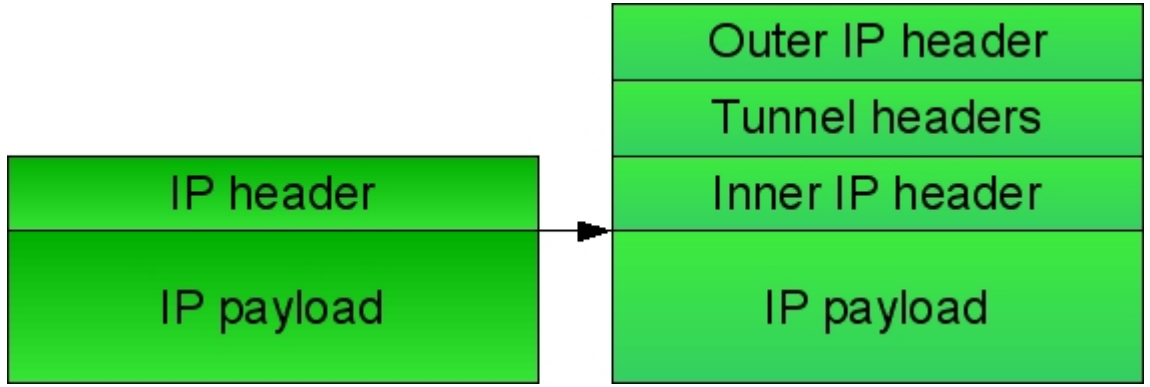



IP Packet With ESP Tunnel



# IP-in-IP

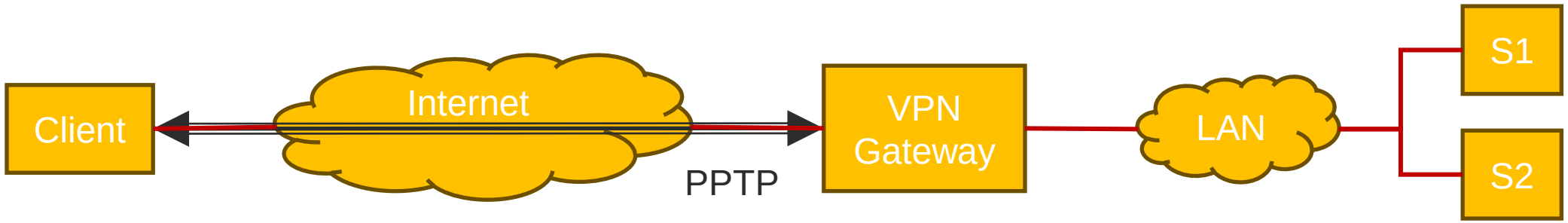
IP-in-IP is a Tunneling Protocol for encapsulating IP packets inside another IP packet.



# Point-to-Point Tunneling Protocol (PPTP)

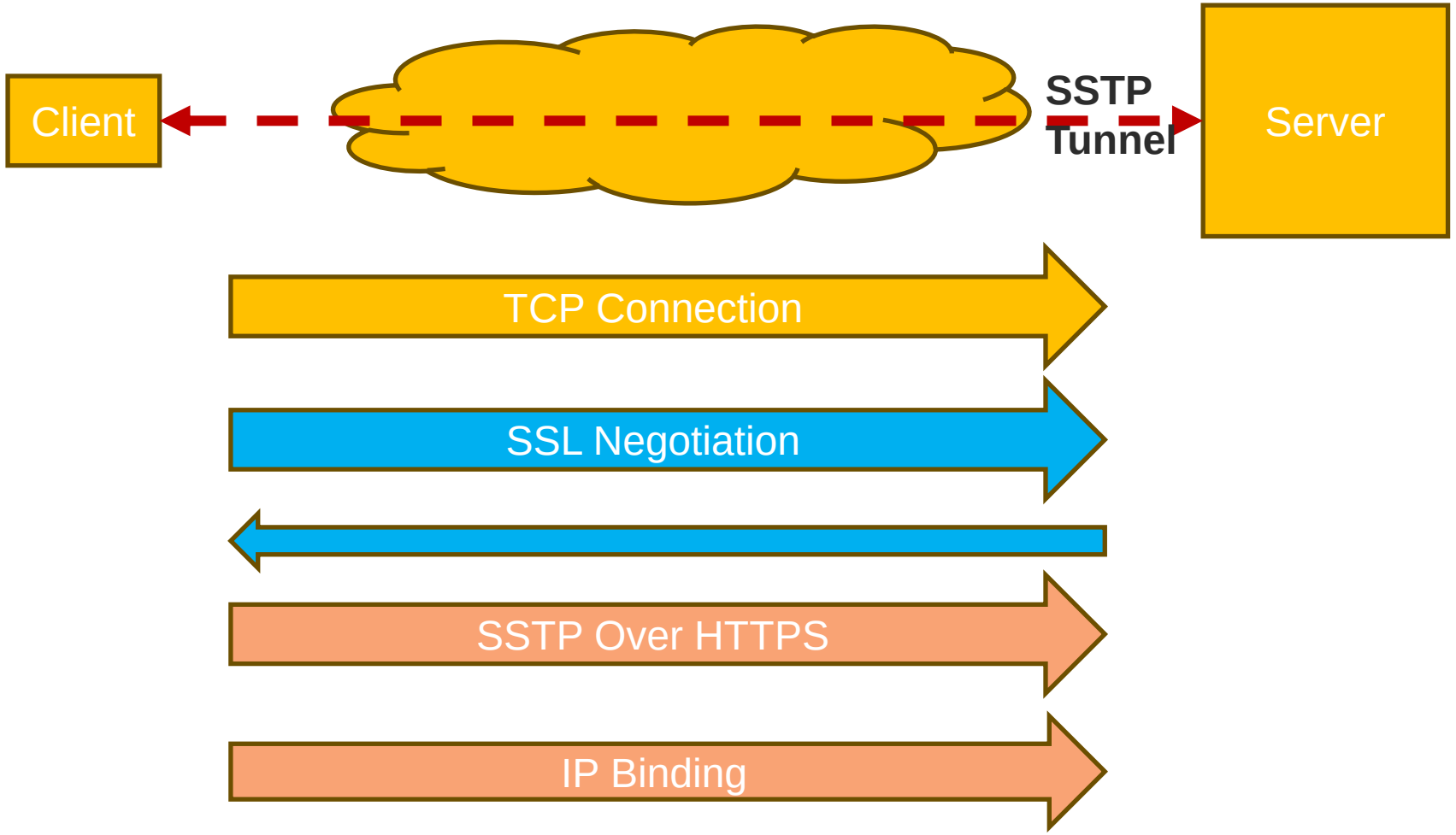
- PPTP or Point-to-Point Tunneling Protocol generates a tunnel and confines the data packet. Point-to-Point Protocol (PPP) is used to encrypt the data between the connection.

- PPTP is one of the most widely used VPN protocols and has been in use since the early release of Windows. PPTP is also used on Mac and Linux apart from Windows.



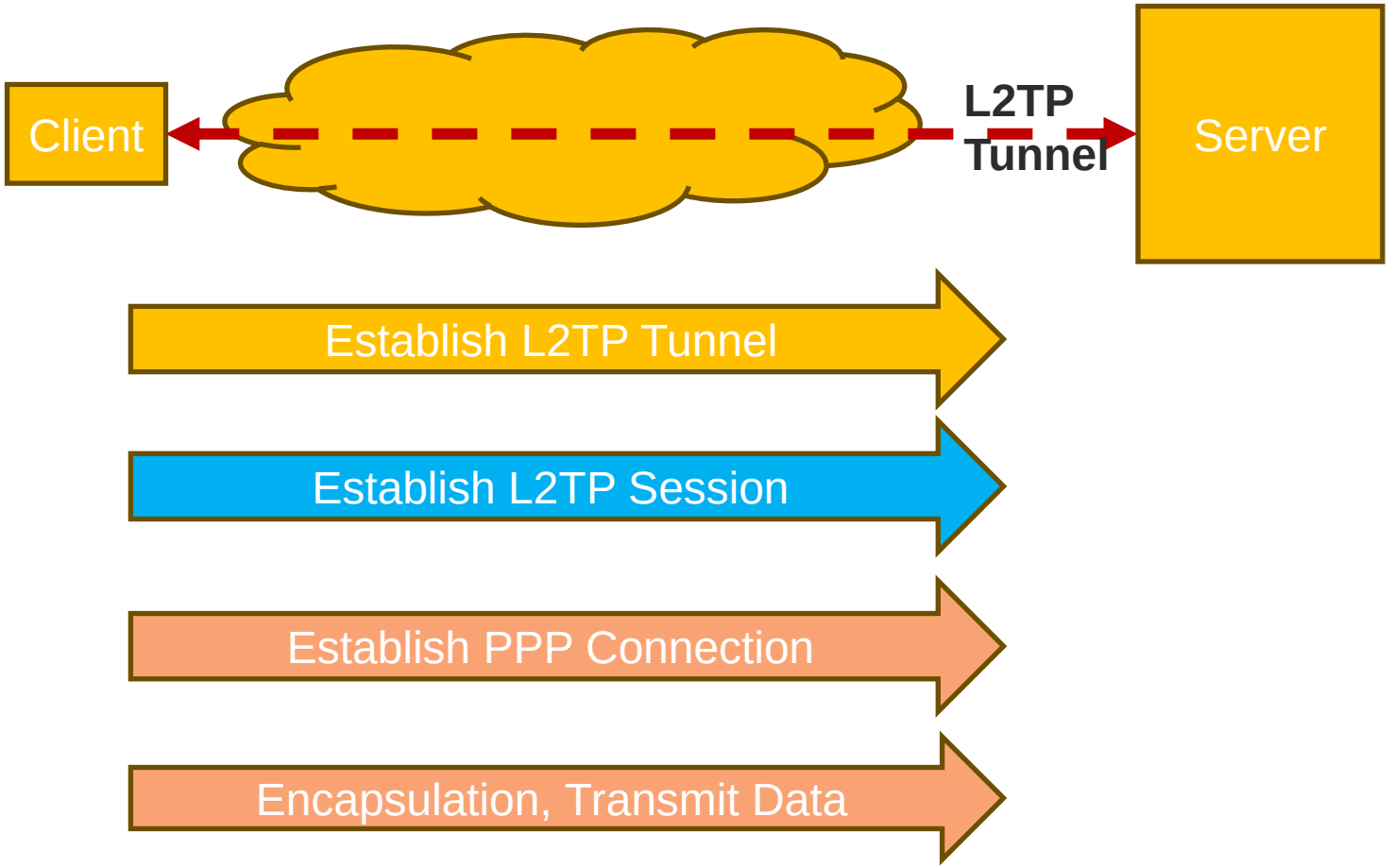
# 6. Secure Socket Tunneling Protocol (SSTP)

A VPN protocol developed by Microsoft that uses SSL to secure the connection, but only available for Windows.



# 7. Layer 2 Tunneling Protocol (L2TP)

- L2TP stands for Layer 2 Tunneling Protocol, published in 2000 as proposed standard RFC 2661.
- It is a computer networking protocol that was designed to support VPN connections used by an Internet service provider (ISP) to enable VPN operation over the Internet.
- L2TP combines the best features of two other tunneling protocols- PPTP(Point-to-Point Tunneling Protocol) from Microsoft and L2F(Layer 2 Forwarding) from Cisco Systems.

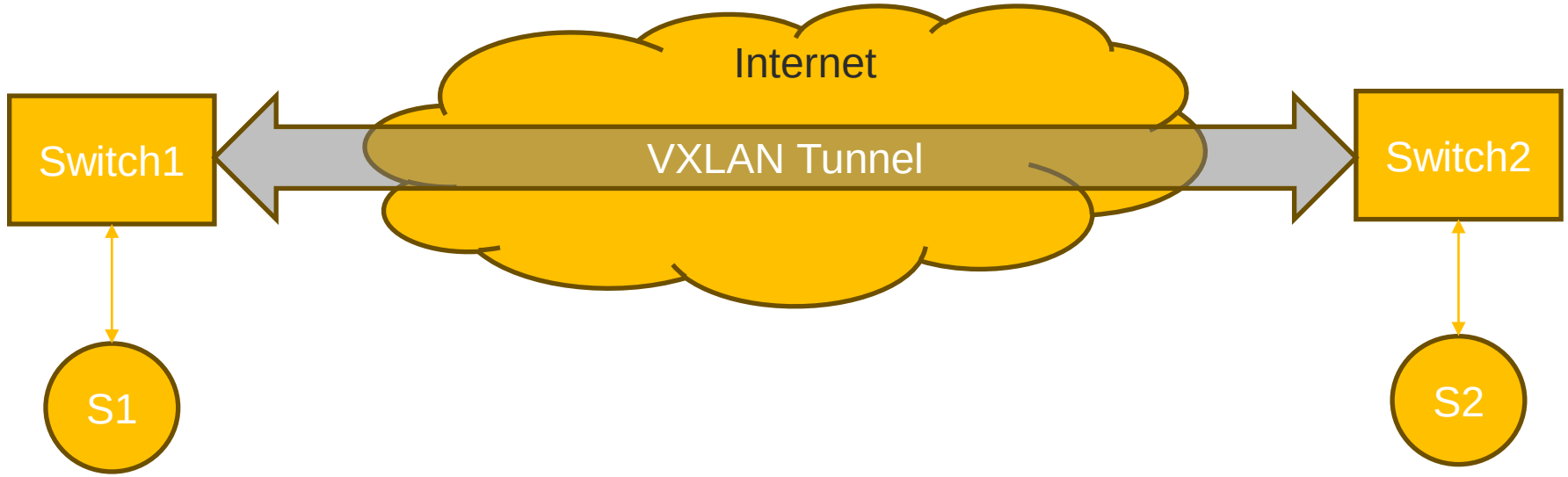



- sudo apt-get install xl2tpd openswan ppp

# 8. Virtual Extensible Local Area Network (VXLAN)

Virtual Extensible Local Area Network is short called VXLAN.

It is a network virtualization technology that stretches layer 2 connections over layer 3 networks by encapsulating Ethernet frames in a VXLAN packet which includes IP addresses to address the scalability problem in a more extensible manner.



IT601 – System and Network Administration

# Network Address Translation

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Introduction

- To access the Internet, one public IP address is needed, but we can use a private IP address in our private network.
  - The idea of NAT is to allow multiple devices to access the Internet through a single public address. To achieve this, the translation of a private IP address to a public IP address is required.
- Network Address Translation (NAT) is a process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts.
  - Also, it does the translation of port numbers i.e. masks the port number of the host with another port number, in the packet that will be routed to the destination.
  - It then makes the corresponding entries of IP address and port number in the NAT table. NAT generally operates on a router or firewall.

# Network Address Translation (NAT) working

- Generally, the border router is configured for NAT i.e the router which has one interface in the local (inside) network and one interface in the global (outside) network.
- When a packet traverse outside the local (inside) network, then NAT converts that local (private) IP address to a global (public) IP address.
- When a packet enters the local network, the global (public) IP address is converted to a local (private) IP address.
- If NAT runs out of addresses, i.e., no address is left in the pool configured then the packets will be dropped and an Internet Control Message Protocol (ICMP) host unreachable packet to the destination is sent.

# Masking Port Numbers

- Consider, in a network, two hosts A and B are connected. Now, both of them request for the same destination, on the same port number, say 1000, on the host side, at the same time.

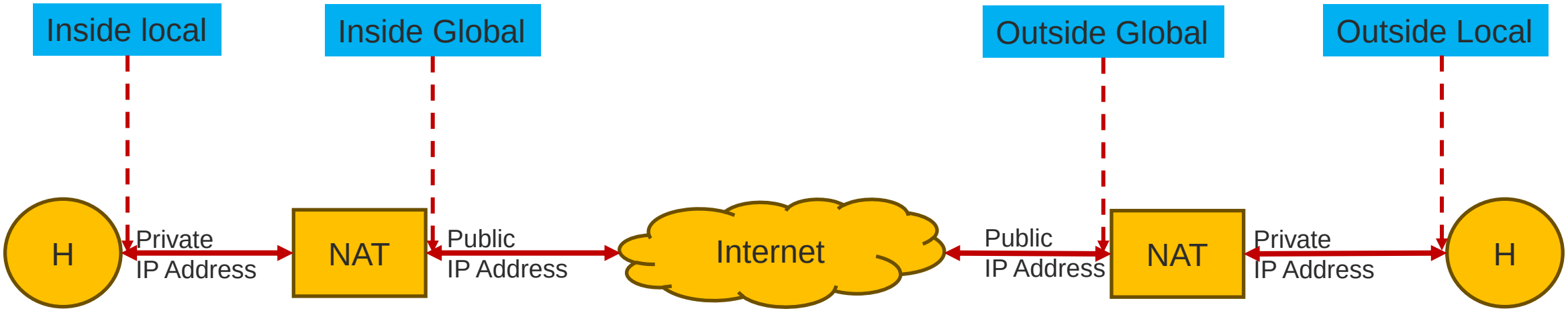
- If NAT does only translation of IP addresses, then when their packets will arrive at the NAT, both of their IP addresses would be masked by the public IP address of the network and sent to the destination.

- Destination will send replies to the public IP address of the router. Thus, on receiving a reply, it will be unclear to NAT as to which reply belongs to which host (because source port numbers for both A and B are the same).

- Hence, to avoid such a problem, NAT masks the source port number as well and makes an entry in the NAT table.

# NAT inside and outside addresses

Inside refers to the addresses which must be translated. Outside refers to the addresses which are not in control of an organization. These are the network Addresses in which the translation of the addresses will be done.



# Network Address Translation (NAT) Types

- There are three types of NAT

## Static NAT

- In this, a single unregistered (Private) IP address is mapped with a legally registered (Public) IP address i.e one-to-one mapping between local and global addresses.
- This is generally used for Web hosting. These are not used in organizations as there are many devices that will need Internet access and to provide Internet access, a public IP address is needed.

## Dynamic NAT

- In this type of NAT, an unregistered IP address is translated into a registered (Public) IP address from a pool of public IP addresses.
- If the IP address of the pool is not free, then the packet will be dropped as only a fixed number of private IP addresses can be translated to public addresses.

## Port Address Translation (PAT)

- Also known as NAT overload. In this, many local (private) IP addresses can be translated to a single registered IP address.
- Port numbers are used to distinguish the traffic i.e., which traffic belongs to which IP address.
- This is most frequently used as it is cost-effective as thousands of users can be connected to the Internet by using only one real global (public) IP address.

# Advantages/Disadvantages of NAT

- **Advantages of NAT –**
  - NAT conserves legally registered IP addresses.
  - It provides privacy as the device's IP address, sending and receiving the traffic, will be hidden.
  - Eliminates address renumbering when a network evolves.

- **Disadvantage of NAT –**
  - Translation results in switching path delays.
  - Certain applications will not function while NAT is enabled.
  - Complicates tunneling protocols such as IPsec.
  - Also, the router being a network layer device, should not tamper with port numbers(transport layer) but it has to do so because of NAT.

# Common Uses of NAT

- Wireless Routers or Access Points
- Virtual Machines
- Container Hosts

# IT601 – System and Network Administration

# IP Tables

Arif Husen

**Department of Computer Science and Information Technology,**

**Virtual University of Pakistan**

# Introduction to IP tables

- All modern operating systems come equipped with a firewall – a software application that regulates network traffic to a computer.
- Firewalls create a barrier between a trusted network (like an office network) and an untrusted one (like the internet).
- Firewalls work by defining rules that govern which traffic is allowed, and which is blocked. The utility firewall developed for Linux systems is iptables.

# Working Principles of IP Tables

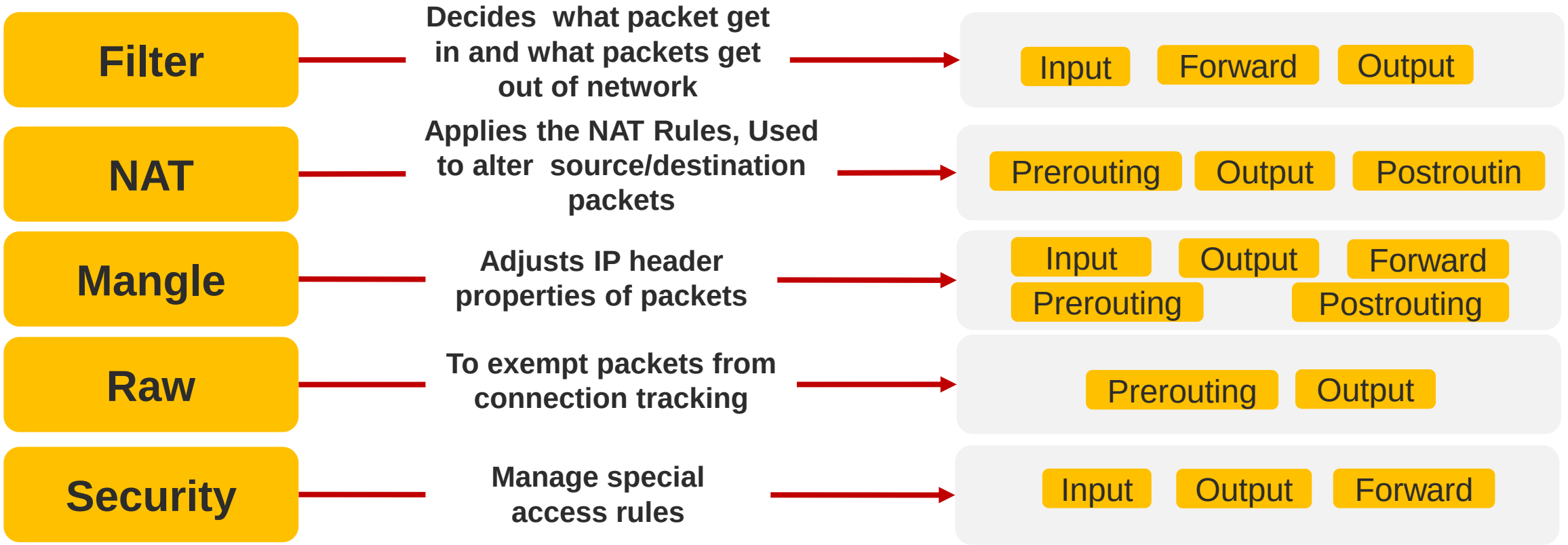
- Network traffic is made up of packets. Data is broken up into smaller pieces (called packets), sent over a network, then put back together. Iptables identifies the packets received and then uses a set of rules to decide what to do with them.

- Iptables filters packets based on:

| Tables | Chains | Rules | Targets |
|---|---|---|---|
| • Tables are files that join similar actions.<br>• A table consists of several chains | • A chain is a string of rules.<br>• When a packet is received, iptables finds the appropriate table, then runs it through the chain of rules until it finds a match. | • A rule is a statement that tells the system what to do with a packet.<br>• Rules can block one type of packet or forward another type of packet.<br>• The outcome, where a packet is sent, is called a target. | • A target is a decision of what to do with a packet.<br>• Typically, this is to accept it, drop it, or reject it (which sends an error back to the sender). |

# Tables and Chains

- Linux firewall iptables has four default tables.



