

VU LEARNING HUB

vulearninghub.xyz

HIGHLIGHTED HANDOUT

IT601

IT601

IT601 by VU Learning HUB

Prepared for VU students • Free study resources

<https://vulearninghub.xyz>

Week 1

IT601 – System and Network Administration

Prepared by DL...

May You All Stay Success in your Whole Life,
Ameen! Summ Ameen... Just Remember us 
in your Deep Prayers. JazakAllah...

Course Introduction

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

- **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**

➤ **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

➤ 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)

➤ Credit Hours:

- Theory : 3
- Practicals : 1

➤ Assessment

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

- Operating Systems
- Data Communications and Computer Networks

➤ **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**

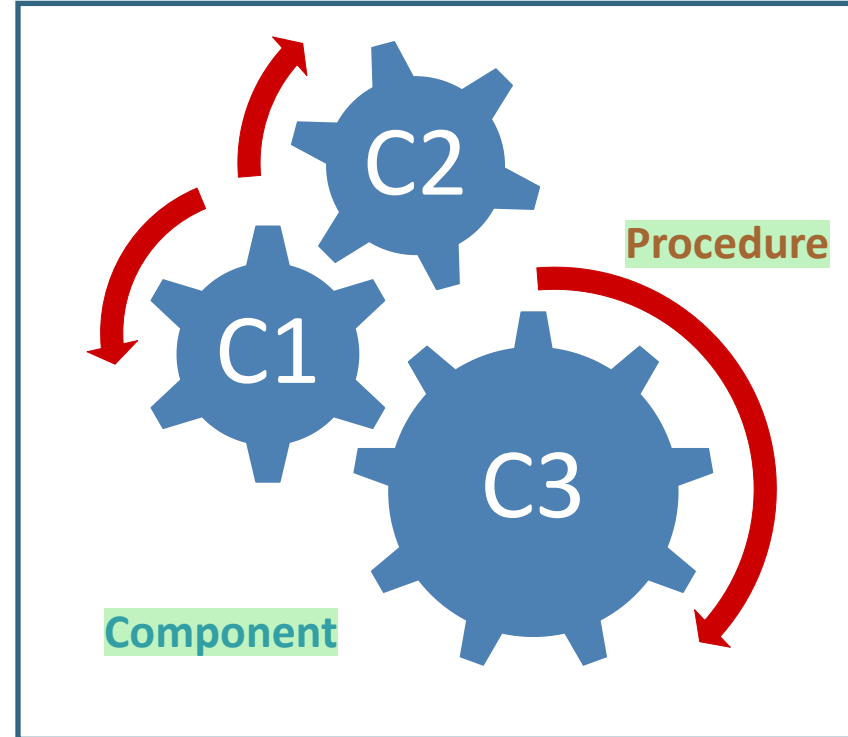
- **IT System**

- **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



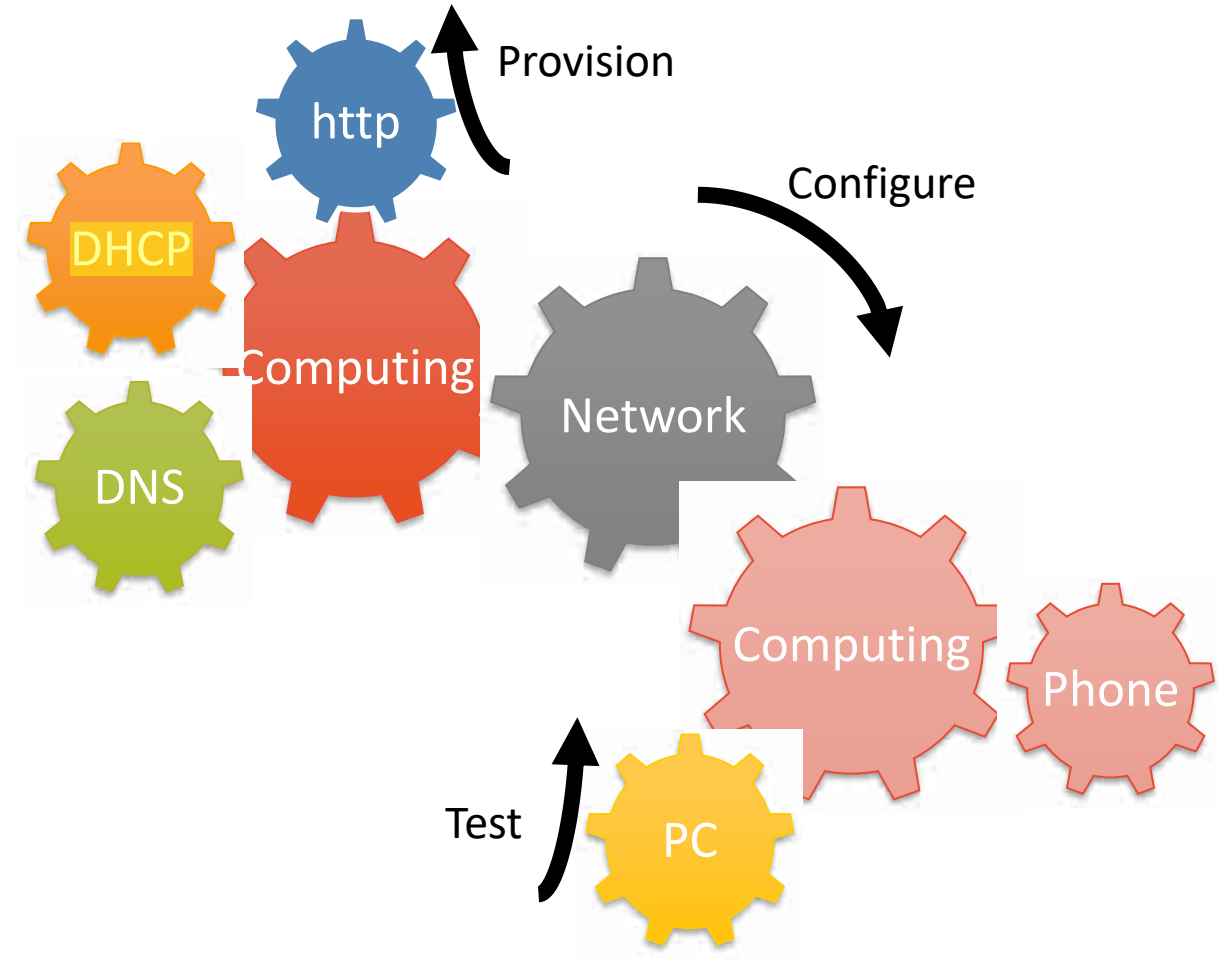
Common Purpose

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

Services

Computing Machines

Software

Processes

Network

Users

IT System Components

Computing
Machines

Software

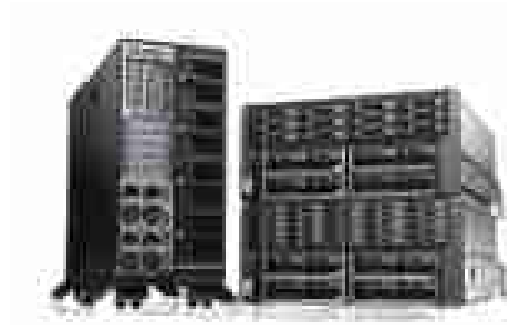
Network