# Chains

# Targets

- A target is what happens after a packet matches a rule criteria. Non-terminating targets keep matching the packets against rules in a chain even when the packet matches a rule.
- With terminating targets, a packet is evaluated immediately and is not matched against another chain. The terminating targets in Linux iptables are:
- Four Targets are defined.

## Accept

accepts the packets to come through the iptables firewall

## Drop

the dropped package is not matched against any further chain

does not generate an error

## Return

sends the packet back to the originating chain

## Reject

rejects a packet

Generates an error to the connecting device

# Installation and Status

- Install IP Tables
  - sudo apt-get update
  - sudo apt-get install iptables

- Check the status of current iptables configuration
  - sudo iptables -L -v

Here, the -L option is used to list all the rules, and -v is for showing the info in a more detailed format. Below is the example output:

```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in out   source destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in out   source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in out   source destination
```

# Defining Chain Rules

- A rule can be inserted with following command
  - sudo iptables -A <chain> -i <interface> -p <protocol (tcp/udp) > -s <source> --dport <port no.> -j <target>

- **-i (interface)**
  - The network interface whose traffic you want to filter, such as eth0, lo, ppp0, etc.
- **-p (protocol)**
  - The network protocol where your filtering process takes place. It can be either tcp, udp, udplite, icmp, sctp, icmpv6, and so on. Alternatively, you can type all to choose every protocol.
- **-s (source)**
  - The address from which traffic comes from. You can add a hostname or IP address.
- **–dport (destination port)**
  - the destination port number of a protocol, such as 22 (SSH), 443 (https), etc.
- **-j (target)**
  - the target name (ACCEPT, DROP, RETURN). You need to insert this every time you make a new rule.

# Allow traffic on localhost

- To allow traffic on localhost, type this command:
  - sudo iptables -A INPUT -i lo -j ACCEPT
- Use lo or loopback interface. It is utilized for all communications on the localhost.
- The command above will make sure that the connections between a database and a web application on the same machine are working properly.

# Enabling Connections on HTTP, SSH, and SSL Port

- **Allow SSH Traffic**
  - sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
- **Allow HTTP Traffic**
  - sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
- **Allow HTTPS Traffic**
  - sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
- **Check Status of IP Tables**

  sudo iptables -L -v

# Filtering Packets Based on Source

- **Allow packets from 192.168.1.3**

  sudo iptables -A INPUT -s 192.168.1.3 -j ACCEPT

- **Deny packets from 192.168.1.3**

  sudo iptables -A INPUT -s 192.168.1.3 -j DROP

- **Deny packets from a range of IP addresses**

  sudo iptables -A INPUT -m iprange --src-range 192.168.1.100-192.168.1.200 -j DROP

- **Dropping all Other Traffic**

  sudo iptables -A INPUT -j DROP

# Deleting Rules

- **remove all rules and start with a clean slate**
  - sudo iptables -F
- **delete a specific rule**
  - sudo iptables -L --line-numbers
- **delete a specific rule from specific chain**
  - sudo iptables -D INPUT 3

# Persisting Changes

- **Save Rules to file**
  - sudo iptables-save > /etc/iptables/rules.v4
  - sudo iptables-save > /etc/iptables/rules.v6
- **Restore rules from file**
  - sudo iptables-restore < /etc/iptables/rules.v4
  - sudo iptables-restore < /etc/iptables/rules.v6
- **Enable Auto Save**
  - sudo apt-get install iptables-persistent

**IT601 – System and Network Administration**

# Simple Network Management Protocol

Arif Husen

**Department of Computer Science and Information Technology,**

**Virtual University of Pakistan**

# Simple Network Management Protocol

- SNMP is a framework that provides facilities for managing and monitoring network resources on the Internet.
- Components of SNMP:

## Agent

**SNMP agent** is software that runs on a piece of network equipment (host, router, printer, or others) and that maintains information about its configuration and current state in a database

## Manager

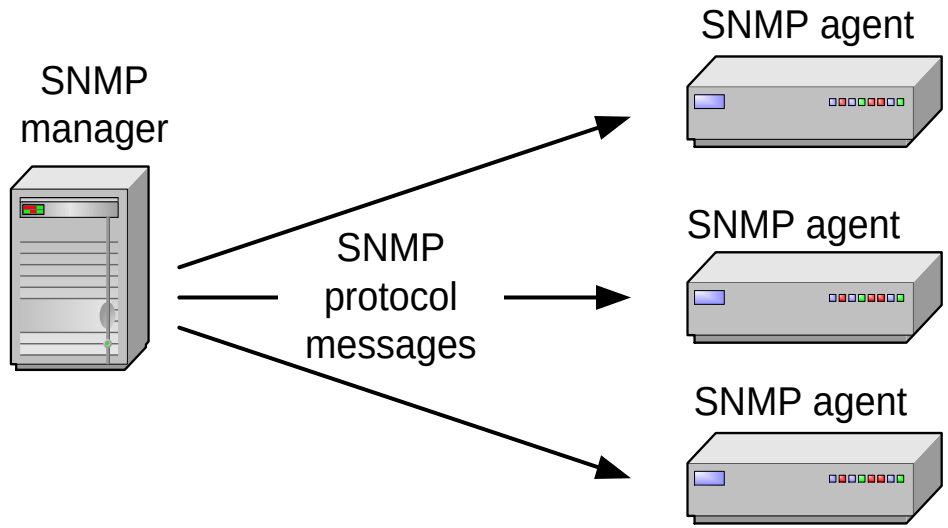
An **SNMP manager** is an application program that contacts an SNMP agent to query or modify the database at the agent.

## MIB

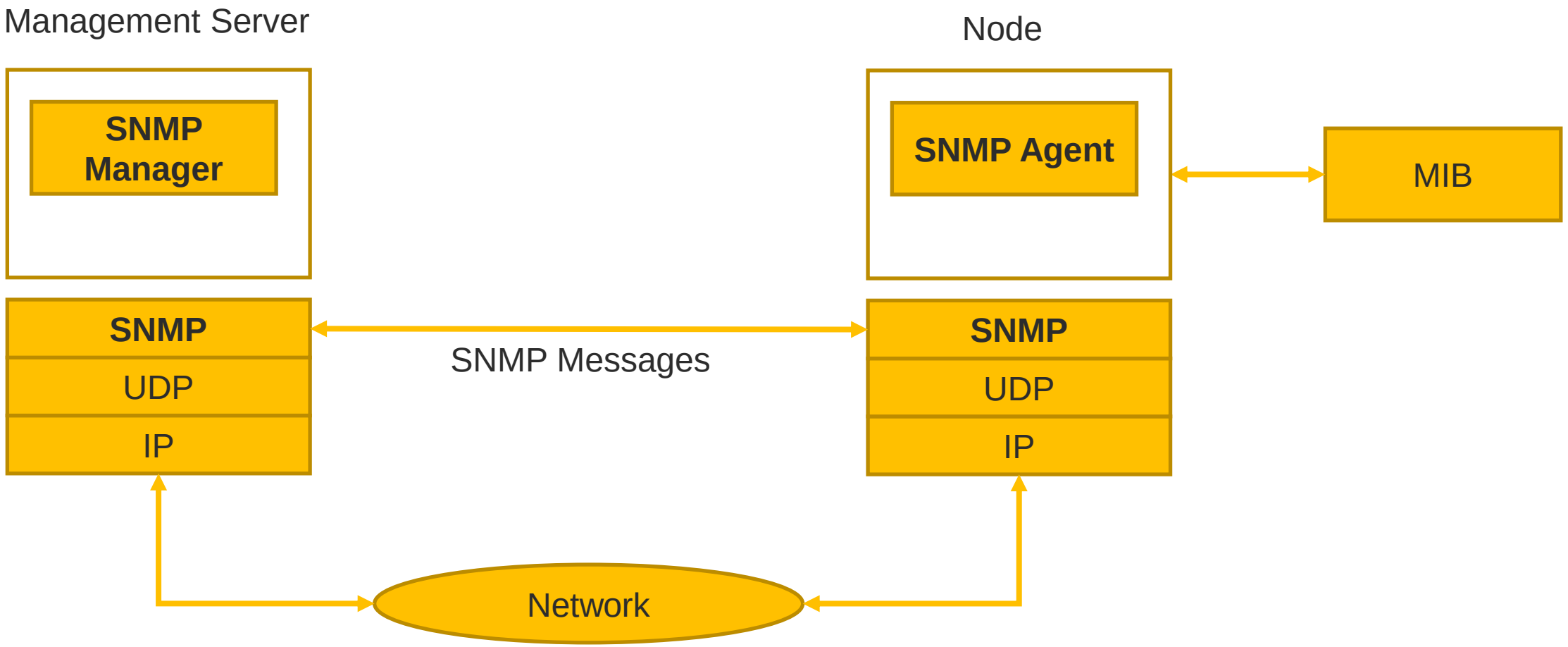
Information in the database is described by **Management Information Bases (MIBs)**

## Protocol

**SNMP protocol** is the application layer protocol used by SNMP agents and managers to send and receive data.



# Interactions in SNMP



# ASN.1 Notation

- Abstract Syntax Notation One (ASN.1) is a standard interface description language for defining data structures that can be serialized and deserialized in a cross-platform way.
- It is broadly used in telecommunications and computer networking, and especially in cryptography

- ASN.1 is a data type declaration notation. It does not define how to manipulate a variable of such a type.
  - Manipulation of variables is defined in other languages such as SDL (Specification and Description Language) for executable modeling or TTCN-3 (Testing and Test Control Notation) for conformance testing.
  - Both these languages natively support ASN.1 declarations. It is possible to import an ASN.1 module and declare a variable of any of the ASN.1 types declared in the module.

```
DemoProtocol DEFINITIONS ::= BEGIN

    DemoQ ::= SEQUENCE {
        trackingNumber INTEGER,
        question       IA5String
    }

    DemoA ::= SEQUENCE {
        questionNumber INTEGER,
        answer         BOOLEAN
    }

END
```

- ASN.1 is a joint standard of the International Telecommunication Union Telecommunication Standardization Sector (IT`U-T) in ITU-T Study Group 17 and ISO/IEC

# MIBS

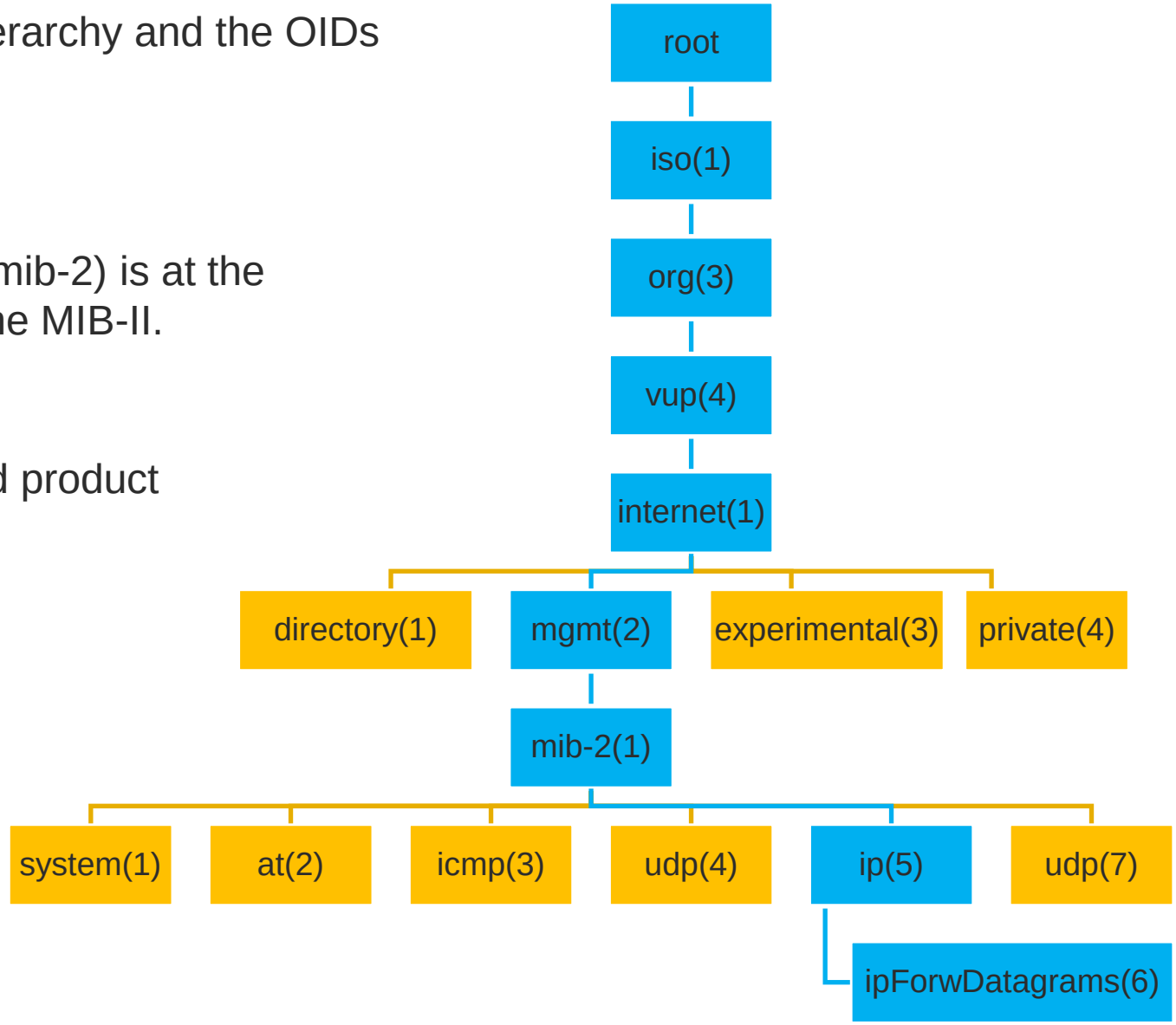
- A MIB specifies the managed objects
- MIB is a text file that describes managed objects using the syntax of ASN.1 (Abstract Syntax Notation 1)
- ASN.1 is a formal language for describing data and its properties
- In Linux, MIB files are in the directory */usr/share/snmp/mibs*
  - *Multiple MIB files*
  - *MIB-II (defined in RFC 1213) defines the managed objects of TCP/IP networks*

# Managed Objects

- Each managed object is assigned an object identifier (OID)
- The OID is specified in a MIB file.
- An OID can be represented as a sequence of integers separated by decimal points or by a text string:
  - Example:
    1.3.6.1.2.1.4.6.
    iso.org.dod.internet.mgmt.mib-2.ip.ipForwDatagrams
- When an SNMP manager requests an object, it sends the OID to the SNMP agent.

# Organization of managed objects

- Managed objects are organized in a tree-like hierarchy and the OIDs reflect the structure of the hierarchy.
- Each OID represents a node in the tree.
- The OID 1.3.6.1.2.1 (iso.org.dod.internet.mgmt.mib-2) is at the top of the hierarchy for all managed objects of the MIB-II.
- Manufacturers of networking equipment can add product specific objects to the hierarchy.



# Definition of managed objects in a MIB

- Specification of ipForwDatagrams in MIB-II.

**ipForwDatagrams OBJECT-TYPE**

SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION

"The number of input datagrams for which this entity was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination. In entities which do not act as IP Gateways, this counter will include only those packets which were Source-Routed via this entity, and the Source- Route option processing was successful."

::= { ip 6 }

# SNMP Messages

- SNMP manager and an SNMP agent communicate using the SNMP protocol
  - Generally: Manager sends queries and agent responds
  - Exception: Traps are initiated by agent.

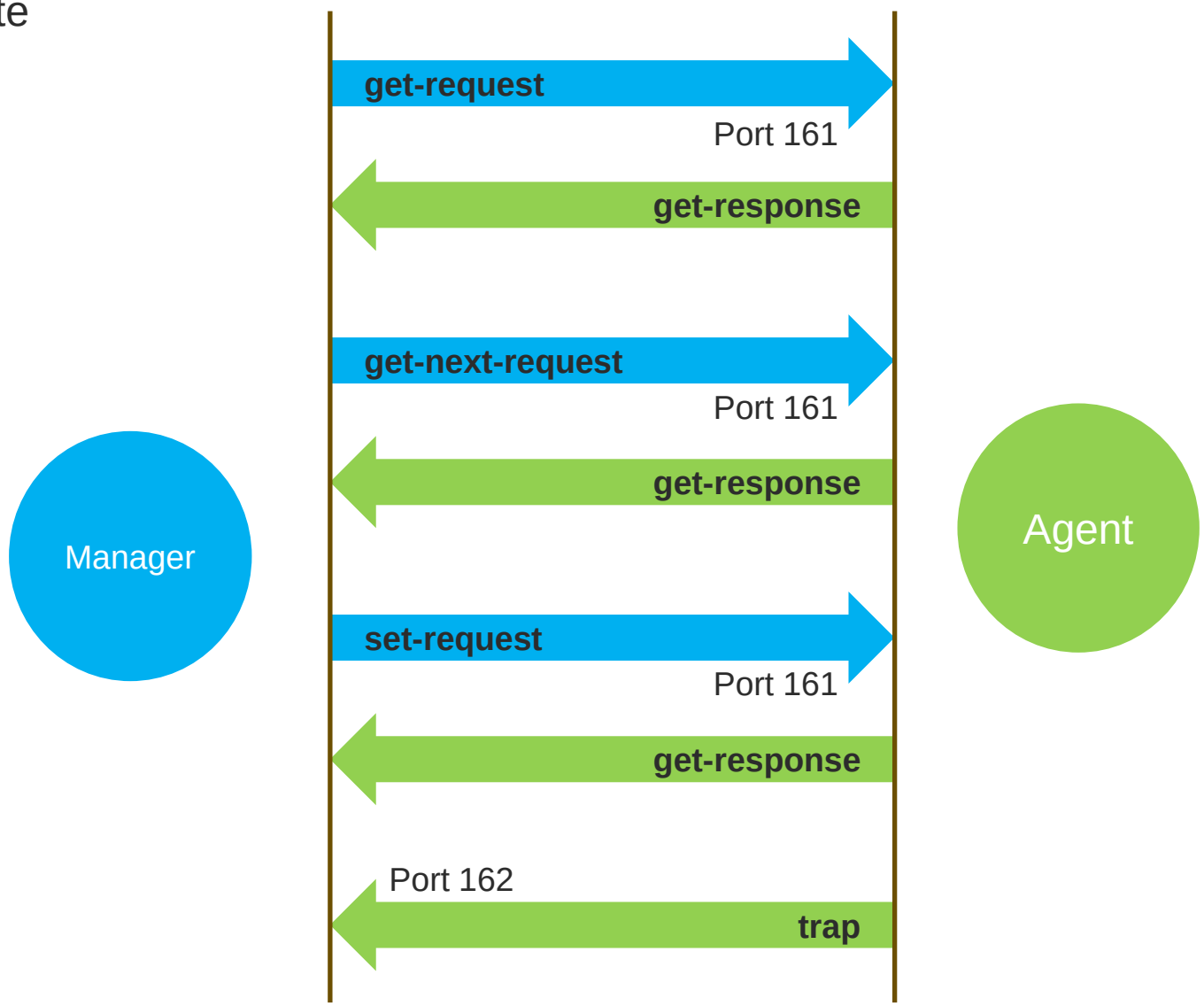
**Get-request.** Requests the values of one or more objects

**Get-next-request.** Requests the value of the next object, according to a lexicographical ordering of OIDs.

**Set-request.** A request to modify the value of one or more objects

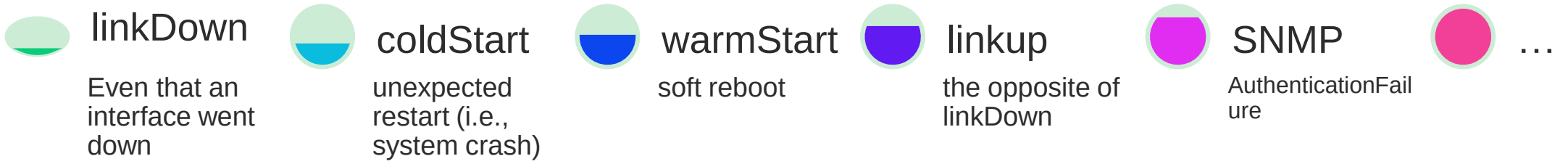
**Get-response.** Sent by SNMP agent in response to a *get-request, get-next-request, or set-request* message.

**Trap.** An SNMP trap is a notification sent by an SNMP agent to an SNMP manager, which is triggered by certain events at the agent.



# Traps

- Traps are messages that asynchronously sent by an agent to a manager
- Traps are triggered by an event
- Examples of Defined traps are



# SNMP Versions

- Three versions are in use today:

## SNMPv1

1990

## SNMPv2c

- 1996
- Adds “GetBulk” function and some new types
- Adds RMON (remote monitoring) capability

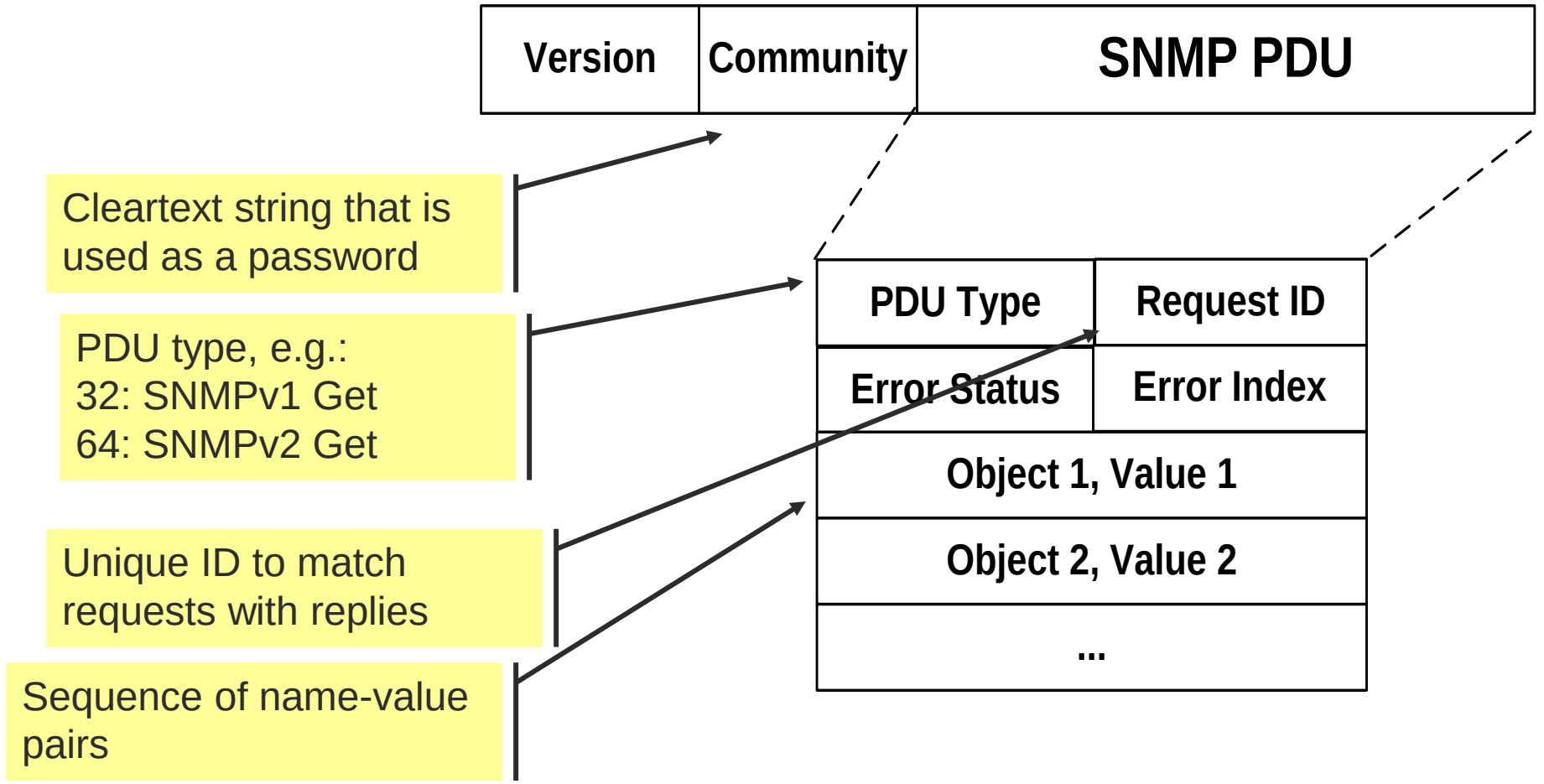
## SNMPv3

- 2002
- SNMPv3 started from SNMPv1 (and not SNMPv2c)
- Addresses security

- All versions are still used today
- Many SNMP agents and managers support all three versions of the protocol.

# Format of SNMP Packets

- SNMPv1 Get/Set messages:



# SNMP Security

- SNMPv1 uses plain text community strings for authentication as plain text without encryption

- SNMPv2 was supposed to fix security problems, but effort de-railed (The “c” in SNMPv2c stands for “community”).

- SNMPv3 has numerous security features:

  - Ensure that a packet has not been tampered with (**integrity**),
  - Ensures that a message is from a valid source (**authentication**)
  - Ensures that a message cannot be read by unauthorized (**privacy**).

# SNMP Security

- Security model of SNMPv3 has two components:

  1.Instead of granting access rights to a community, SNMPv3 grants access to users.

  2. Access can be restricted to sections of the MIB (V*ersion-based Access Control Module* (VACM). Access rights can be limited

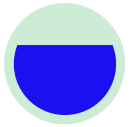
  - by specifying a range of valid IP addresses for a user or community,
  - or by specifying the part of the MIB tree that can be accessed.

# Security levels in SNMPv2

- SNMP has three security levels:

*noAuthNoPriv*

Authentication with matching a user name



*authNoPriv*

Authentication with MD5 or SHA message digests

*authPriv*

Authentication with MD5 or SHA message digests, and encryption with DES encryption

- Compare this to SNMPv1 and SNMPv2c:
  - SNMPv1, SNMPv2: Authentication with matching a community string.

IT601 – System and Network Administration

# Email Services

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

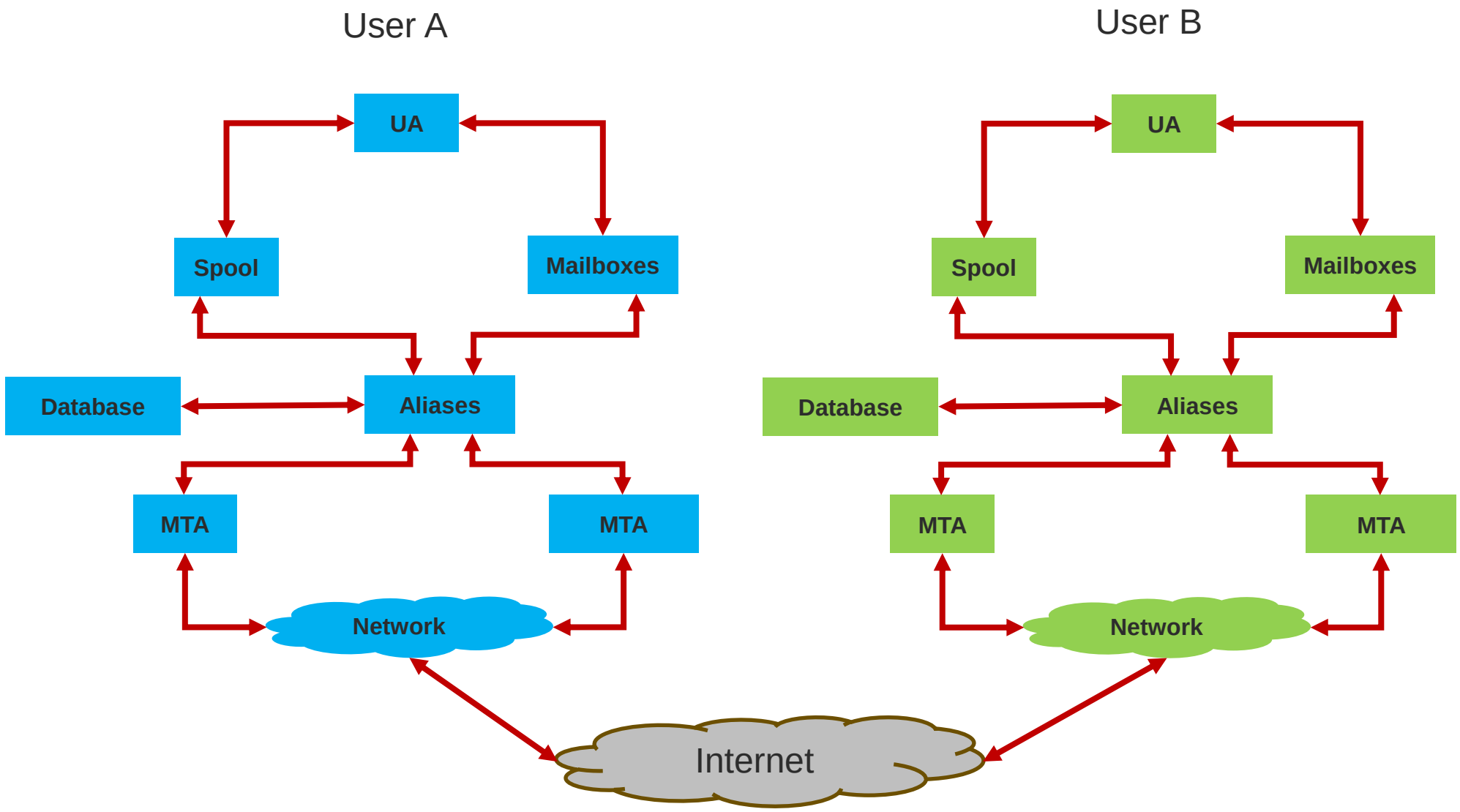
# Email Service

- Email (electronic mail) is a way to send and receive messages across the Internet. It's similar to traditional mail, but it also has some key differences.
- Elements of Email

| Addresses | Delivery | Time |
|---|---|---|
| • Traditional mail service uses postal addresses that consists of name, house no, street address, city, state and country.<br>• Sender postal address<br>• Receiver postal address<br>• Different formats | • Traditional mail is sent as sealed envelope and dropped in local mailbox.<br>• Sender -> sMailbox -> rMailbox -> Receiver<br>• User physical transport means and involves the courier services | • Takes time to reach destination. |
| • Email address use a standard format.<br>  • Standard format consist of username and email server address joined by @.<br>  • Uses passwords or other authentication mechanisms | • Email is sent from sender device to sender mailbox on providers premises.<br>• Mailboxes implements IMAP or POP3 to retrieve emails.<br>• senderDevice ->sSMTP ->rSMTP<br>• Transmission is encrypted | • Instantly delivered to receiver |

# Email Architecture

- **Components**
  - Email Client
  - Email Server
  - Mailboxes
  - MTA
  - Spool
  - Network
- **Protocols**
  - SMTP
  - SSL/TLS
  - IMAP
  - POP3
  - IP



# Email Service Components

- Normally a program which is used to send and receive mail
- Examples are Outlook, Thunderbird etc.
- Should support Email Operations

## Email Client

- This file contains mails that are to be sent.
- User agent appends outgoing mails in this file using SMTP.
- MTA extracts pending mail from spool file for their delivery.
- E-mail allows one name, an alias, to represent several different e-mail addresses.
- It is known as mailing list, Whenever user have to sent a message, system checks recipient's name against alias database.

## Spool

- responsible for transfer of mail from one system to another.
- To send a mail, a system must have client MTA and system MTA.
- The delivery from MTA to another MTA is done by SMTP Protocol

## MTA

- it is used as a blanket term for both mail transfer agents (MTA) and mail delivery agents (MDA), each of which perform a slightly different function

## Mail Server

- A local file to collect mails.
- Delivered mails are in this file.
- To use e-mail system Users must have a mailbox .
- Access to mailbox is only to owner of mailbox.

## Mailbox

MTA used IP networks to deliver email.

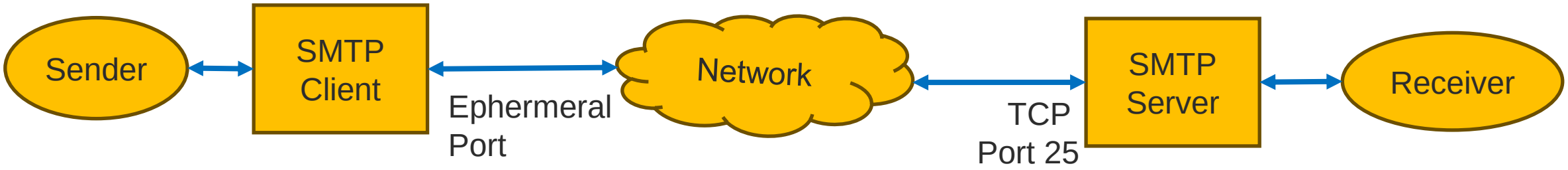
## Network

# Email Service Protocols

| SMTP | POP | IMAP | SSL/TLS | IP |
|---|---|---|---|---|
| • SMTP stands for Simple Mail Transfer Protocol.<br>• SMTP is the principal email protocol that is responsible for the transfer of emails between email clients and email servers. | • POP stands for Post Office Protocol.<br>• Email clients use the POP protocol support in the server to download the emails.<br>• This is primarily a one-way protocol and does not sync back the emails to the server. | • IMAP stands for Internet Message Access Protocol.<br>• IMAP Protocol is used to sync the emails in the server with the email clients.<br>• It allows two-way sync of emails between the server and the email client, while the emails are stored on the server. | • SSL/TLS certificates are used to authenticate and encrypt emails | • IP protocol is underlyaing network protocols |

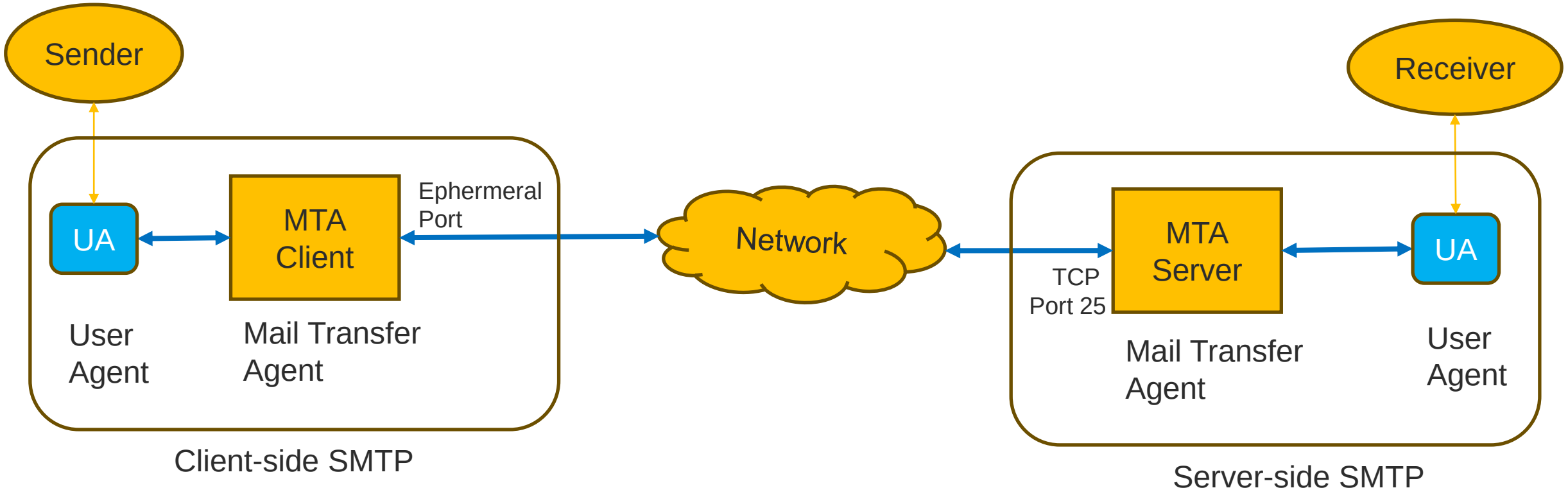
# SMTP Protocol

- It developed in 1982  in rfc0821 by Jon Postel
- Basic Purpose: To transfer mail reliably and efficiently
- Basic Architecture



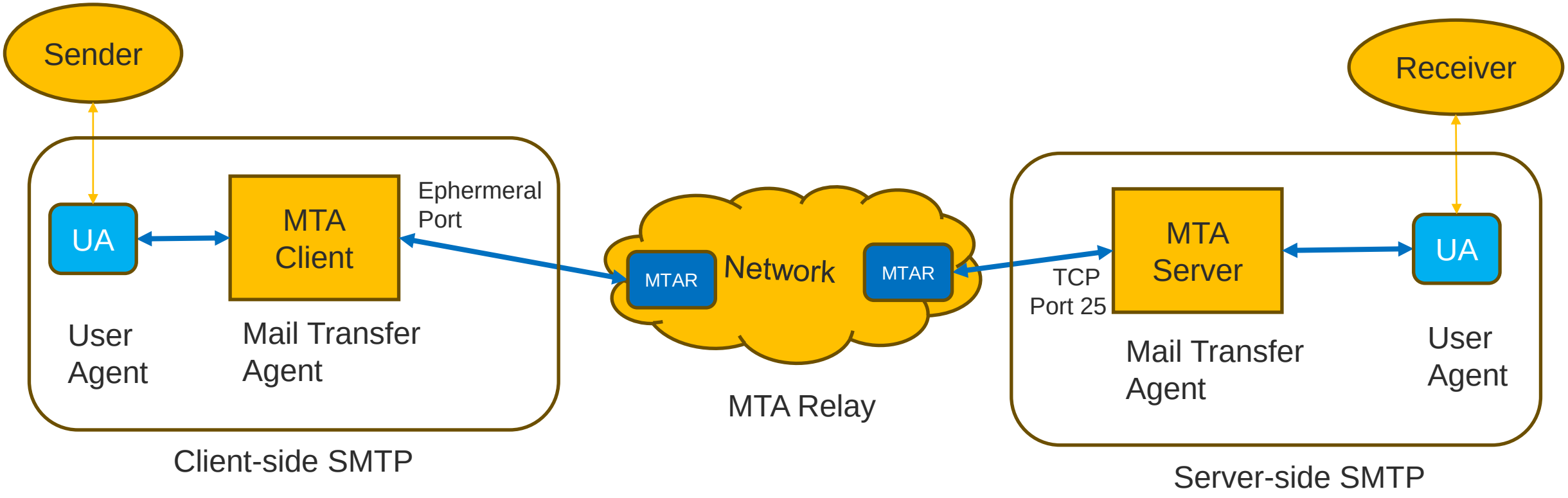
# SMTP : UA and MTA

- **SMTP clients and servers have two main components**
  - User Agents – Prepares the message, encloses it in an envelope. (Eudora for example)
  - Mail Transfer Agent – Transfers the mail across the internet
- **These components are required on both sides.**



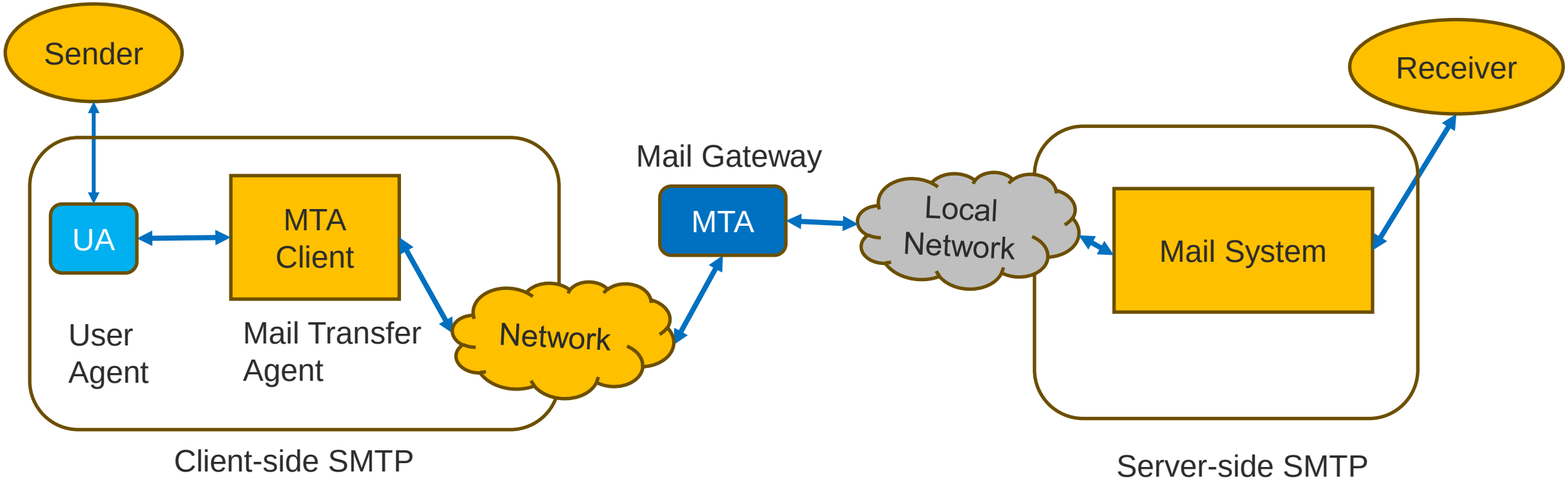
# SMTP Relaying

- **SMTP also allows the use of Relays allowing other MTAs to relay the mail**



# SMTP Gateway

- **Mail Gateways are used to relay mail prepared by a protocol other then SMTP and convert it to SMTP**



# Email Structure

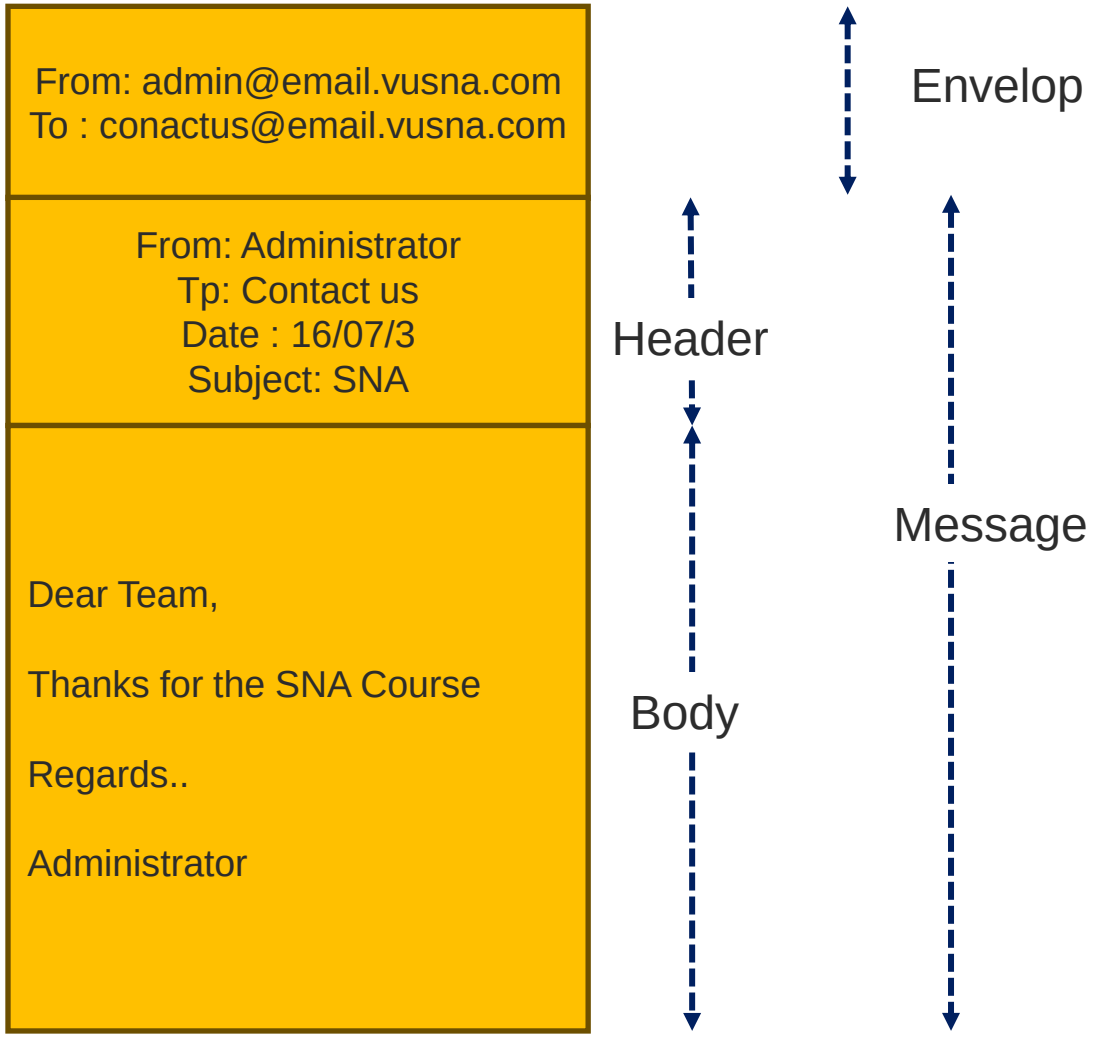
- Mail is a text file

Envelope –

- sender address
- receiver address
- other information

Message –

- Mail Header – defines the sender, the receiver, the subject of the message, and some other information
- Mail Body – Contains the actual information in the message



# Email Delivery Example

POP Mailbox

POP mail route

Receiver Mailbox

```
Return-Path: <admin@email.vusna.com>
Delivered-To: admin@email.vusna.com

Received: by email.vusna.com (Postfix, from userid 62)
        id 17FBD328DE; Wed,  16 July 2023 11:27:02
Received: from email.vusna.com
        (email.vusna.com [192.168.19.9])
        by email.vusna.com (Postfix) with ESMTP id
        5F41832893 for <admin@email.vusna.com>; Wed,  16    July 2023
        11:27:01
Received: by email.vusna.com (Postfix, from userid  62)id 47509456C;
Wed,  5 Nov 2003 11:27:01
Received: from email.vusna.com(email.vusna.com     [192.168.19.9])by
email.vusna.com (Postfix)          with SMTP id 7C2943D79 for <admin@
email.vusna.com>; Wed,  16 July 2023 11:26:34
Message-Id: <30031105162634.3C2943D79@ email.vusna.com>
Date: Wed,  5 Nov 2003 11:26:34
From: admin@email.vusna.com
To: admin@email.vusna.com: ;

MIME-Version: 1.0

Welcome to IT601P course.
```