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

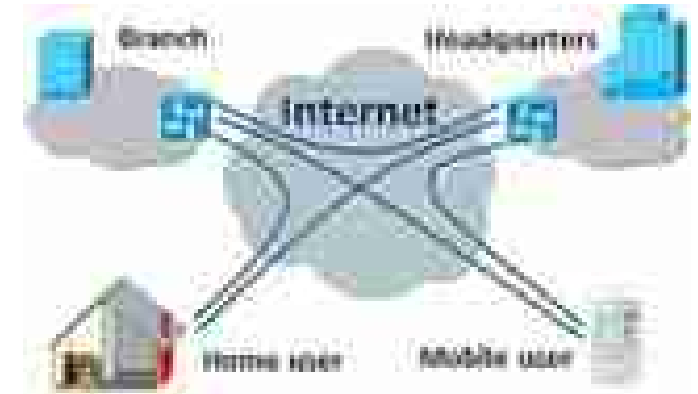
Wireless



Cable and Connectors



Router
Switch
Firewall
DPI
Loadbalancers



VPNs

IT System Components

Computing
Machines

Software

Network

Services

Users

Processes



Application Services



Webservices



Network Services

IT System Components

Computing
Machines

Software

Network

Services

Users

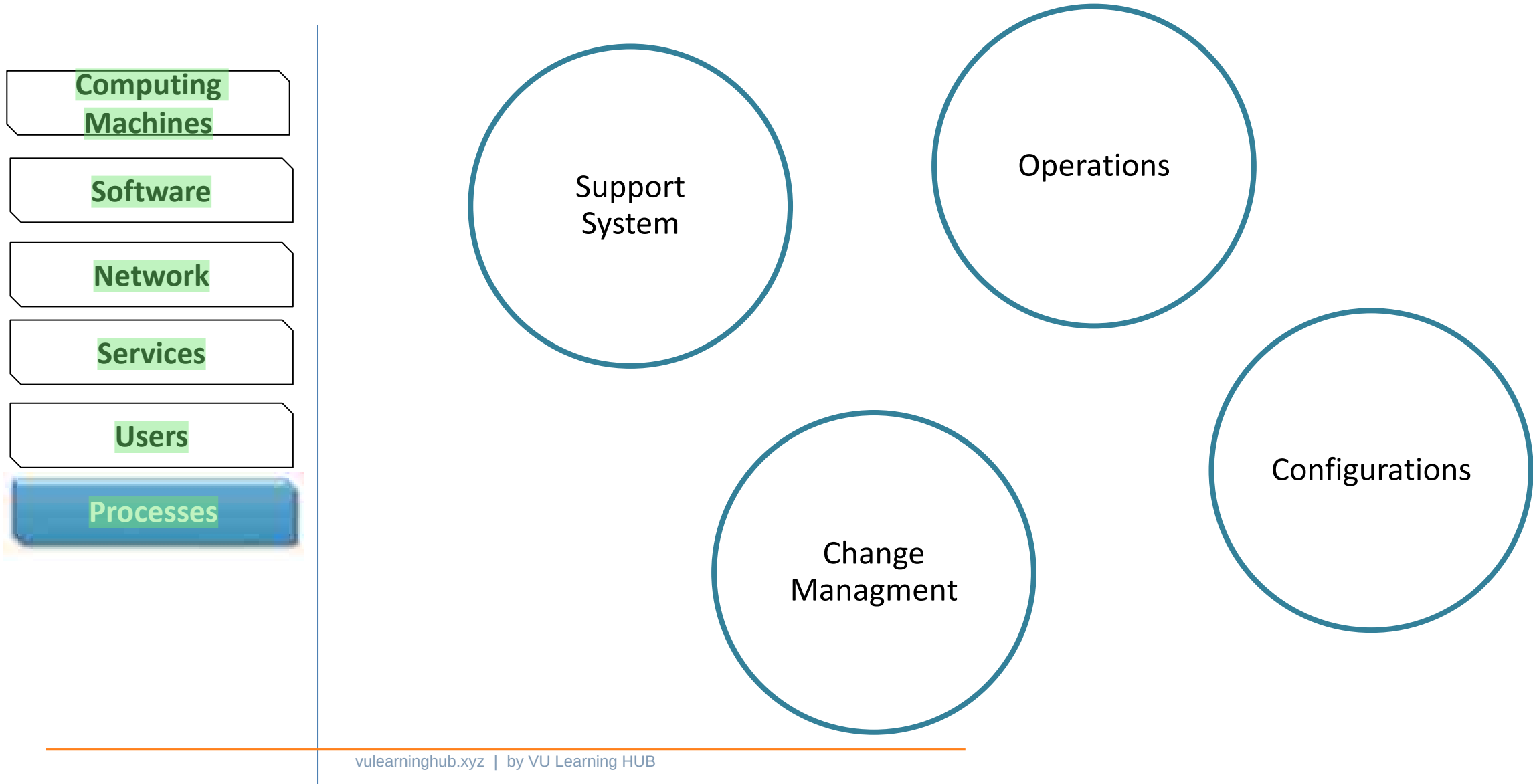