# How SMTP works (A-PDU's)

- The Essentials Commands are

**HELO**

DNS of Sender host

Identifies the domain name of the sending host to SMTP

**MAIL FROM**

Sender Email Address

**RCPT TO**

Receiver Email Address

**DATA**

Message Body

**QUIT**

Close

# How SMTP works (A-PDU's)

- The Additional Commands are

**RSET**
Reset

**VRFY**
Verify Name

**NOOP**
No operation

Used for testing purpose

**TURN**
The roles of server and client to be reversed in a session

**EXPN**
Mailing list

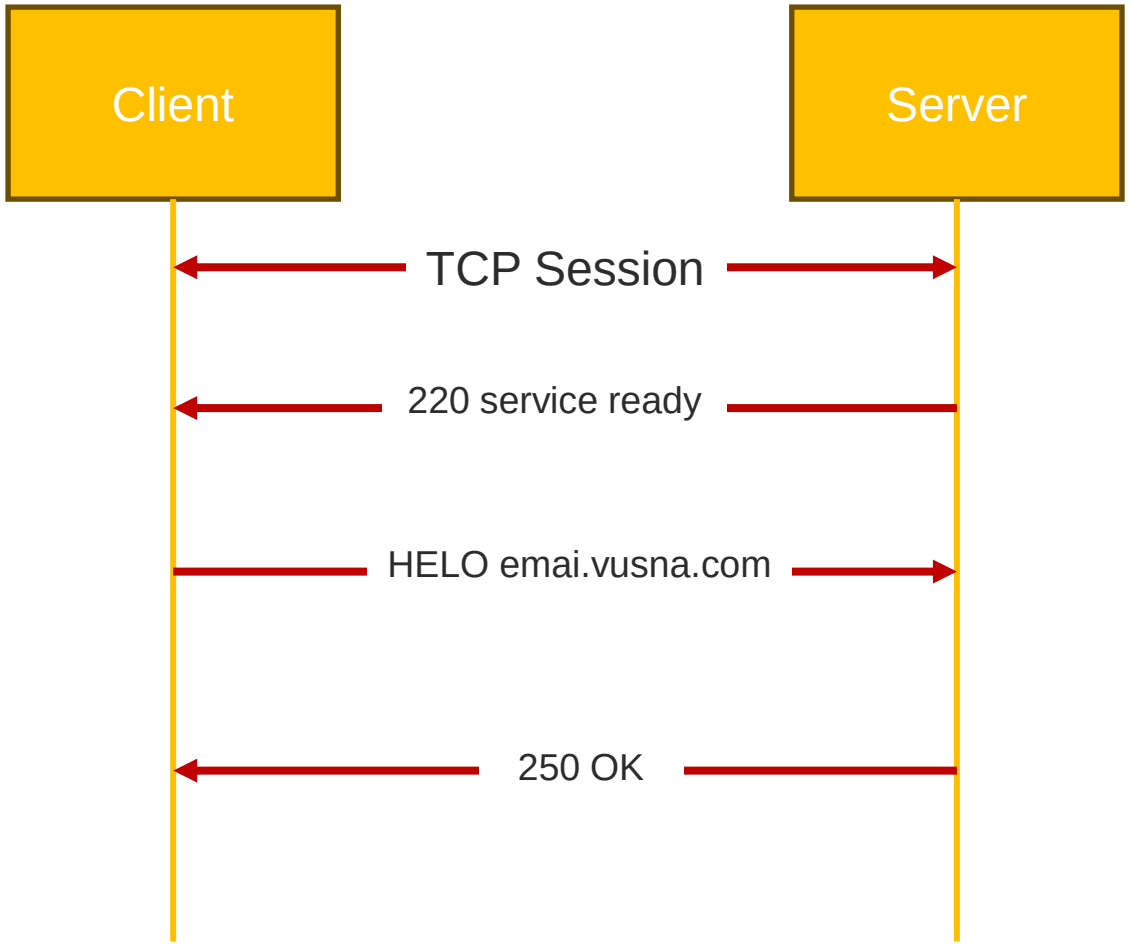
**HELP**
Command Name

# Status Codes

The Server responds with a 3 digit code that may be followed by text info

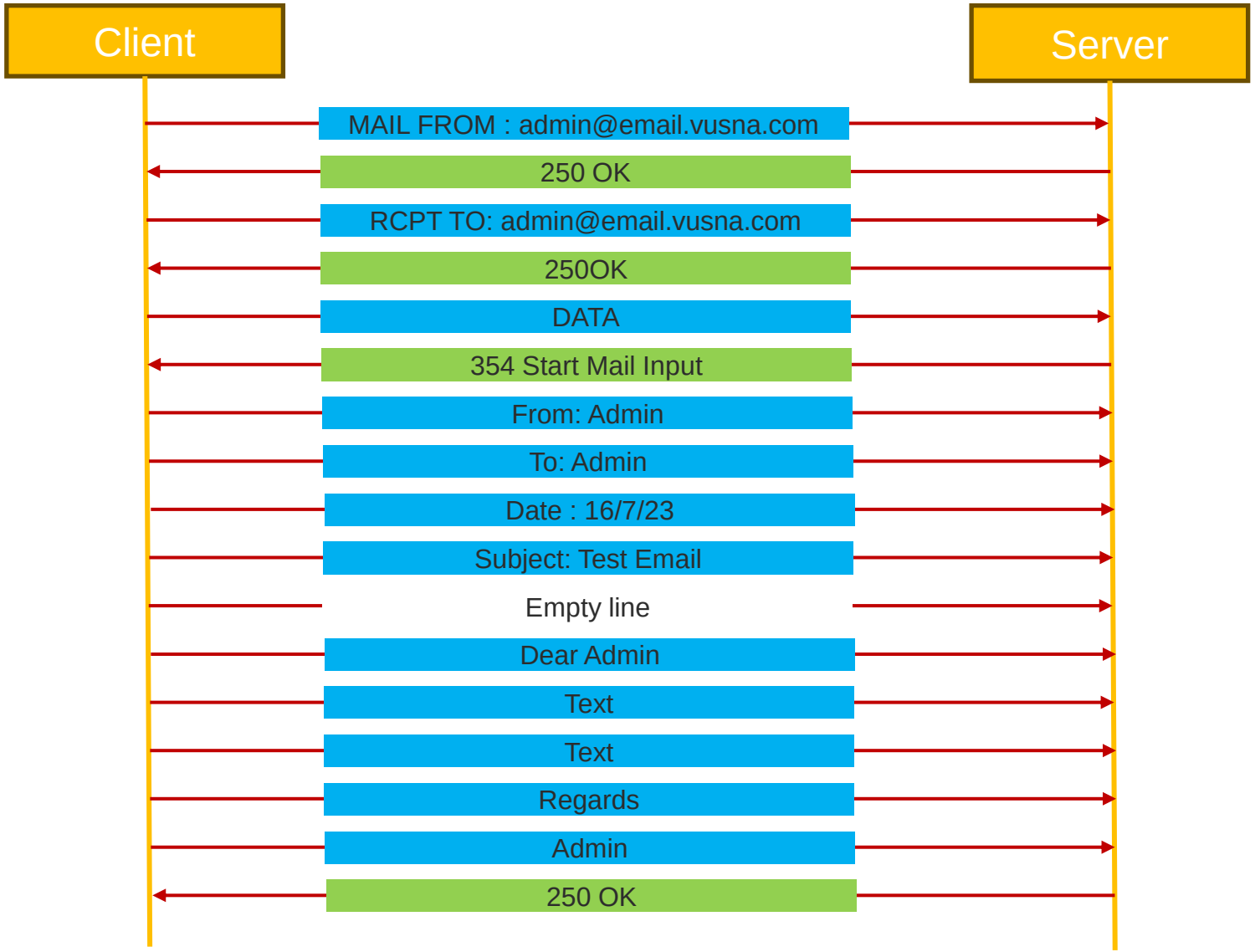
2## - Success
3## - Command can be accepted with more information
4## - Command was rejected, but error condition is temporary
5## - Command rejected, Bad User!

# SMTP MTA Connection Establishment

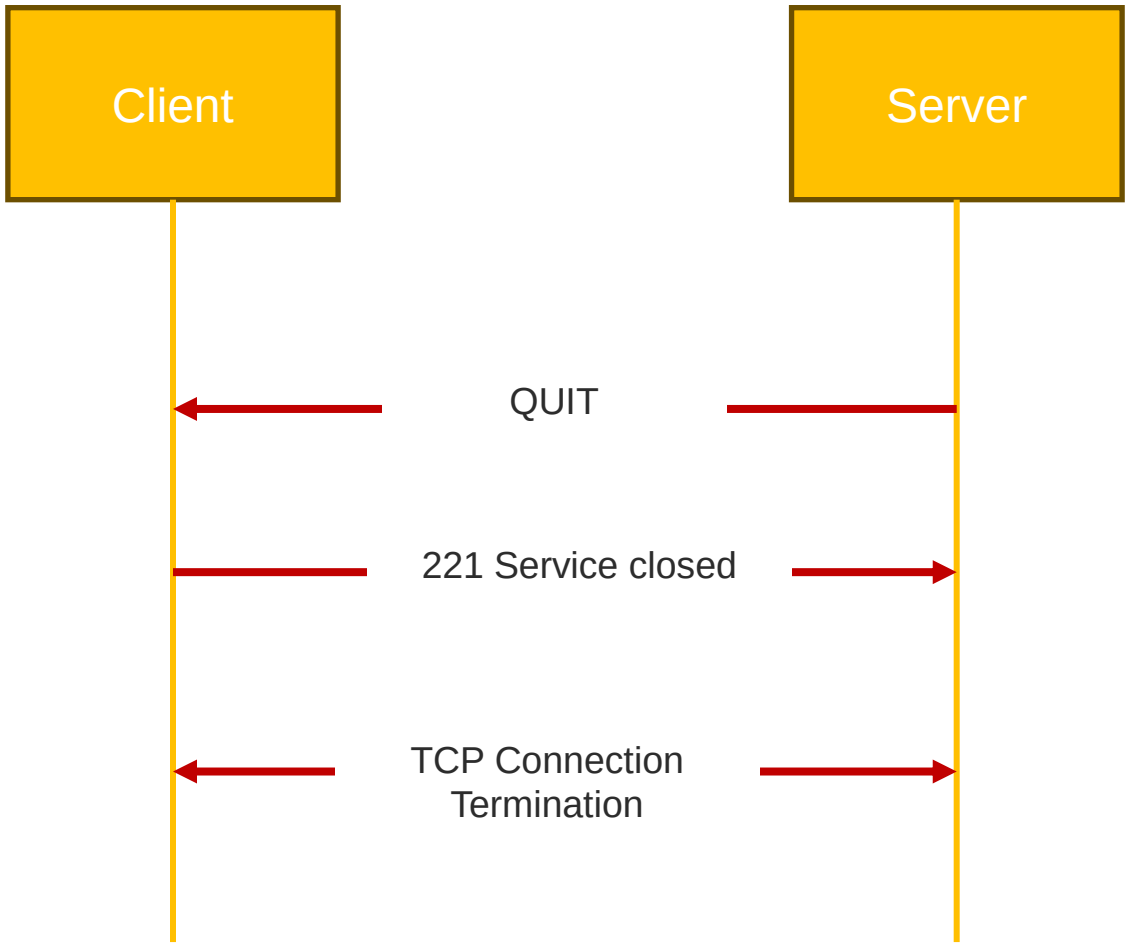


# Message Progress

- Email sending sequences are are followed.



# Connection Termination



# Problems with SMTP

- No security
  - Authentication
  - Encryption
- Possible Solutions:
  - VRFY command
  - Signature
- Only uses Non Virtual Terminal (NVT) 7 bit ASCII format
- Emails can easibly forged

# Extensions to SMTP

- MIME – Multipurpose Internet Mail Extensions
  - Transforms non-ASCII data to NVT (Network Virtual Terminal) ASCII data
    - Text
    - Application
    - Image
    - Audio
    - Video

# MIME Headers

- Goes between the Email Header and Body
  - MIME-Version: 1.1
  - Content-Type
  - Content-Transfer-Encoding
  - Content-Id
  - Content-Description

# MIME Headers

- Content-Type – Type of data used in the body of the message
  - Text – plain, unformatted text; HTML
  - Multipart – Body contains multiple independent parts
  - Message – The body is whole mail message, part of a message, or a pointer to a message

# MIME Headers

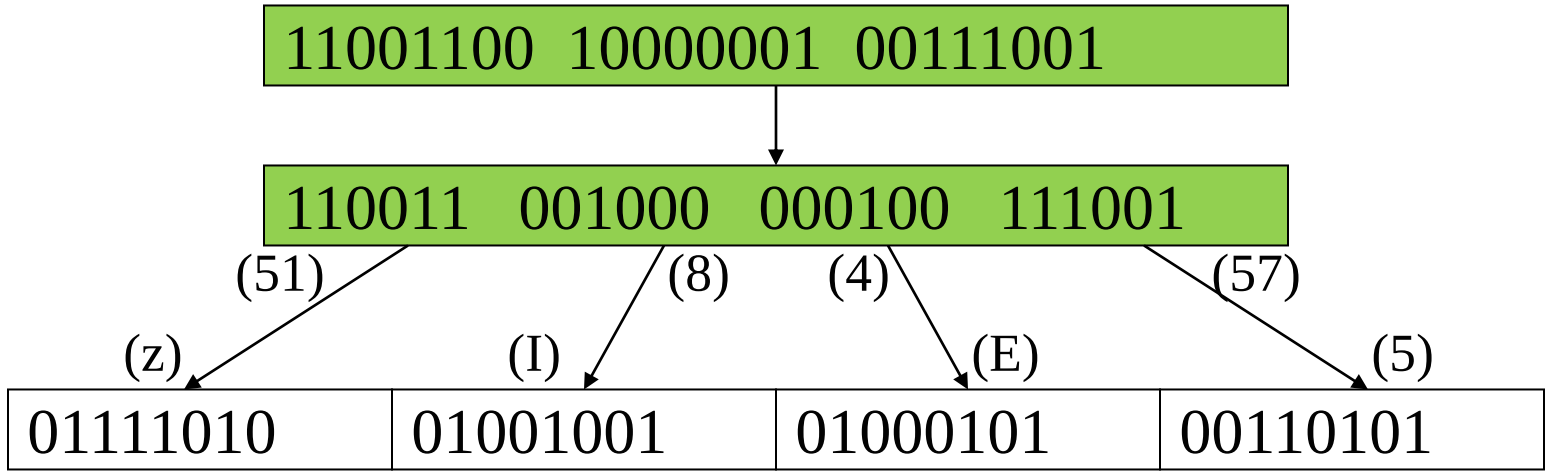
- Image – The message is a stationary  e.g image (JPEG or GIF)
- Video – The message is an animation  e.g Mpeg
- Audio – The message is 8 kHz standard e.g audio data
- Application – The message is a type of data not previously defined

# MIME Headers

- Content-Transfer-Encoding – The method used to encode the messages
  - 7 bit – no encoding needed
  - 8 bit – Non-ASCII, short lines
  - Binary – Non-ASCII, unlimited length lines
  - Base64 – 6 bit blocks encoded into 8-bit ASCII
  - Quoted-printable – send non-ASCII characters as 3 ASCII characters, =##, ## is the hex representation of the byte

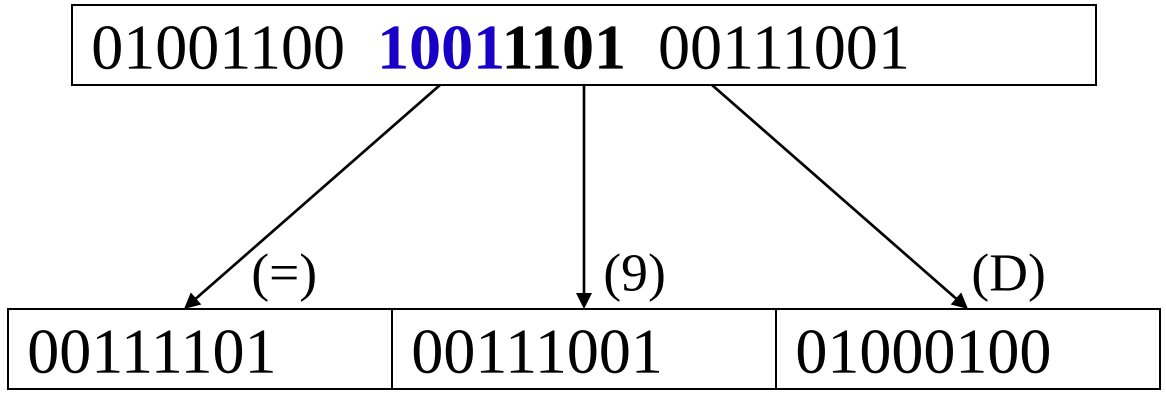
# Base64 Encoding

- Divides binary data into 24 bit blocks
- Each block is then divided into 6 bit chunks
- Each 6-bit section is interpreted as one character causes 25% overhead



# Quoted-Printable Encoding

- Used when the data has a small non-ASCII portion
- Non-ASCII characters are sent as 3 characters
- First is ‘=‘, second and third are the hex representation of the byte



# MIME Headers

- The following headers are defined in MIME:

**MIME-Version**

MIME 1.0

**Content-Type**

Contains media type
And its subtype

**Content-Transfer-Encoding**

Base64
quoted-printable
7bit/8bit – no enc

**Content-ID**

identify body parts

**Content-Disposition**

how to present a message or a body part

**Content-Description**

Description of content

**Content-Base**

Special header

**Content-Location**

Special header

# Example Multipart, Encoded MIME Message

From: admin@email.vusna.com
To: admin@email.vusna.com
Subject: Test Message
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=17

- 17
Content-Type: text/enriched; charset="us-ascii"
Content-Transfer-Encoding: 8bit
Content-Description: Greetings
Welcome to IT601P

- 17
Content-Type: application/octet-stream
Content-Transfer-Encoding: base64
Content-Description: Spec sheet saved as MS Word file
- 17 -

# Mail Transfer Agents

- MTAs do the actual mail transfers
- MTAs are not meant to be directly accessed by users.
- MMDF
- SENDMAIL

# Mail Access Protocols

- The MTAs place the email in the user’s mailbox
- The Mail Access Protocols are used by the users to retrieve the email from the mailbox
  - POP3
  - IMAP4

# Post Office Protocol v3

- Simple
- Allows the user to obtain a list of their Emails
- Users can retrieve their emails
- Users can either delete or keep the email on their system
- Minimizes server resources

# Internet Mail Access Protocol v4

- Has more features then POP3
- User can check the email header before downloading
- Emails can be accessed from any location
- Can search the email for a specific string of characters before downloading
- User can download parts of an email
- User can create, delete, or rename mailboxes on a server

# Linux Packages for Email Service

- Postfix
- Courier-imap
- Dovecot
- E.t.c

# Postfix

- It is Wietse Venema's mail server that started life at IBM research as an alternative to the widely-used Sendmail program. Now at Google, Wietse continues to support Postfix.

- Postfix attempts to be fast, easy to administer, and secure. The outside has a definite Sendmail-ish flavor, but the inside is completely different.

    - apt update
    - apt install postfix

# Postfix Configuration Types

**No configuration**

- The installation process will not configure any parameters.

**Internet Site**

- Use Postfix for sending emails to other MTAs and receiving email from other MTAs.

**Internet with smart host**

- Uses postfix to receive email from other MTAs but using another smart host to relay emails to the recipient.

**Satellite system**

- Uses smart host for sending and receiving email.

**Local only**

- Uses  emails are transmitted only between local user accounts.

# Authentication

- **Configure mailbox location**

```
sudo postconf -e 'home_mailbox = Maildir/'
```

- **SMTP authentication**

SMTP-AUTH allows a client to identify itself through the Simple Authentication and Security Layer (SASL) authentication mechanism, using Transport Layer Security (TLS) to encrypt the authentication process. Once it has been authenticated, the SMTP server will allow the client to relay mail.

- **Configure SMTP authentication**

```
sudo postconf -e 'smtpd_sasl_type = dovecot'
sudo postconf -e 'smtpd_sasl_path = private/auth'
sudo postconf -e 'smtpd_sasl_local_domain ='
sudo postconf -e 'smtpd_sasl_security_options = noanonymous,noplaintext'
sudo postconf -e 'smtpd_sasl_tls_security_options = noanonymous'
sudo postconf -e 'broken_sasl_auth_clients = yes'
sudo postconf -e 'smtpd_sasl_auth_enable = yes'
sudo postconf -e 'smtpd_recipient_restrictions = \
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination'
```

# TLS

- **Configure TLS**
  - Obtain a digital certificate for TLS. MUAs connecting to your mail server via TLS will need to recognize the certificate used for TLS. This can either be done using a certificate from Let's Encrypt, from a commercial CA or with a self-signed certificate that users manually install/accept.
  - For MTA-to-MTA, TLS certificates are never validated without prior agreement from the affected organizations.
  - For MTA-to-MTA TLS, there is no reason not to use a self-signed certificate unless local policy requires it. See our guide on security certificates for details about generating digital certificates and setting up your own Certificate Authority (CA).
  - Once you have a certificate, configure Postfix to provide TLS encryption for both incoming and outgoing mail:

```
sudo postconf -e 'smtp_tls_security_level = may'
sudo postconf -e 'smtpd_tls_security_level = may'
sudo postconf -e 'smtp_tls_note_starttls_offer = yes'
sudo postconf -e 'smtpd_tls_key_file = /etc/ssl/private/server.key'
sudo postconf -e 'smtpd_tls_cert_file = /etc/ssl/certs/server.crt'
sudo postconf -e 'smtpd_tls_loglevel = 1'
sudo postconf -e 'smtpd_tls_received_header = yes'
sudo postconf -e 'myhostname = mail.example.com'
```

# SASL

Postfix supports SMTP-AUTH as defined in RFC2554. It is based on SASL. However it is still necessary to set up SASL authentication before you can use SMTP-AUTH.

**Configure SASL**

Postfix supports two SASL implementations: Cyrus SASL and Dovecot SASL.

To enable Dovecot SASL the dovecot-core package will need to be installed:

```
sudo apt install dovecot-core
```

# Configure SASL

Next, edit /etc/dovecot/conf.d/10-master.conf and change the following:

```
service auth {
unix_listener auth-userdb {
    #mode = 0600
    #user =
    #group =
  }

  # Postfix smtp-auth
  unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    user = postfix
    group = postfix
  }
 }
```

auth_mechanisms = plain login

sudo systemctl restart dovecot.service

IT601 – System and Network Administration

# Database Services

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Database Services

- **Database is essential to any web-based application for saving records and user data. A database is an organized collection of data so that it can be easily accessed. To manage these databases, Database Management Systems (DBMS) are used.**

- **Types of DBMS**

  In general, there are two common types of databases:
  - Non-Relational
  - Relational

- **Non-Relational Database Management System (Non-RDBMS)**
  - In Non-RDBMS, data is stored in key-value pairs. For example:

    ```
    Customers = [ {id: 1, name=furqan, age=40}, {id=2,name=aisha,age=22}]
    ```

  - Commonly used Non-RDBMS: MongoDB, Amazon DynamoDB, Redis, ES etc.

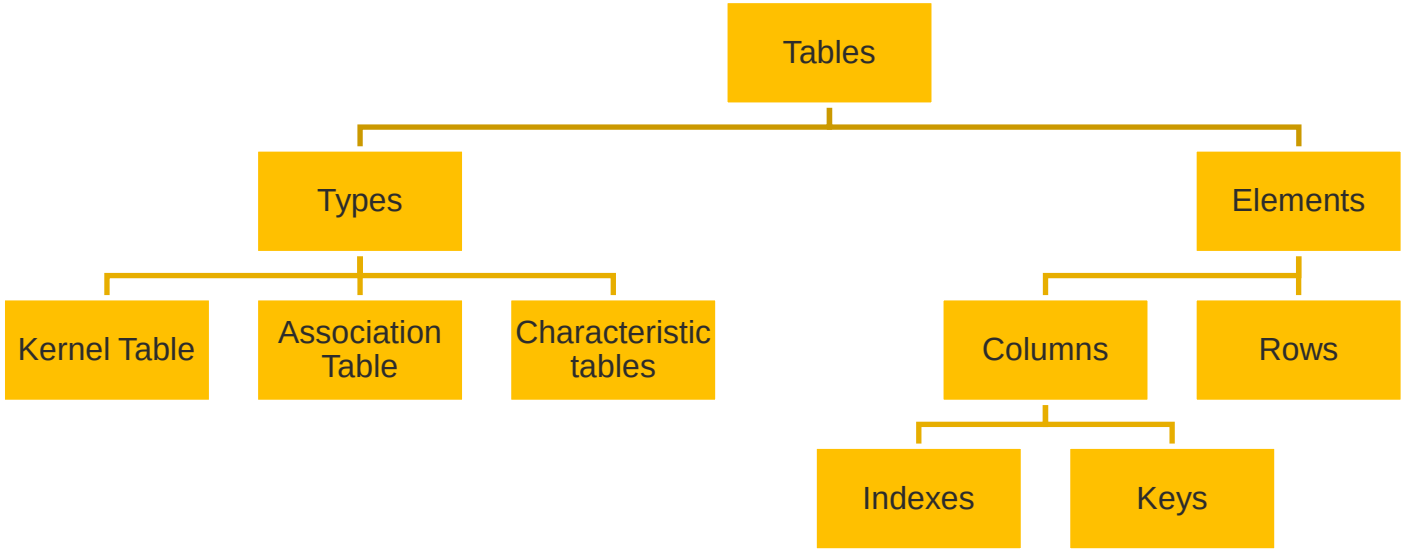
- **In RDBMS, data is stored in tabular format. For example,**

| ID | Name | Age |
|---|---|---|
| 1 | Furqan | 40 |
| 2 | Aisha | 22 |

  - Commonly used RDBMS: MySQL, PostgreSQL, MSSQL, Oracle etc.

# Elements of Relational Databases

- Relational databases are based on the relational model.
  - The relational model is a group of rules set forth by E. F. Codd based on mathematical principles (relational algebra), and it defines how database management systems should function.
  - The basic structures of a relational database (as defined by the relational model) are tables, columns (or fields), rows (or records), and keys.



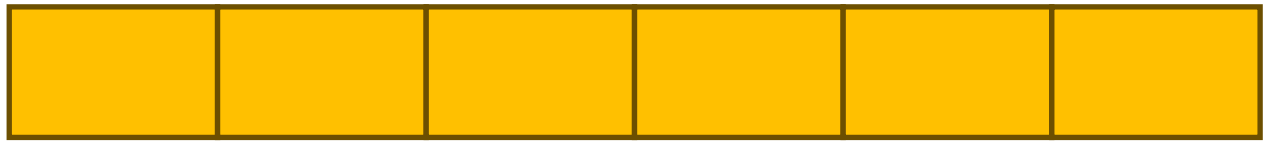
# Types of Tables

Tables are generally grouped into three types:

| Kernel tables | Association tables | Characteristic tables |
| --- | --- | --- |
| • Tables that are independent entities. Kernel tables often represent or model things that exist in the real world.<br>• Some example kernel tables are customers, vendors, employees, parts, goods, and equipment. | • Tables that represent a relationship among entities.<br>• For example, an order represents an association between a customer and goods. | • Tables whose purpose is to qualify or describe some other entity.<br>• Characteristic only have meaning in relation to the entity they describe.<br>• For example, order-lines might describe orders; without an order, an order-line is useless. |

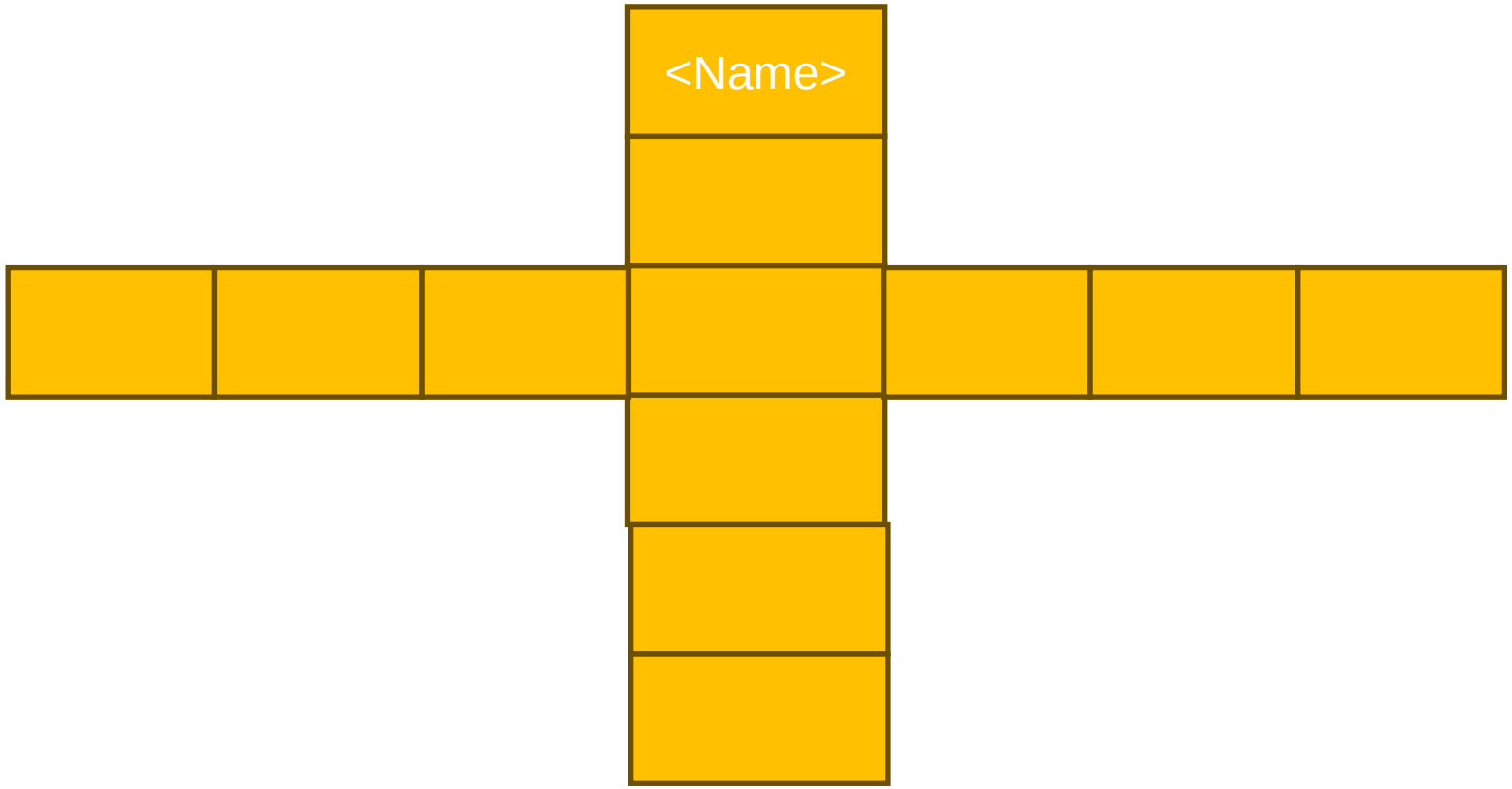
# Elements of Tabe : Row

- A row is a single occurrence of the data contained in a table; each row is treated as a single unit. In the Customer table image in Tables, there are four rows, and each row contains information about an individual customer.



# Elements of Tabe : Column

- Rows are organized as a set of columns (or fields). All rows in a table comprise the same set of columns. In the Customer table image in Tables, the columns are Cust Number, Name, and Street.



# Elements of Tabe : Keys

- **Keys identify a unique row**

## Primary key

- A primary key is a column (or group of columns) whose value uniquely identifies each row in a table. Because the key value is always unique, you can use it to detect and prevent duplicate rows.
- A good primary key has the following characteristics:

❑ mandatory
❑ unique
❑ stable
❑ short

## Foreign Key

- A foreign key is a column value in one table that is required to match the column value of the primary key in another table.
- If the foreign key value is not null, then the primary key value in the referenced table must exist.
- It is this relationship of a column in one table to a column in another table that provides the relational database with its ability to join tables.

# Elements of Tabe : Indexes

- An index in a database operates like the index tab on a file folder. It points out one identifying column, such as a customer's name, that makes it easier and quicker to find the information you want.
  - A single column can be used to define a simple index, or a combination of columns to define a composite or compound index.
  - To decide which columns to use, you first need to determine how the data in the table is accessed.
  - If users frequently look up customers by last name, then the last name is a good choice for an index. It is typical to base indexes on primary keys

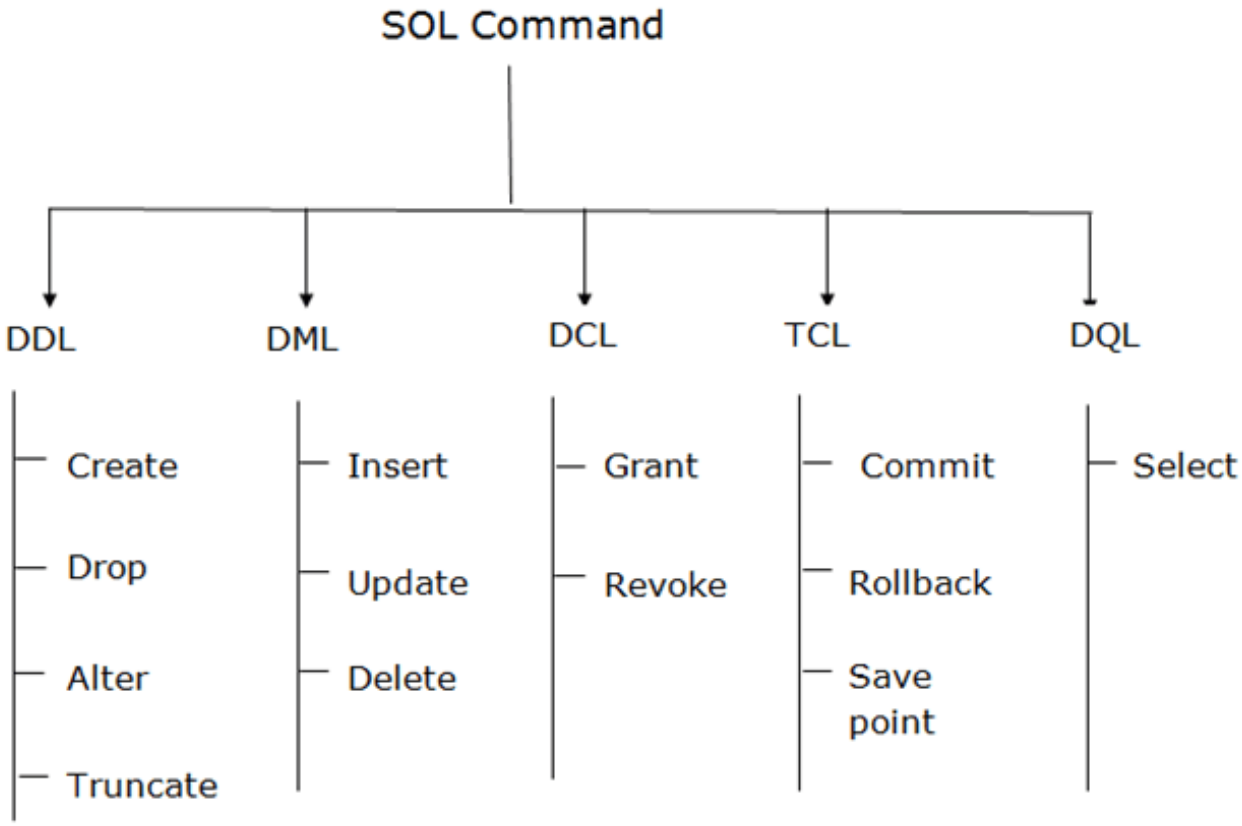
# Database Schema

- **A database schema defines how data is organized within a relational database.**
  - It covers logical constraints such as, table names, fields, data types, and the relationships between these entities.
- **Schemas commonly use visual representations to communicate the architecture of the database, becoming the foundation for an organization's data management discipline.**
- **Types of database schemas**

| Conceptual schema | Logical schema | Physical schema |
|---|---|---|
| • It provides a big-picture view of what the system will contain, how it will be organized, and which business rules are involved.<br>• Conceptual models are usually created as part of the process of gathering initial project requirements. | • It is less abstract, compared to conceptual schemas.<br>• It clearly define schema objects with information, such as table names, field names, entity relationships, and integrity constraints. | • It provide the technical information that the logical database schema type lacks in addition to the contextual information, such as table names, field names, entity relationships, et cetera. |

# Strcutured Query Language

- **Structured Query Language (SQL) is a standard query language that is used to work with relational databases.**
  - The SQL  is used to perform several operations.
  - SQL queries are general divided into 5 classes
    - Data Definition Language (DDL)
    - Data Manipulation Language (DML)
    - Data Control Language(DCL)
    - Transaction Control Language(TCL)
    - Data Query Language (DQL)




- **Example : SELECT first_name, last_name FROM Customers;**

- SQL is generally used with relational databases, however there is no standard way of using non-relational databases

# MYSQL

- MySQL is a very popular open-source relational database management system (RDBMS).

  - MySQL is a relational database management system
  - MySQL is open-source
  - MySQL is free
  - MySQL is ideal for both small and large applications
  - MySQL is very fast, reliable, scalable, and easy to use
  - MySQL is cross-platform
  - MySQL is compliant with the ANSI SQL standard
  - MySQL was first released in 1995
  - MySQL is developed, distributed, and supported by Oracle Corporation
  - MySQL is named after co-founder Monty Widenius's daughter: My

- Applications

  - Huge websites like Facebook, Twitter, Airbnb, Booking.com, Uber, GitHub, YouTube, etc.
  - Content Management Systems like WordPress, Drupal, Joomla!, Contao, etc.
  - A very large number of web developers around the world

# Common RDBMS

- MySQL,
- Microsoft SQL Server
- Oracle
- Microsoft Access.

# Installing MYSQL Server

- **Installation MySQL Service**

```
sudo apt update
```

```
sudo apt install mysql-server
```

```
mysql --version
```

- **Secure MySQL Service**

```
sudo mysql_secure_installation
```

  - **Add a Dedicated MySQL User**

```
sudo mysql
```

```
mysql> CREATE USER ‘<username>’@’<hostname>' IDENTIFIED WITH authentication_plugin BY 'password';
```

```
mysql> CREATE USER ‘<username>’@’<hostname>' IDENTIFIED BY 'password';
```

# Installing MYSQL Server

- **Grant Privileges to Secure MySQL**

```
mysql> GRANT PRIVILEGE ON database.table TO 'username'@'host';
```

```
GRANT CREATE, ALTER, DROP, INSERT, UPDATE, DELETE, SELECT, REFERENCES, RELOAD
on *.* TO ‘username'@'localhost' WITH GRANT OPTION;
```

```
FLUSH PRIVILEGES;
```

- **Managing MySQL Service**

```
systemctl status mysql.service
```

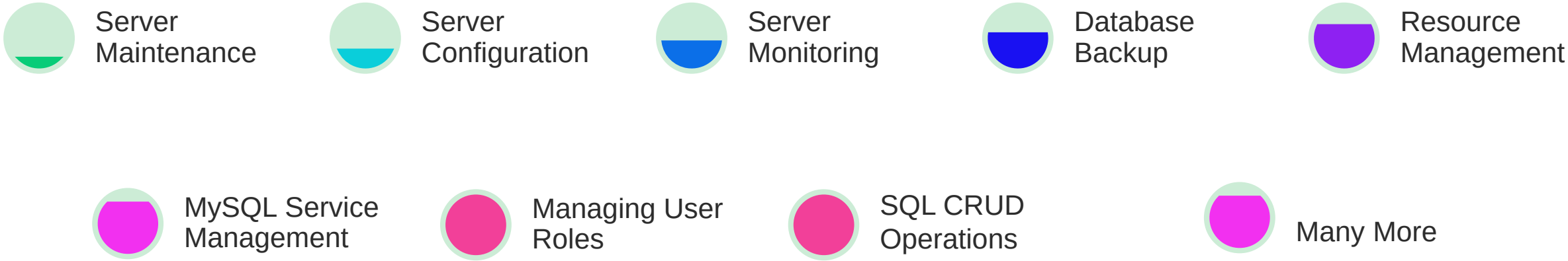
```
systemctl start|restart|enable mysql.service
```

- **Log in to your MySQL Server**

```
sudo mysql -u root
```

# Use of mysqladmin

- mysqladmin package used by Database Administrators to easily perform basic tasks in MySQL.
- It has several valuable tools which can be used for

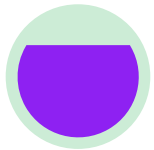


# Common Tasks Performed with mysqladmin

- mysqladmin has a controlled set of procedures and workflow. It can perform Database operations and queries with the help of standard and easy-to-use Structured Query Language (SQL).
  - It assigns users permissions to work on the Database Server Management and Maintenance activities.
  - Here are some of the important tasks that can be performed with mysqladmin.

Monitor MySQL Server processes

Create and drop Databases in MySQL Server

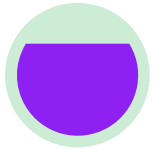


Flush information logs, statistics, status variables, and tables

Reload/reset MySQL privileges

Kill running queries

Start and stop the server with backups



Start and stop replicas

Check server configuration and status

# Benefits of mysqladmin

- Below are some of the advantages of mysqladmin.

  - Provides defined and improved structure of settings vital for the performance of the MySQL Server.
  - Visualize the flow by displaying a graphical representation, making it easy for users to read, interpret, and fine-tune the settings of the MySQL Server.
  - Takes care of Security Risk Management, hence you can feel super safe working around your data.
  - Easily import and export data files from the MySQL Server depending on the limited file size.
  - It maintains User Accounts, their Passwords, and is also capable of locking or unlocking users whenever needed.
  - It provides open-source flexibility and secure transactional support with high scalability and continuous uptime.

# Administering MYSQL with mysqladmin

| | |
|---|---|
| **Create a new Database.** | **create db_name** |
| Delete a Database. | drop db_name |
| Check the status of all MySQL Server variables. | extended-status |
| Flush all information in the host cache. | flush-hosts |
| **Flush all tables.** | refresh |
| **Set a new password.** | password new_password |
| **Stop the server.** | shutdown |
| **Display the server variables.** | variables |
| **Change the MySQL Root Password** | mysqladmin -u root password [New_Password] |
| **Change the MySQL Root Password** | mysqladmin -u root -pOld_Password password 'New_Password' |
| **see if your MySQL server is up and running.** | mysqladmin -u root -pPassword ping |
| **Check MySQL Server Uptime** | mysqladmin -u root -pPassword status |
| **check the version of the MySQL server** | mysqladmin -u root -pPassword version |
| **Check the Status of a MySQL Server** | mysqladmin -u root -pPassword status |
| **Extended Status of a MySQL Server** | mysqladmin -u root -pPassword extended-status |
| **Check MySQL Server Variables** | mysqladmin -u root -pPassword variables |
| **Check the MySQL Process List** | mysqladmin -u root -pPassword processlist |
| **kill the MySQL client process** | mysqladmin -u root -pPassword kill 195003 |

# phpmyadmin

- Phpmyadmin is an  Web based alternative to mysqlamin. Its main features are
  - Security
    - It has some built-in security features, but it may not be as robust as Debian.
  - Stability
    - It is generally stable but may not be as reliable as Debian.
  - Compatibility
    - It may require additional steps to install and configure PHPMyAdmin on a separate system.
  - Familiarity
    - It may require some learning if not familiar with Ubuntu.

# Installaing PHPMYADMIN

- Update APT Repositories

```
sudo apt update
```

- Install PHP Support

```
sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

- Install phpMyAdmin

```
sudo apt install phpMyAdmin
```

```
phpenmod mbstring
```

- Restart Web service

```
sudo systemctl restart apache2
```

- Access via Web Interface http://<hostname>/phpmyadmin

# Non-Relational Databases

- Relational Databases are not suitable for large data, unstructured or semistrucutre data
  - NoSQL databases are non-relational databases and addresses above issues

- There are four major types of NoSQL Databases

| Document databases | Key-value stores | Column stores | Graph databases |
|---|---|---|---|
| • Store data as semi-structured documents, such as JSON or XML, and can be queried using document-oriented query languages | • Store data as key-value pairs, and are optimized for simple and fast read/write operations. | • These databases store data as column families, which are sets of columns that are treated as a single entity.<br>• Optimized for fast and efficient querying of large amounts of data. | • Store data as nodes and edges, and are designed to handle complex relationships between data. |

# Key Characteristics of NoSQL

- Key Characteristics of NoSQL databases are

### Dynamic schema

- Do not have a fixed schema and can accommodate changing data structures without the need for migrations or schema alterations.

### Horizontal scalability

- Designed to scale out by adding more nodes to a database cluster, making them well-suited for handling large amounts of data and high levels of traffic.

### Document-based

- Some NoSQL Dbs, such as MongoDB, use a document-based data model, where data is stored in semi-structured format, such as JSON or BSON.

### Key-value-based

- NoSQL based Redis, use a key-value data model, where data is stored as a collection of key-value pairs.

### Column-based

- Some NoSQL databases, such as Cassandra, use a column-based data model, where data is organized into columns instead of rows.

### Distributed and high availability

- NoSQL databases are often designed to be highly available and to automatically handle node failures and data replication across multiple nodes in a database cluster.

### Flexibility

- NoSQL databases allow developers to store and retrieve data in a flexible and dynamic manner, with support for multiple data types and changing data structures

### Performance

- NoSQL databases are optimized for high performance and can handle a high volume of reads and writes, making them suitable for big data and real-time applications.

# Merits/Demerits

- NoSQL has the following Benefits

### High scalability

- Use sharding for horizontal scaling
- Vertical Scaling Complex
- MongoDB, Cassandra are examples of horizontal scaling DBs.
- Handle a huge amount of data, as the data grows it scale itself to handle that data in an efficient manner

### Flexibility

- Designed to handle unstructured or semi-structured data, which means that they can accommodate dynamic changes to the data model.
- This makes it suitable for applications that need to handle changing data requirements.

### High availability

- Auto replication feature in NoSQL databases makes it highly available because in case of any failure data replicates itself to the previous consistent state.

### Performance

- Designed to handle large amounts of data and traffic, which means that they can offer improved performance compared to traditional relational databases.

### Cost-effectiveness

- More cost-effective than traditional relational databases, as they are typically less complex and do not require expensive hardware or software.

### Agility

- Ideal for agile development.

# Merits/Demerits

- NoSQL has the following demerits

| | | |
|---|---|---|
| **No standardization** | **No ACID compliance** | **Narrow focus** |
| **No complex queries** | **Not Mature** | **Complex Management** |

# Applications of NoSQL Databases

- NoSQL databases are often used in applications where there is a high volume of data that needs to be processed and analyzed in real-time, such as social media analytics, e-commerce, and gaming.
- They can also be used for other applications, such as content management systems, document management, and customer relationship management.
- NoSQL databases may not be suitable for all applications, as they may not provide the same level of data consistency and transactional guarantees as traditional relational databases.
- It is important to carefully evaluate the specific needs of an application when choosing a database management system.

# Available NoSQL Databases

- **Following are commonly available NoSQL based databases.**

| Graph Databases | Key value store | Tabular | Document-based |
|---|---|---|---|
| ➢Amazon Neptune<br>➢ Neo4j | ➢Memcached<br>➢Redis<br>➢Coherence | • Hbase<br>• Big Table<br>• Accumulo | • MongoDB<br>• CouchDB<br>• Cloudant<br>• Elasticsearch ? |

# Elasticsearch

- **To some people known as  "an index," "a search engine," "an analytics database," "a big data solution," "it's quick and scalable," or "it's like Google."**
  - All of above are correct, which is part of Elasticsearch's appeal.

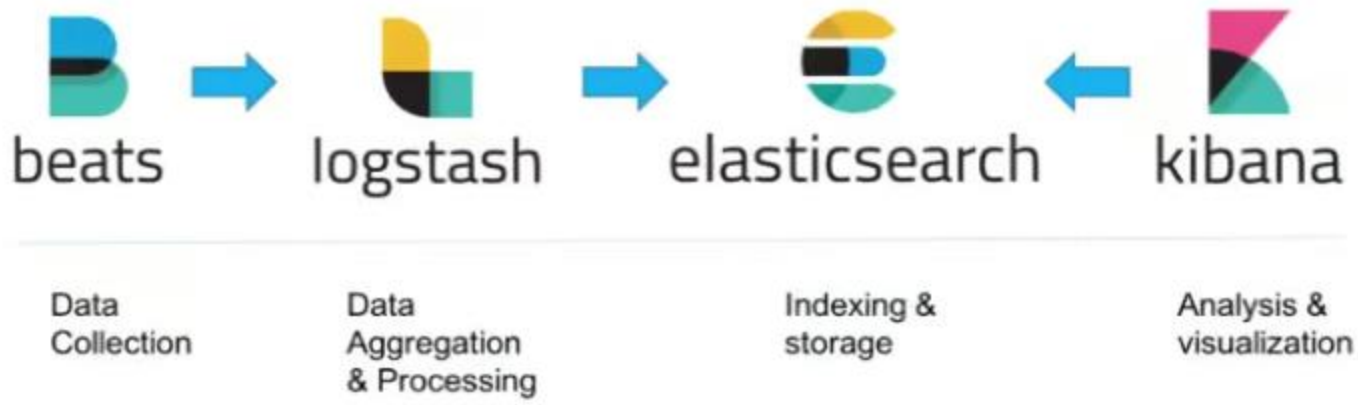
- **Elasticsearch is a distributed, free and open search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured.**
  - Elasticsearch is built on Apache Lucene and was first released in 2010 by Elasticsearch N.V. (now known as Elastic).

- **It is known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the central component of the Elastic Stack, a set of free and open tools for data ingestion, enrichment, storage, analysis, and visualization.**
  - Commonly referred to as the ELK Stack (after Elasticsearch, Logstash, and Kibana), the Elastic Stack now includes a rich collection of lightweight shipping agents known as Beats for sending data to Elasticsearch.

# Operations of ES

- Raw data flows into Elasticsearch from a variety of sources, including logs, system metrics, and web applications.
    - Data ingestion is the process by which this raw data is parsed, normalized, and enriched before it is indexed in Elasticsearch.

- Once indexed in Elasticsearch, users can run complex queries against their data and use aggregations to retrieve complex summaries of their data.



# Elasticsearch index

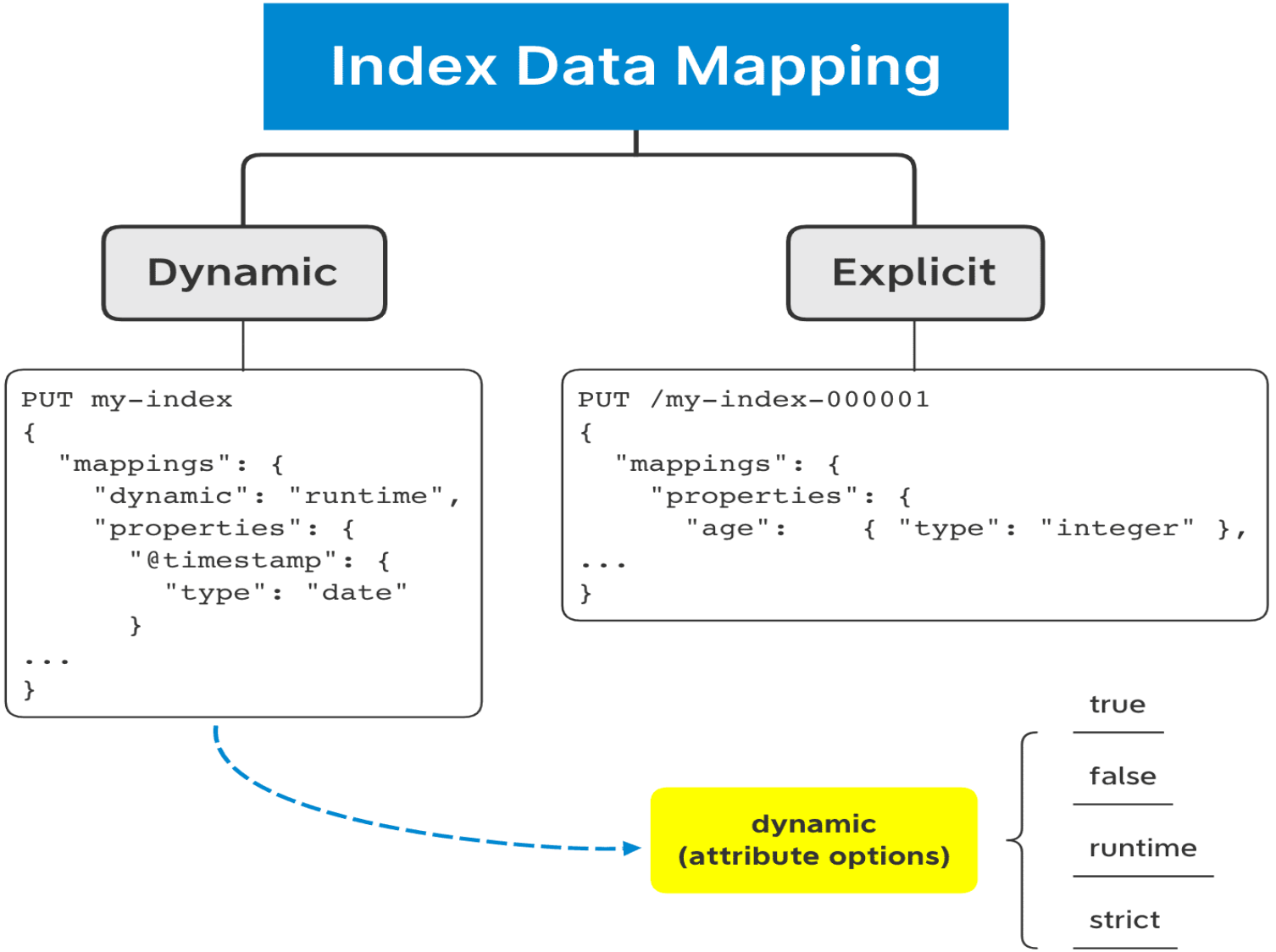
- **An Elasticsearch index is a collection of documents that are related to each other. Elasticsearch stores data as JSON documents.**
  - Each document correlates a set of keys (names of fields or properties) with their corresponding values (strings, numbers, Booleans, dates, arrays of values, geolocations, or other types of data).

- **Elasticsearch uses a data structure called an inverted index, which is designed to allow very fast full-text searches. An inverted index lists every unique word that appears in any document and identifies all of the documents each word occurs in.**

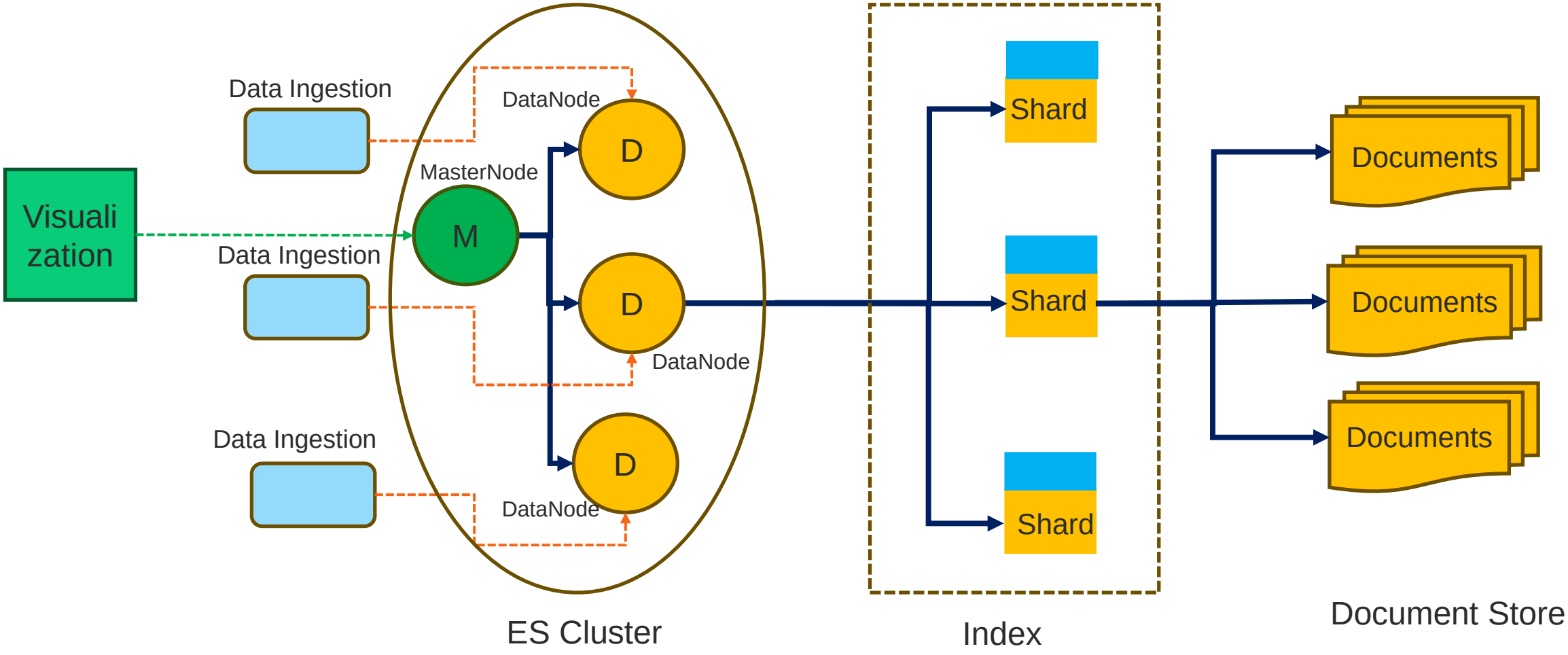
  - An inverted index is a mapping of each specific ‘word' (token) to the list of documents (locations) containing that word, allowing users to easily find documents containing given keywords. Index data is contained in one or more partitions, also defined as shards. Elasticsearch also automatically distributes and allocates shards to cluster nodes.

- **During the indexing process, Elasticsearch stores documents and builds an inverted index to make the document data searchable in near real-time. Indexing is initiated with the index API, through which you can add or update a JSON document in a specific index.**

# Elasticsearch index



# Architecture of Elasticsearch



# Basic Terms in ES

- Some basic terminology related to ES are

## JVM

- JVM allows running java programs on specified servers.

## Shard

- Shards, on the other hand, are the "Apache Lucene" application itself that provides indexing of data within nodes.

## Index

- Each record in ElasticSearch consists of JSON documents. Elasticsearch indexes is a collection of JSONson documents. In short, each index is a kind of database.

## Segment

- The Lucene index is split into parts, which are smaller directories. A segment is a subset of the Lucene index.

## Mapping

- Mapping is the method of specifying how a document and the fields contained inside it will be stored and indexed. Each index has a single mapping form that defines how the text is indexed.

## Node

- Any single instance (elasticsearch installed machine ) is defined as Node.

## Document

- In Elasticsearch a document represents a basic unit of information that can be indexed.

## Replica

- Elasticsearch sends a copy of each data to other machines, thus preventing data loss if one of the machines is down. Theise replicated machines or shards are defined as Replica.

## Cluster

- A cluster in Elasticsearch is a set of nodes with the same cluster namecluster .name attribute. As nodes join or exit a cluster, the cluster reorganizes itself to spread data equally over the available nodes.

## Type

- In Elasticsearch, a type represents a class of related documents and is identified by a name such as customer or product.

# Why use Elasticsearch?

- **Elasticsearch is fast.**
  - built on top of Lucene which excels at full-text search.
  - a near real-time search platform, Lowest Latency, typicaly 1s
    - A requirements for time-sensitive use cases such as security analytics and infrastructure monitoring
- **Elasticsearch is distributed by nature.**
  - The documents stored in Elasticsearch are distributed across different containers known as shards
  - Shards are duplicated to provide redundant copies of the data in case of hardware failure
  - scale out to hundreds (or even thousands) of servers and handle petabytes of data.
- **A wide set of features**
  - In addition to its speed, scalability, and resiliency, it has a number of powerful built-in features such as data rollups and index lifecycle management.
- **Simplified data ingest, visualization, and reporting**
  - Integration with Beats and Logstash makes it easy to process data before indexing into Elasticsearch.
  - Kibana provides real-time visualization of Elasticsearch data as well as UIs for quickly accessing application performance monitoring (APM), logs, and infrastructure metrics data.

# ES Applications

- Application search
- Website search
- Infrastructure metrics and container monitoring
- Application performance monitoring
- Enterprise search
- Geospatial data analysis and visualization
- Security analytics
- Logging and log analytics
- Business analytics

# ES Node Roles

➢ **The main Node roles of ES are as follows**

### Master

- A node that has the master role, which makes it eligible to be elected as the master node, which controls the cluster.

### Remote_cluster_client

- A node that has the remote_cluster_client role, which makes it eligible to act as a remote client.

### ml

- Allows to use machine learning features, there must be at least one machine learning node in your cluster.

### Ingest

- Ingest nodes are able to apply an ingest pipeline to a document in order to transform and enrich the document before indexing.
- With a heavy ingest load, it makes sense to use dedicated ingest nodes and to not include the ingest role from nodes that have the master or data roles.

### Data

- Data nodes hold data and perform data related operations such as CRUD, search, and aggregations. A node with the data role can fill any of the specialised data node roles.

- Data_content
- Data_hot
- Data_warm
- Data_cold
- Data_frozen

### Transform

- Used to transform data, there must be at least one transform node in your cluster.

# Data Ingestion

- **Data ingestion refers to the tools & processes used to collect data from various sources and move it to a target site, either in batches or in real-time.**
  - The data ingestion layer is critical to your downstream data science, BI, and analytics systems which depend on timely, complete, and accurate data.
- **ES provides following tools for data ingestion**

| Elastic Beats | Logstash | Language clients | Kibana Dev Tools |
|---|---|---|---|
| • Elastic Beats are a set of lightweight data shippers that allow to conveniently send data to Elasticsearch Service. | • Powerful and flexible tool to read, process, and ship data of any kind | • Python, ruby etc | |

# Elasticbeats

| | |
|---|---|
| Filebeat | ▪ Used to read, preprocess and ship data from sources that come in the form of log files.<br>▪ Filebeat further supports a number of other data sources including TCP/UDP, containers, Redis, and Syslog.<br>▪ Large No. of module ease collection and parsing of log formats for applications such as Apache, MySQL, and Kafka. |
| Metricbeat | ▪ Collects and preprocesses system and service metrics.<br>▪ System metrics include information about running processes, as well as CPU / memory / disk / network utilization numbers.<br>▪ Can collect data from many different services including Kafka, Palo Alto Networks, Redis, and many more. |
| Packetbeat | •Collects and preprocesses live networking data, therefore enabling application monitoring, as well as security and network performance analytics.<br>•Among others, Packetbeat supports the following protocols: DHCP, DNS, HTTP, MongoDB, NFS, and TLS. |

# Elasticbeats

**Winlogbeat**

- is all about capturing event logs from Windows operating systems, including application events, hardware events, and security and system events.
- The vast information available from the Windows event log is of much interest for many use cases.

**Auditbeat**

- detects changes to critical files and collects events from the Linux Audit Framework.
- Different modules ease its deployment, which is mostly used in the security analytics use cases.

**Heartbeat**

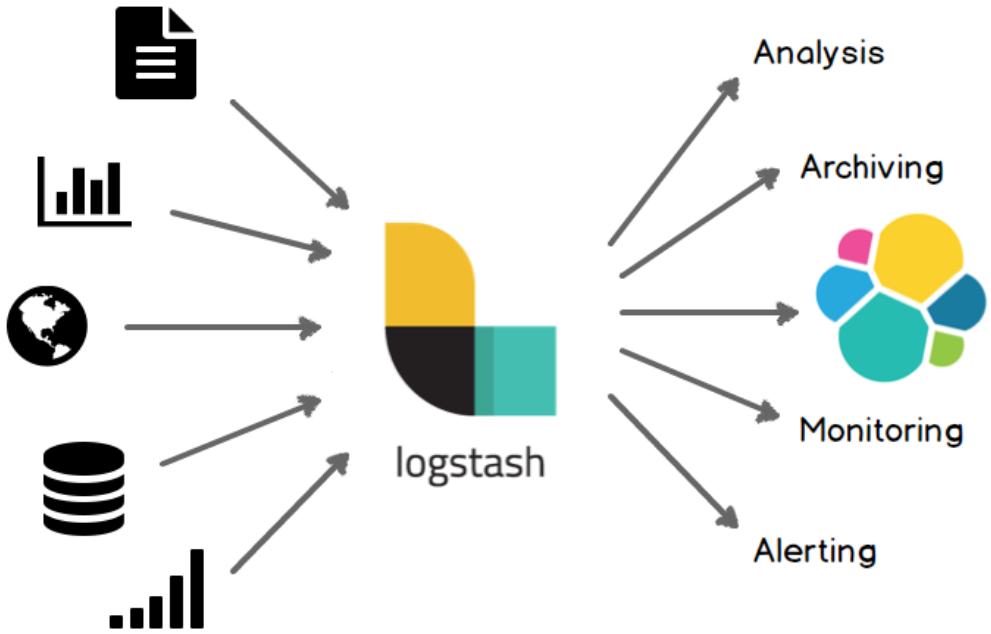
- uses probing to monitor the availability of systems and services.
- Heartbeat is useful in a number of scenarios such as infrastructure monitoring and security analytics. ICMP, TCP, and HTTP are supported protocols.

**Functionbeat**

- collects logs and metrics from within a serverless environment such as AWS Lambda.

# Logstash

- Logstash is a powerful and flexible tool to read, process, and ship data of any kind.
  - Logstash provides several capabilities that are not currently available or too costly to perform with Beats, such as enriching documents by performing lookups against external data sources.
- However, this functionality and flexibility of Logstash comes at a price. Also, hardware requirements for Logstash are significantly higher than for Beats.
  - Logstash should generally not be deployed on low-resource devices. Logstash is therefore used as an alternative to Beats, should the functionality of the latter be insufficient for a specific use case.

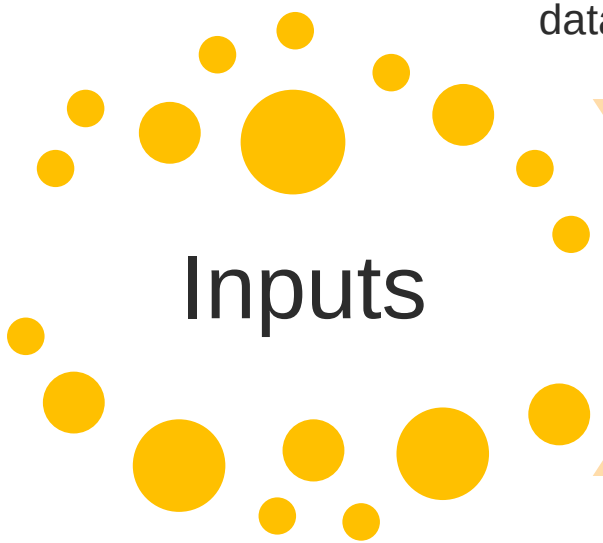



- A common architectural pattern is to combine Beats and Logstash: use Beats to collect data and use Logstash to perform any data processing that Beats are not capable of doing.

# Logstash Pipeline

- **Logstash works by executing event processing pipelines, whereby each pipeline consists of at least one of each of the following:**

- Process and enrich the data in various ways.
- Filters can parse CSV, JSON, key/value pairs, delimited unstructured data, and complex unstructured data on the basis of regular expressions (grok filters).



Inputs

Filters

Outputs

- Read from data sources such as files, http, imap, jdbc, kafka, syslog, tcp, and udp.

- Enrich data by performing DNS lookups, adding geoinformation about IP addresses, or by performing lookups against a custom dictionary or an Elasticsearch index.

- Additional filters allow for diverse transformations of the data, for example, to rename, remove, copy data fields and values (mutate filter).

- write the parsed and enriched data to data sinks and are the final stage of the Logstash processing pipeline.

- While many output plugins are available, here we focus on ingestion into Elasticsearch Service using the Elasticsearch output.

# Logstash Example Pipeline

**input**

```
input {
  rss {
    url => "/blog/feed"
    interval => 120
  }
}
```

**Transformations**

```
filter {
  mutate {
    rename => [ "message", "blog_html" ]
    copy => { "blog_html" => "blog_text" }
    copy => { "published" => "@timestamp" }
  }
  mutate {
    gsub => [
      "blog_text", "<.*?>", "",
      "blog_text", "[\n\t]", " "
    ]
    remove_field => [ "published", "author" ]
  }
}
```

**Output**

```
output {
  stdout {
    codec => dots
  }
  elasticsearch {
    hosts => [ "https://<your-elsaticsearch-url>" ]
    index => "elastic_blog"
    user => "elastic"
    password => "<your-elasticsearch-
password>"
  }
}
```

# Installing ES on Linux Server

- Refer to LAB Session, Week 15.

IT601 – System and Network Administration

# IT Operations & Support Process

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

# Contents

- Introduction to Help Desk
- Development and Operations (DevOps)

# Introduction to Help Desk

- **Helpdesk is the primary mechanism to provide customer support**
  - A helpdesk is a place, real or virtual, where people can get answers to their computing questions, report problems, and request new services.
    - It may be a physical desk that people walk to, or it may be a virtual helpdesk that people access electronically.

- **Significance of Helpdesk**
  - Nothing is more important than it for ITOps.
  - It is the face of an organization. The HD staff is the first impression on customers and maintain relationship, good or bad, with them.
  - The HD fix the issues , part of living with computers and are the heroes. Customers call in an emergency.
  - A good helpdesk reflects well on your organization. The typical customer sees only the customer support portion of your organization and often assumes that this is your entire organization.
  - Customers have no idea which back-office operations and infrastructure duties are also performed. In short, a helpdesk is for helping the customers.
  - Don’t forget the help in helpdesk.

# Introduction to Help Desk

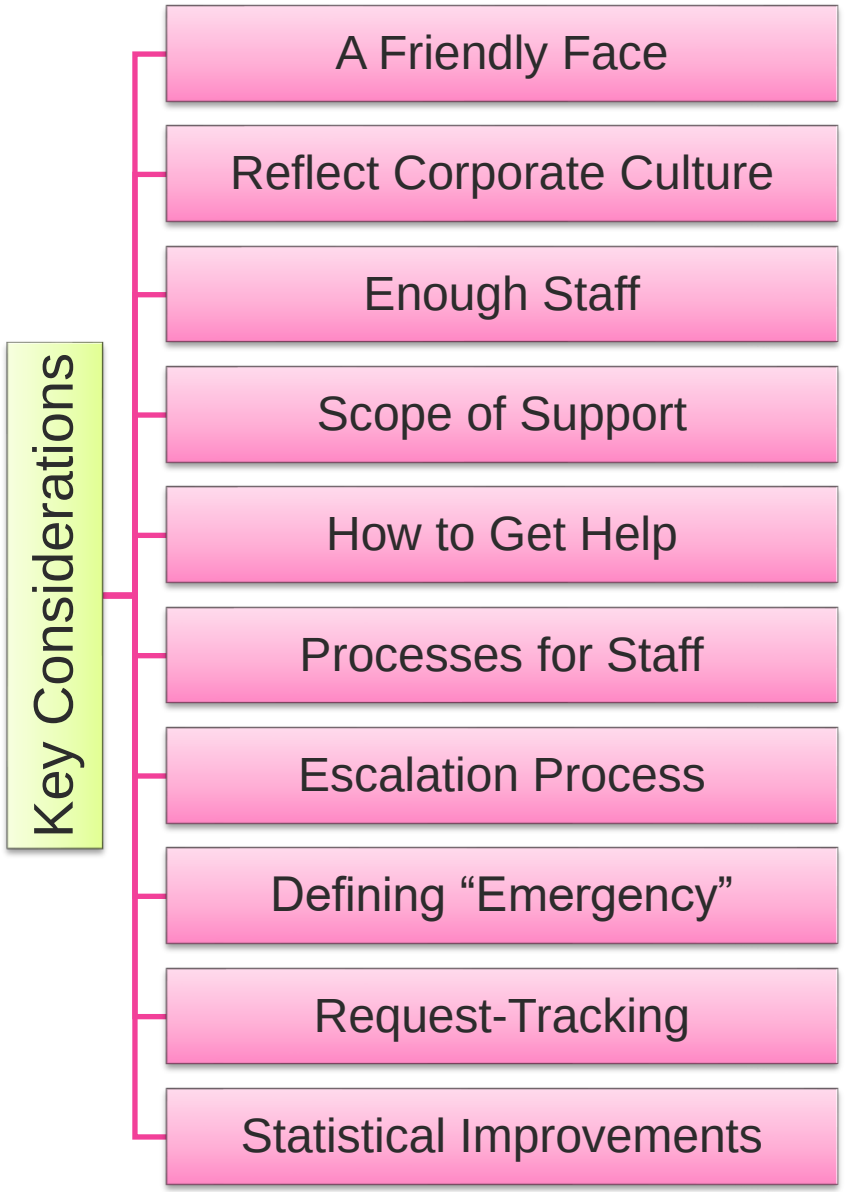
- **Is a Helpdesk really required?**
  - Every organization has a helpdesk.
    - It may be physical  like walk-up counter
    - Virtual like by phone or email.
    - Sometimes unofficial.
  - Small orgnizations may not have formal helpdesk, but still leads to issues.
  - Large orgnizations needs anyhow a formal helpdesk
    - Developing a formal helpdesk should be part of that organizational planning
  - Symptoms of lacking formal helpdesk
    - Communication Problems
    - SAs unable to complete Project Tasks
    - Continuous SAs Interuuptions

*Source : https://klik.solutions/managed-it-services/it-help-desk-services/*

# Introduction to Help Desk

- The transition from ad hoc to formal helpdesk can be uncomfortable to customers.
  - SAs should expect this push-back and do their best to ease the transition.
  - Communicating the new helpdesk procedures clearly is important.
- Helpdesks do not need to be purely physical locations but instead can be virtual.
  - Problems can be reported and replies returned via email.
  - Telephone, text-based, and audio chat systems can also be used.
  - Self-help systems are also popular but should not be considered a replacement for systems that involve human interaction.
    - These systems can reduce the workload of helpdesk attendants but cannot provide interactive debugging or resolve workflow issues that require real-time interaction.
    - There should be a phone number to call to report that the self-help system is down.
- A simple repository of documentation for customers on such topics as how to get help or request service activation and solutions to common problems.

# Key Consierations for Helpdesk



# Friendly Face

- **A helpdesk should have a friendly face.**
  - For a physical helpdesk, the interior design should be pleasant and welcoming.
    - A web-based virtual helpdesk is equally welcoming, Use a design based on soothing colors and readable fonts with the most selected items at the top left of the first page.
- **The faces of the staff should be welcoming and friendly, as should their personalities.**
  - When hiring HD staff, A key factor is that some people have personalities that are suited for customer service; others don't.
- **The roll or supervisor is key factor.**
  - The tone set by the staff will reflect that set by the supervisor.
    - A supervisor who yells at the staff will find staff yelling at customers.
  - A good-natured supervisor who can laugh and is always friendly will attract similar staff, who will reflect such an attitude with customers.
  - It is easier to build a reputation for being friendly initially than to restore a bad reputation.

  - The supervisor should be the friendly person you want your staff to be. Be a role model.

# Reflect Corporate Culture

- **The look and feel of your helpdesk of an organization reflects its corporate culture.**
  - A helpdesk doesn't garner respect in a company when people working at the helpdesk buck the corporate culture.
    - A company that is very strict and formal may reflect this with strict dress codes and ways of conducting business, but the people at the helpdesk wear logo T-shirts and jeans, and a visitor hears a video game being played in the background.
    - A little asking around will find that the helpdesk has a reputation of being a bunch of slackers, no matter how hard they work or how high the quality of the service they provide.
  - The opposite can also happen.

- **Spend time to consider the culture and "look" of your helpdesk as compared to that of the customers they serve. Try to evolve to a culture that suits the customers served.**

# HelpDesk Staff

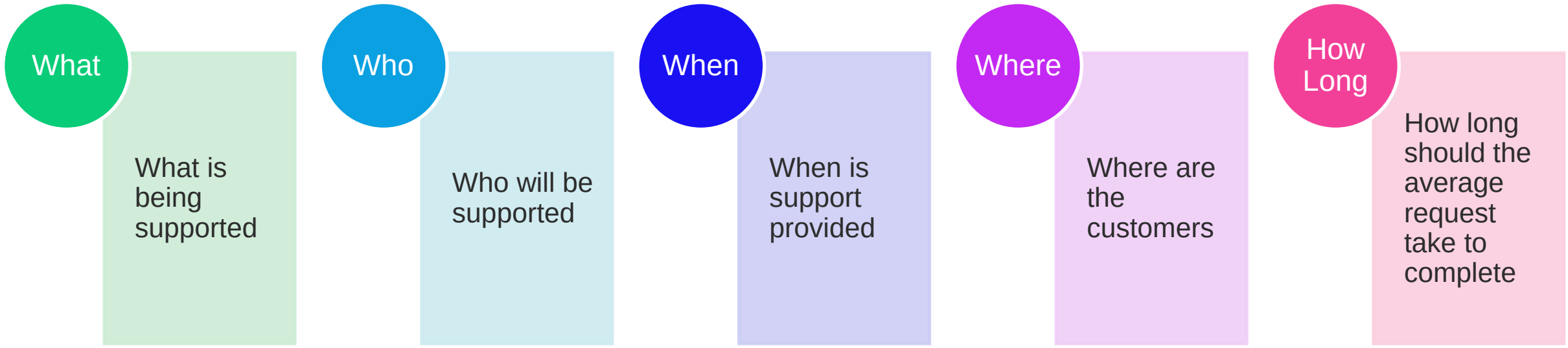
- **A helpdesk can be helpful only if it has enough people to serve customers in a timely manner.**
  - Otherwise, people will look elsewhere for their support.
- **Metrics for Sizing Helpdesk Staff**

| Customer to HD Staff (CHS) Ratio | Call Volume Ratio (CVR) |
|---|---|
| ▪ Universities often have 1000s of students per HD Staff. Corporates may have a higher ratio or a lower ratio. | ▪ It is better to focus on callvolume ratios and time-to-call completion. |
| ▪ An indirect metric | ▪ A Direct Metric |
| ▪ In a commercial computer science research dept., the ratio is often 40:1, with same skill level of the first-tier SAs and second-tier SSs. | ▪ The rate at which customers receive busy signals or wait to receive a response, minutes required to resolve issues excluding time spent in “customer wait,” |
| ▪ E-commerce sites usually have a separate and depending on the services being offered, the ratio can be 10,000:1 or 1,000,000:1. | ▪ Managing resources based on call volume also presents a more diverse set ofpotential solutions. |
| ▪ Ratios are a no-win situation. Management will always push to have a higher ratio; customers for lower ratio. | ▪ It is required to have appropriate metrics to make decisions about improving processes. |
| ▪ The the ratio can be increased by providing less service to the customers, which usually costs the organization. | ▪ Metrics can reveal good candidates for new automation, documentation, ortraining for both SAs and customers. |
| | ▪ Metrics can reveal which processes are more effective, which are used heavily, or which are not used at all. |

# Scope of Support

- A helpdesk should have a policy defining the scope of support. This document explains what an SA group is and isn't responsible for.

- The components of scope are what, who, where, when, and how long




- SAs should have a writen the scope-of-support policy.
  - What is in scope and what is out of scope

# Specify How to Get Help

- **The companion to the scope-of-support document is a document that specifies how to get help: by phone, email, a ticket system, and so on.**
  - Certain types of requests may be directed to certain departments, or a unified helpdesk might be the single point of contact that forwards requests as appropriate to individual departments.

- **An image or document specifying how to get help should appear on default Windows background wallpaper images:**
  - “CompanyName IT helpdesk: [phone number] [email address] [web site].”

- **If customers have not been given clear directions on the proper way to get help, they will contact SAs directly, interrupting them at inappropriate times, and making it impossible to get larger projects done.**

# Define Processes for Staff

- **Helpdesk staff should have well-defined processes to follow.**
  - In a smaller environment, this is not as important, because the processes are more ad hoc or are undocumented because they are being used by the people who built them.
  - However, for a large organization, the processes must be well documented.
- **Very large helpdesks should use scripts as part of their training. Every service supported has an associated flow of dialogue to follow to support that service.**
- **Some Scripts required identify verifications**
  - The script for a request to reset a password would, for security reasons, require callers to prove who they are, possibly by knowing a unique piece of personal information, before a new password would be set.

# Establish an Escalation Process

- **Escalation is a process by which an issue is moved from the current staff person to someone with more expertise.**
  - The first line of operators should be able to handle 80 percent to 90 percent of all calls and escalate the remaining calls to a second tier of support.
  - The people at this second tier may have more experience, more training, and, possibly, other responsibilities.
  - Larger organizations can have four or more tiers; the higher tiers may include the people who built or currently maintain the service in question.
- **The escalation process is also what customers use when they are dissatisfied with the support they are receiving.**
  - Large numbers of calls being escalated to the second tier is a warning sign of a larger, systemic problem.
  - Usually, it indicates that the first-tier staff people need more training or do not have the tools to do their job properly.
  - If large numbers of calls are escalated to management, there may be systemic problems with the support the helpdesk is providing.

# Define “Emergency” in Writing

- **Often, SAs are overloaded because every customer claims to have an emergency that requires immediate attention.**

  SAs may feel that customers are using this claim to boss them around, which decreases morale and increases stress levels.

- **Having a written policy empowers SAs to know when to push back and gives them a document to point to when they need it.**

  - If the customer still disagrees with this assessment, the SA can pass the issue up to someone in management, who can make the decision.
  - This lets the SA focus on technical duties and lets management focus on setting priorities and providing resources.

- **Every company should be able to define what constitutes an emergency.**
  - At a factory, an emergency is anything that stops the assembly line.
  - At a web-based service or ISP, an emergency might be anything that will prevent the service from meeting an SLA.

# Use Request-Tracking Software

- **Every helpdesk needs some kind of software to help it manage requests.**
  - The alternative is a collection of notes written on scraps of paper. Although it is simple in the beginning and sufficient for environments with one or two SAs, a system based on notes on paper doesn't scale.
  - Requests get lost, and management has no ability to oversee the process to better allocate resources.
- **Those are the first qualities that you need in helpdesk software. As a helpdesk grows, software can help in other areas.**
- **Features of Helpdesk Software**
  - Helpdesk software should permit some kind of priority to be assigned to tickets.
  - Another important aspect of helpdesk software is that it collects logs about which kinds of requests are made and by whom.
  - Helpdesk software should also automate the collection of data on customer satisfaction.
  - It is critical that helpdesk software match the workflow of the people who use it.
  - Choosing helpdesk software is not an easy process. Most software will need a lot of customizing for your environment.

# Statistical Improvements

- **Many sophisticated statistics can be gathered about a helpdesk.**
  - For example, you can monitor the rate of escalations to determine where more training is needed.
- **when dealing with upper management for budgeting and planning purposes, historical statistics become much more valuable.**
  - You can make a better case for your budget if you can show multiyear trends of customer growth, call volume, types of calls, technologies used, services provided, and customer satisfaction.
  - When you are asked to support a new technology or service, you can use past data to predict what the support costs may be.
- **The value of statistics increases as the organization grows, because the management becomes less directly involved in the work being done.**
  - As an organization grows, statistics are easier to collect, and it becomes more important that they be collected.

# After-Hours and 24/7 Coverage

- **As computers become critical to an ever-expanding list of business processes, customers are asking for 24/7 coverage more often.**
  - Although a full three-shift staff may be required in some organizations, some very simple ways to provide 24/7 coverage are not as expensive.
- **Options**
  - Set up a voicemail box that alerts a pager when new messages arrive. The pager can be rotated among various staff members.
  - Have all managers of the customer groups know the home phone number of the helpdesk's supervisor, who then takes responsibility for calling SAs in turn until one is found.
- **No matter how SAs are contacted after hours, the person must be compensated.**
  - Some organizations have a salary incentive for oncall time, equivalent to a fraction of the employee's salary and time and a half if the person is called.
  - Other organizations issue compensation time either officially or unofficially.

# Better Advertising for the Helpdesk

- **Defining your policies and providing announcements online is nice, but rarely will anyone seek them out.**
- **Options are**
  - Publish on Website
  - Email to customers esp. new policies
  - Workshops

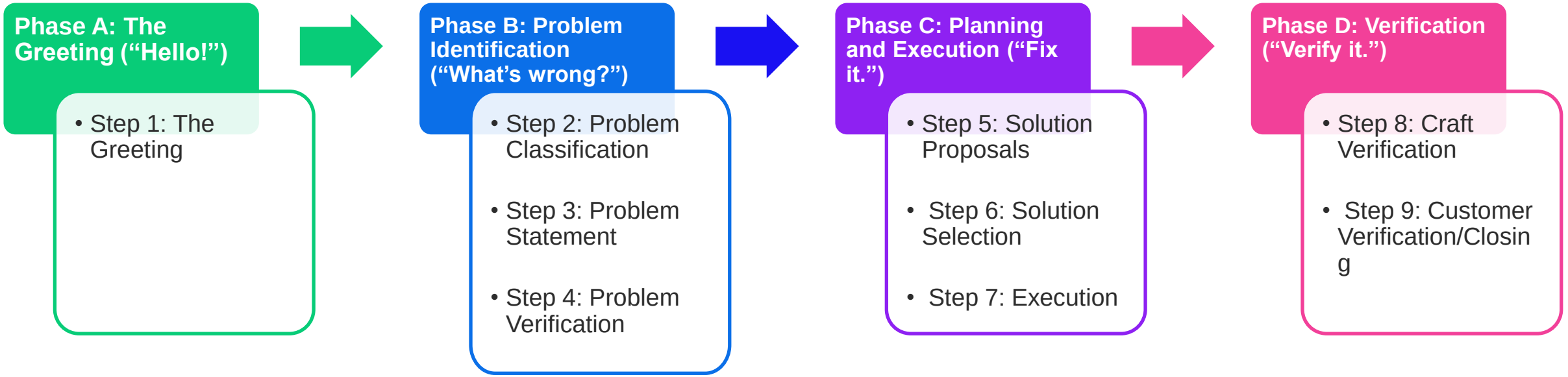
# Multiple Helpdesks

- **When an organization grows, it may make sense to have two separate helpdesks:**
  - One for requesting new services.
  - Second for reporting problems that arise after the service has been successfully enabled.
  - A third group deals with installing the new service, especially if it requires physical work.

- **This third group may be an internal helpdesk that installers all over the organization can call to escalate installation problems. It is not uncommon, though, for this third group to be the second tier of one of the other helpdesks.**

# Processing  for processing customer requests

- **The method for processing these customer requests has nine steps, which can be grouped into four phases:**

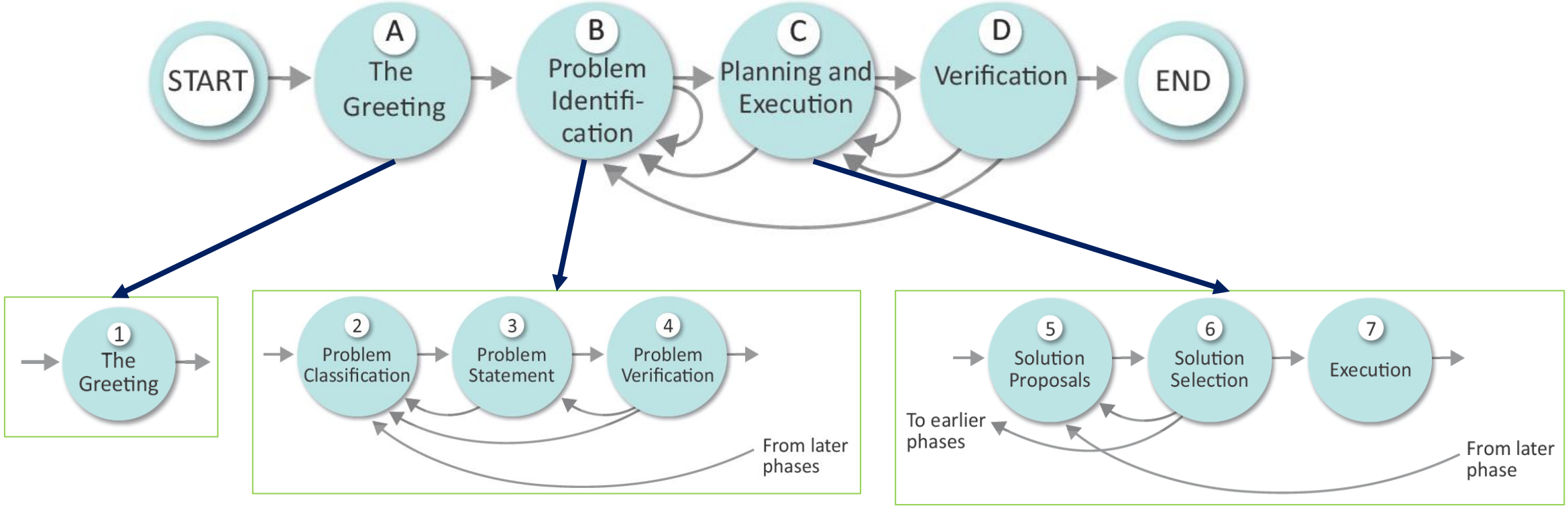



- **This method gives structure to what is, for newer SAs, a more haphazard process.**
  - It helps SAs solve problems more efficiently by keeping them focused and helps them avoid mistakes. It introduces a common set of terminology that, when used by the entire SA team, increases the ability to communicate within the group.

# Problem Solving Process

- **Problem Solving Process consist of four phases**

A. Reporting the problem
B. Identifying the problem
C. Planning and executing a solution
D. Verifying that the problem resolution is complete



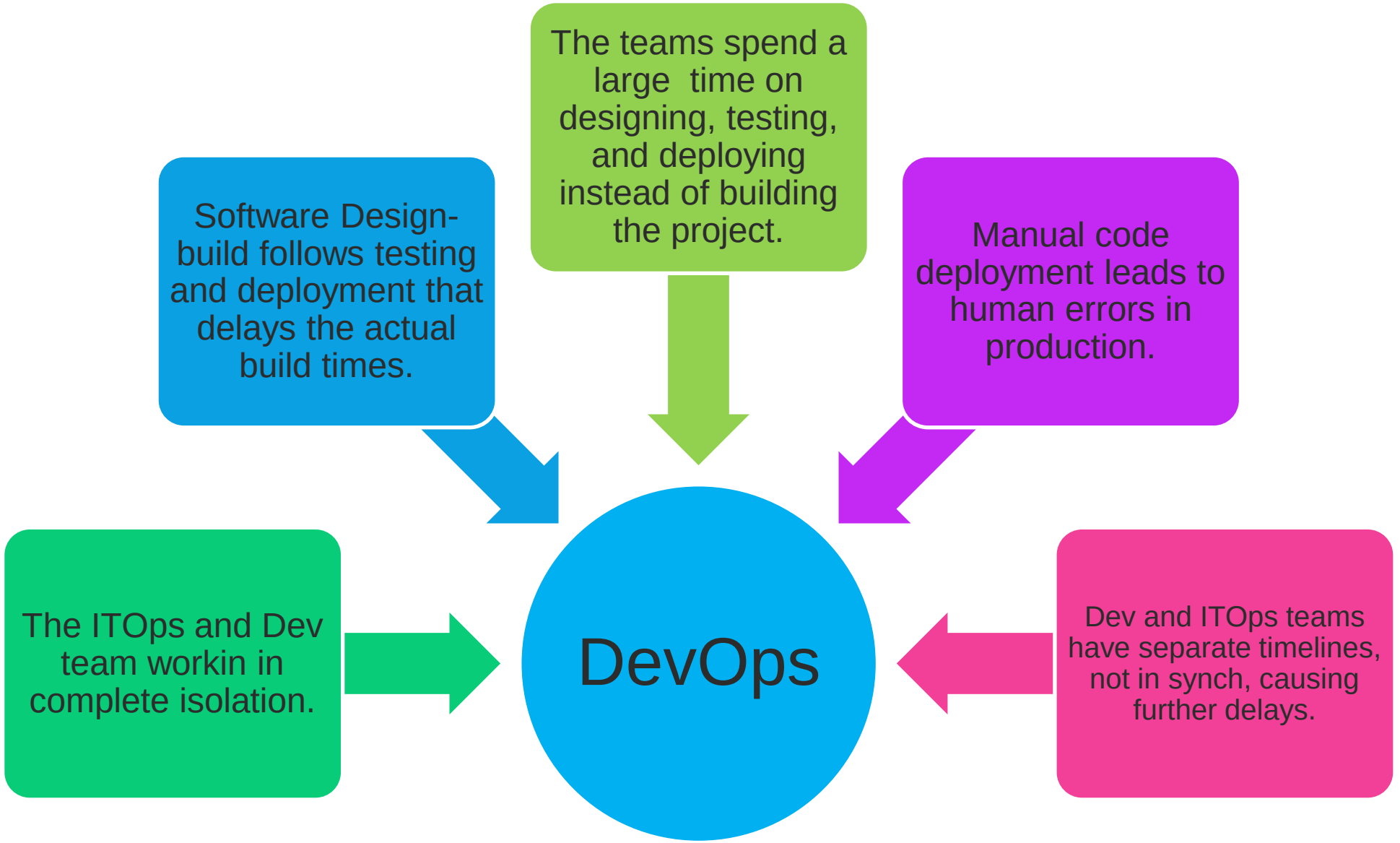
# Development and Operations Together

- **Large organizations  often have software development teams and IT operations team.**
  - System and Network Administration  is part of IT operations
- **Some problems reported to system administrators or tasks requires the collaboration with software development team**
  - Thus development and Operations are not standalone they are strongly coupled
- **The collaboration of software development and IT operations  is also applicable even if organizations donot have in house software development/ IT operations teams (Outsouricing Model)**
  - Thus development and Operations are not standalone they are strongly coupled
- **DevOps is a model which allows agile collaboration between administratively independent software development and IT operations teams.**

# DevOps

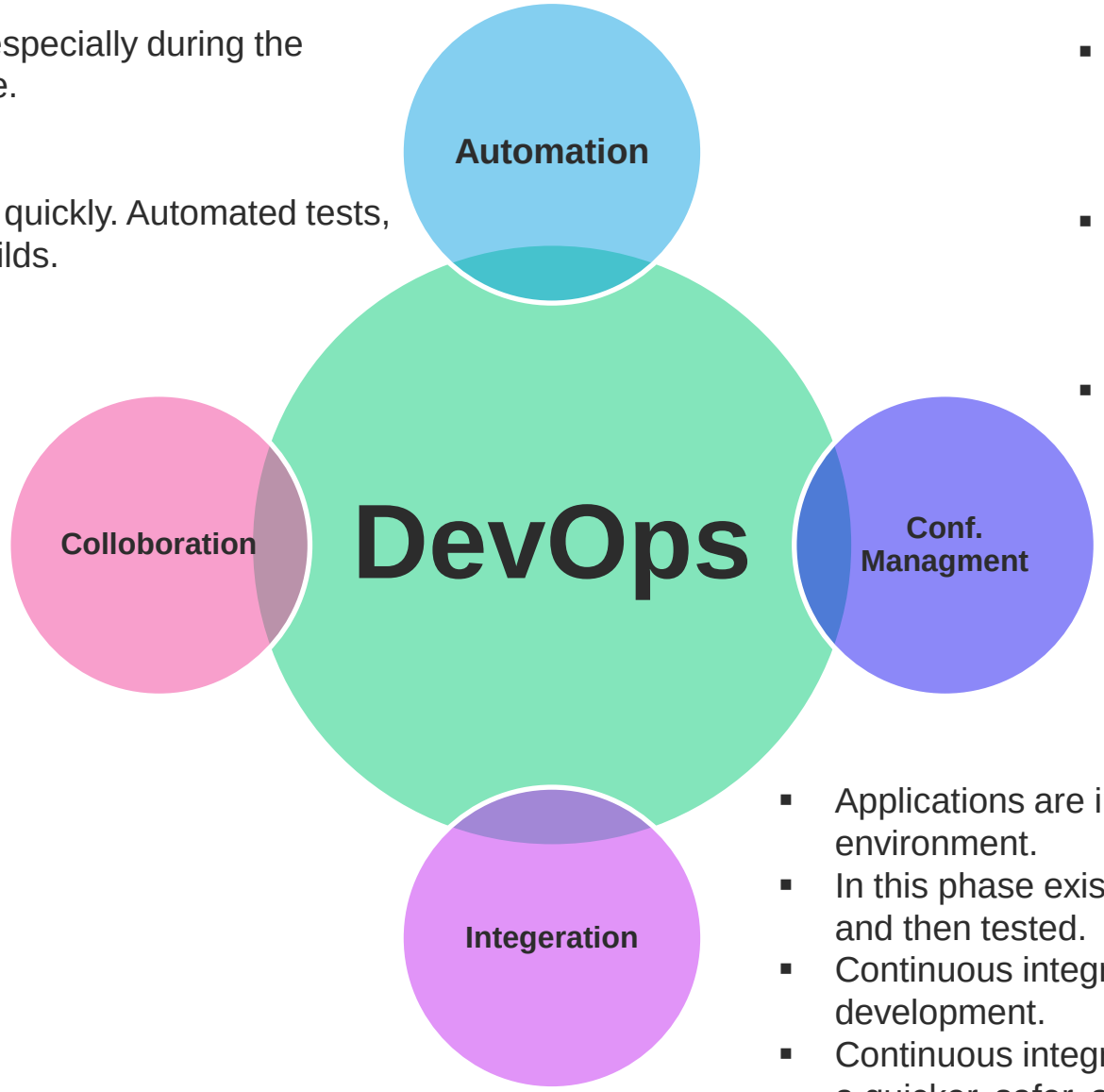
- **The DevOps is a mixture of two words, one is software Development, and second is Operations.**
  - It  allows to jointly handle the entire application lifecycle, from development to testing, deployment, and operations.
  - DevOps helps to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.

# The Need of DevOps



# Features of DevOps

- Reduces time consumption, especially during the testing and deployment phase.
- Increases Productivity
- SW Releases are quicker.
- Catching bugs and their fixed quickly. Automated tests, cloud-based services, and builds.




- It ensures the Apps to interact with only those resources that are concerned with the environment in which it runs.
- The conf files are not created where the external configuration to the application is separated from the source code.
- The conf file can be written during deployment, or they can be loaded at the run time, depending on the environment in which it is running.

- The Dev and ITOps team collaborates that improves the cultural model.
- Improves productivity, accountability and ownership.
- Share responsibilities and work closely in sync, making the deployment to production faster.

- Applications are integrated with other components in the environment.
- In this phase existing code is combined with new functionality and then tested.
- Continuous integration and testing enable continuous development.
- Continuous integration and delivery are implemented to deliver in a quicker, safer, and reliable manner.

# Mertis and Demerits of DevOps

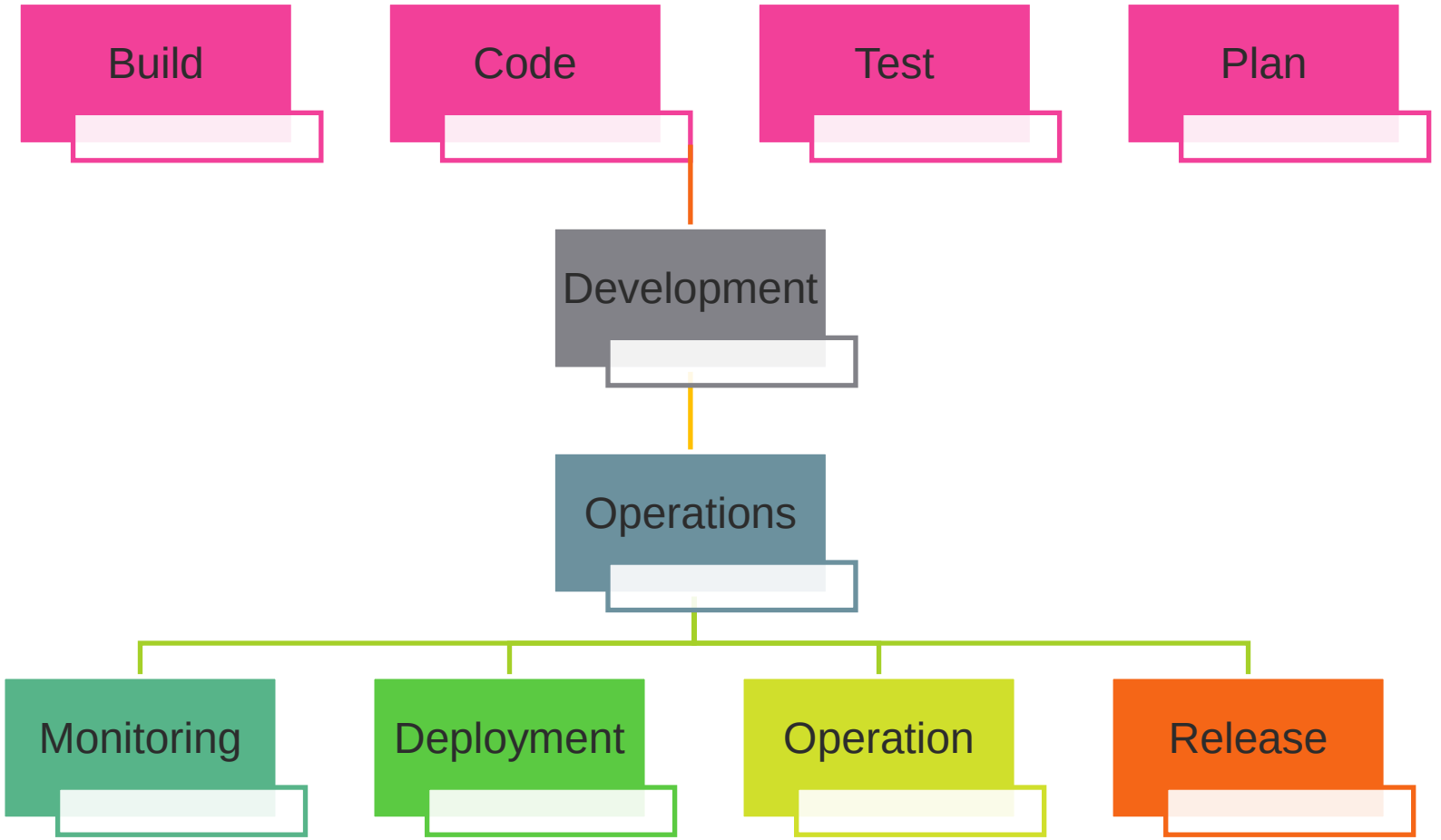
## Merits

- DevOps is an excellent approach for quick development and deployment of applications.
- It responds faster to the market changes to improve business growth.
- DevOps escalate business profit by decreasing software delivery time and transportation costs.
- DevOps clears the descriptive process, which gives clarity on product development and delivery.
- It improves customer experience and satisfaction.
- 
- DevOps simplifies collaboration and places all tools in the cloud for customers to access.
- DevOps means collective responsibility, which leads to better team engagement and productivity.

## Demerits

- DevOps professional or expert's developers are less available.
- Developing with DevOps is so expensive.
- Adopting new DevOps technology into the industries is hard to manage in short time.
- Lack of DevOps knowledge can be a problem in the continuous integration of automation projects.

# DevOps Architecture



# DevOPs Components

## 1 - Build

- Without DevOps, the cost of the consumption of the resources was evaluated based on the pre-defined individual usage with fixed hardware allocation.

- With DevOps, the usage of cloud, sharing of resources comes into the picture, and the build is dependent upon the user's need, which is a mechanism to control the usage of resources or capacity.

## 2 - Coding

- Many good practices such as Git enables the code to be used, which ensures writing the code for business, helps to track changes, getting notified about the reason behind the difference in the actual and the expected output, and if necessary reverting to the original code developed.

- The code can be appropriately arranged in files, folders, etc. And they can be reused.

## 3 - Testing

The application will be ready for production after testing.

In the case of manual testing, it consumes more time in testing and moving the code to the output.

The testing can be automated, which decreases the time for testing so that the time to deploy the code to production can be reduced as automating the running of the scripts will remove many manual steps.

## 4 - Planing

- DevOps use Agile methodology to plan the development.

- With the operations and development team in sync, it helps in organizing the work to plan accordingly to increase productivity.

# DevOPs Components

## 5 - Monitoring

- Continuous monitoring is used to identify any risk of failure. Also, it helps in tracking the system accurately so that the health of the application can be checked.

- The monitoring becomes more comfortable with services where the log data may get monitored through many third-party tools such as Splunk.

## 6 - Deployment

- Many systems can support the scheduler for automated deployment.

- The cloud management platform enables users to capture accurate insights and view the optimization scenario, analytics on trends by the deployment of dashboards.

## 7 - Operation

- DevOps changes the way traditional approach of developing and testing separately.

- The teams operate in a collaborative way where both the teams actively participate throughout the service lifecycle.

- The operation team interacts with developers, and they come up with a monitoring plan which serves the IT and business requirements.

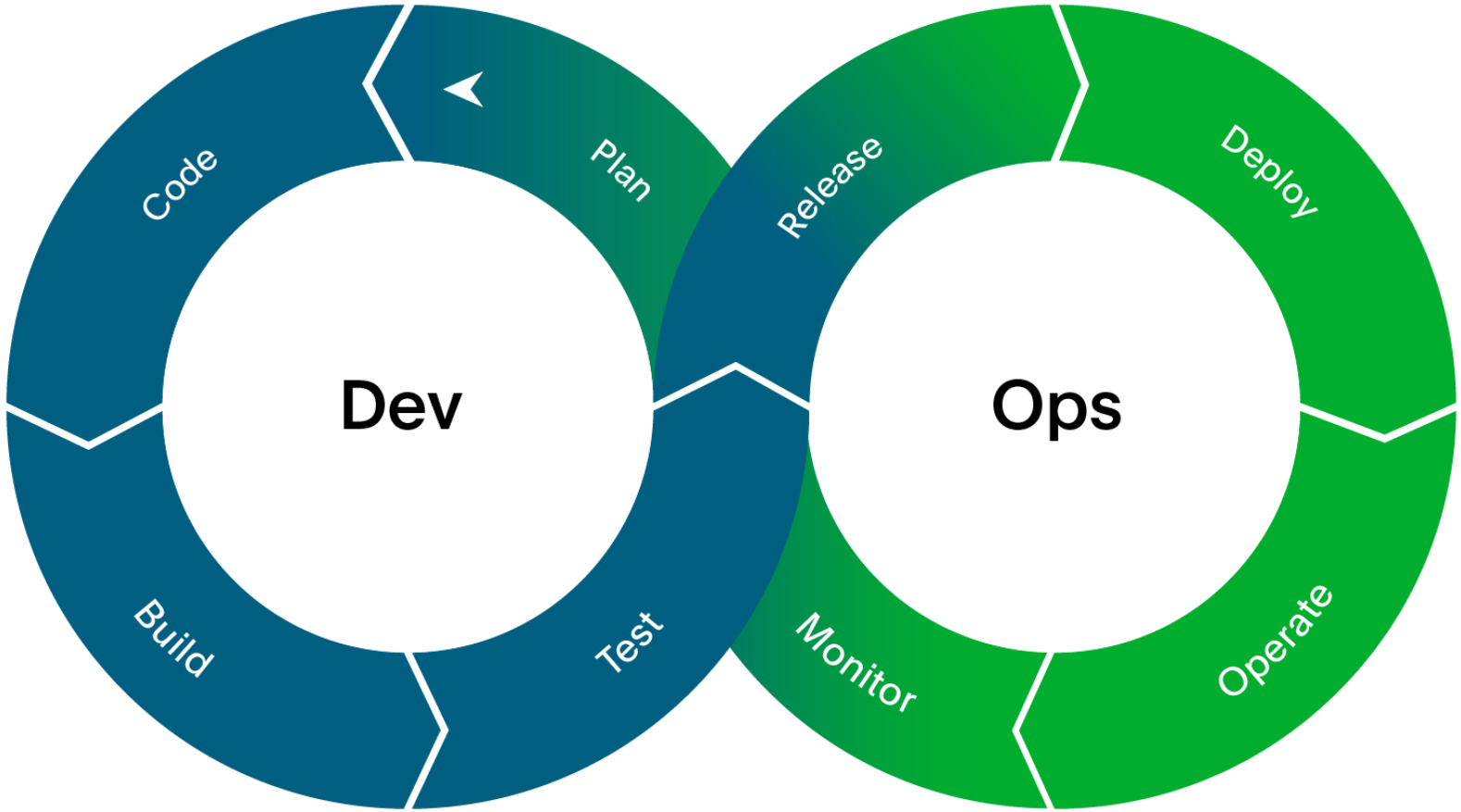
## 8 - Release

- Deployment to an environment can be done by automation.

- But when the deployment is made to the production environment, it is done by manual triggering. Many processes involved in release management commonly used to do the deployment in the production environment manually to lessen the impact on the customers.

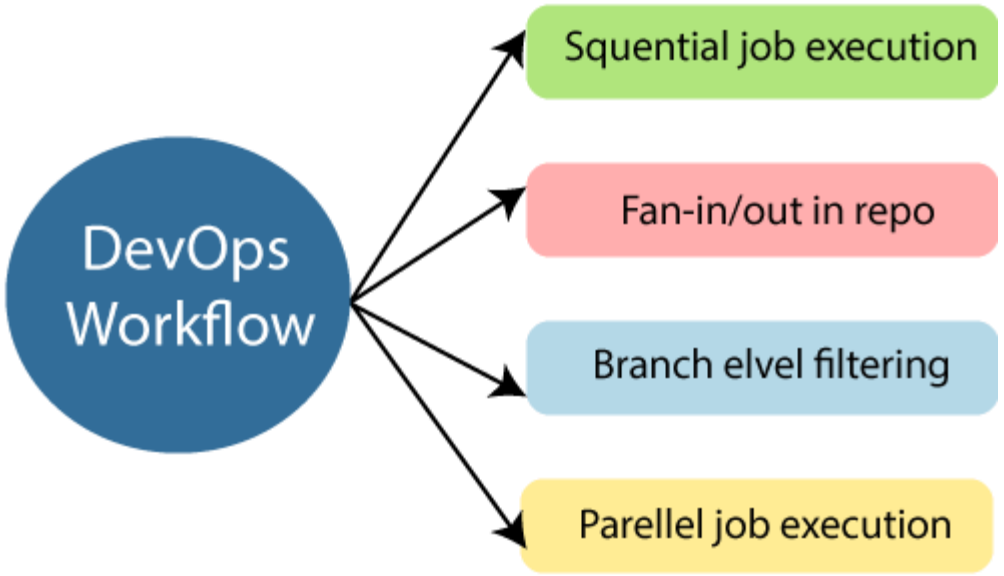
# DevOps Cycle

- All components discussed reviously work in continuous model



# Workflow

- DevOps workflow provides a visual overview of the sequence in which input is provided. Also, it tells about which one action is performed, and output is generated for an operations process.




- DevOps workflow allows the ability to separate and arrange the jobs which are top requested by the users. Also, it gives the ability to mirror their ideal process in the configuration jobs.

# DevOps Principles

- The main principles of DevOps are Continuous delivery, automation, and fast reaction to the feedback.

### End to End Responsibility

- DevOps team need to provide performance support until they become the end of life. It enhances the responsibility and the quality of the products engineered.

### Continuous Improvement:

- DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the growth of products or services offered.

### Automate Everything

- Automation is an essential principle of the DevOps process. This is for software development and also for the entire infrastructure landscape.

### Custom Centric Action

- DevOps team must take customer-centric for that they should continuously invest in products and services.

### Monitor and test everything

- The DevOps team needs to have robust monitoring and testing procedures.

### Work as one team

- In the DevOps culture role of the designers, developers, and testers are already defined. All they needed to do is work as one team with complete collaboration.

- These principles are achieved through several DevOps practices, which include frequent deployments, QA automation, continuous delivery, validating ideas as early as possible, and in-team collaboration.

# DevOps Practices

- Some identified DevOps practices are:

  - Self-service configuration
  - Continuous build
  - Continuous integration
  - Continuous delivery
  - Incremental testing
  - Automated provisioning
  - Automated release management

# DevOps Tools

## Puppet

- Puppet is the most widely used DevOps tool.
- It allows the delivery and release of the technology changes quickly and frequently.
- It has features of versioning, automated testing, and continuous delivery.
- It enables to manage entire infrastructure as code without expanding the size of the team.

## Ansible

- Ansible is a leading DevOps tool.
- Ansible is an open-source IT engine that automates application deployment, cloud provisioning, intra service orchestration, and other IT tools.
- It makes it easier for DevOps teams to scale automation and speed up productivity.
- Ansible is easy to deploy because it does not use any agents or custom security infrastructure on the client-side, and by pushing modules to the clients.
- These modules are executed locally on the client-side, and the output is pushed back to the Ansible server.

## Docker

- Docker is a high-end DevOps tool that allows building, ship, and run distributed applications on multiple systems.
- It also helps to assemble the apps quickly from the components, and it is typically suitable for container management.

## Nagios

- Nagios is one of the more useful tools for DevOps.
- It can determine the errors and rectify them with the help of network, infrastructure, server, and log monitoring systems.

## CHEF

- A chef is a useful tool for achieving scale, speed, and consistency.
- The chef is a cloud-based system and open source technology. This technology uses Ruby encoding to develop essential building blocks such as recipes and cookbooks.
- The chef is used in infrastructure automation and helps in reducing manual and repetitive tasks for infrastructure management.
- Chef has got its convention for different building blocks, which are required to manage and automate infrastructure.

# DevOps Tools

## Jenkins

- Jenkins is a DevOps tool for monitoring the execution of repeated tasks.
- Jenkins is a software that allows continuous integration. Jenkins will be installed on a server where the central build will take place.
- It helps to integrate project changes more efficiently by finding the issues quickly.

## Git

- Git is an open-source distributed version control system that is freely available for everyone.
- It is designed to handle minor to major projects with speed and efficiency.
- It is developed to co-ordinate the work among programmers. The version control allows you to track and work together with your team members at the same workspace.
- It is used as a critical distributed version-control for the DevOps tool.

## SALTSTACK

- Stackify is a lightweight DevOps tool.
- It shows real-time error queries, logs, and more directly into the workstation.
- SALTSTACK is an ideal solution for intelligent orchestration for the software-defined data center.

## Splunk

- Splunk is a tool to make machine data usable, accessible, and valuable to everyone.
- It delivers operational intelligence to DevOps teams.
- It helps companies to be more secure, productive, and competitive.

## Selenium

- Selenium is a portable software testing framework for web applications.
- It provides an easy interface for developing automated tests.