Processes

Diverse
Users

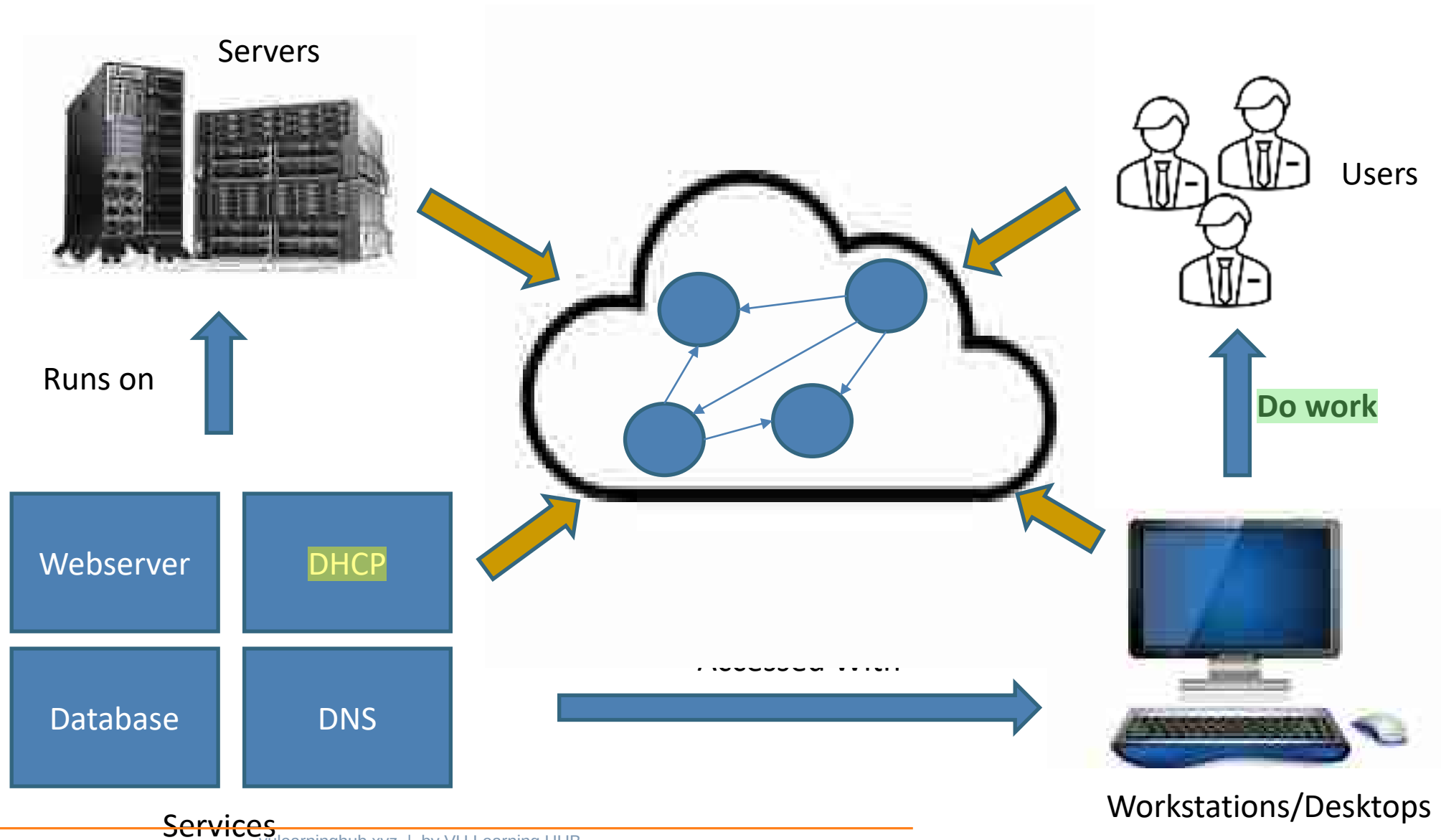
Different
Roles

Different
Levels

IT System Components



IT System



IT601 – System and Network Administration

STATES OF COMPUTER MACHINES

Arif Husen

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

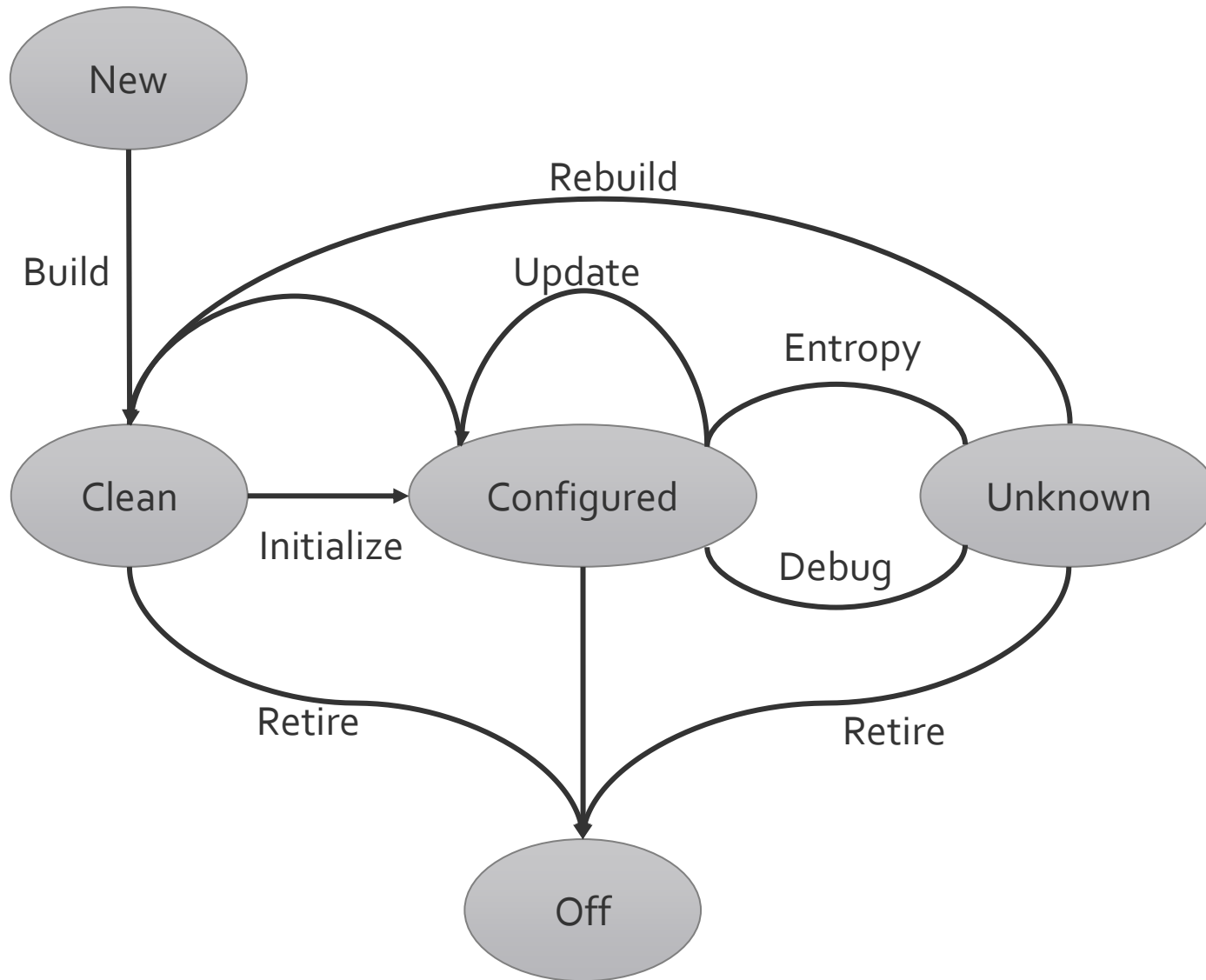
- A Computer Machine Consists of

- 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



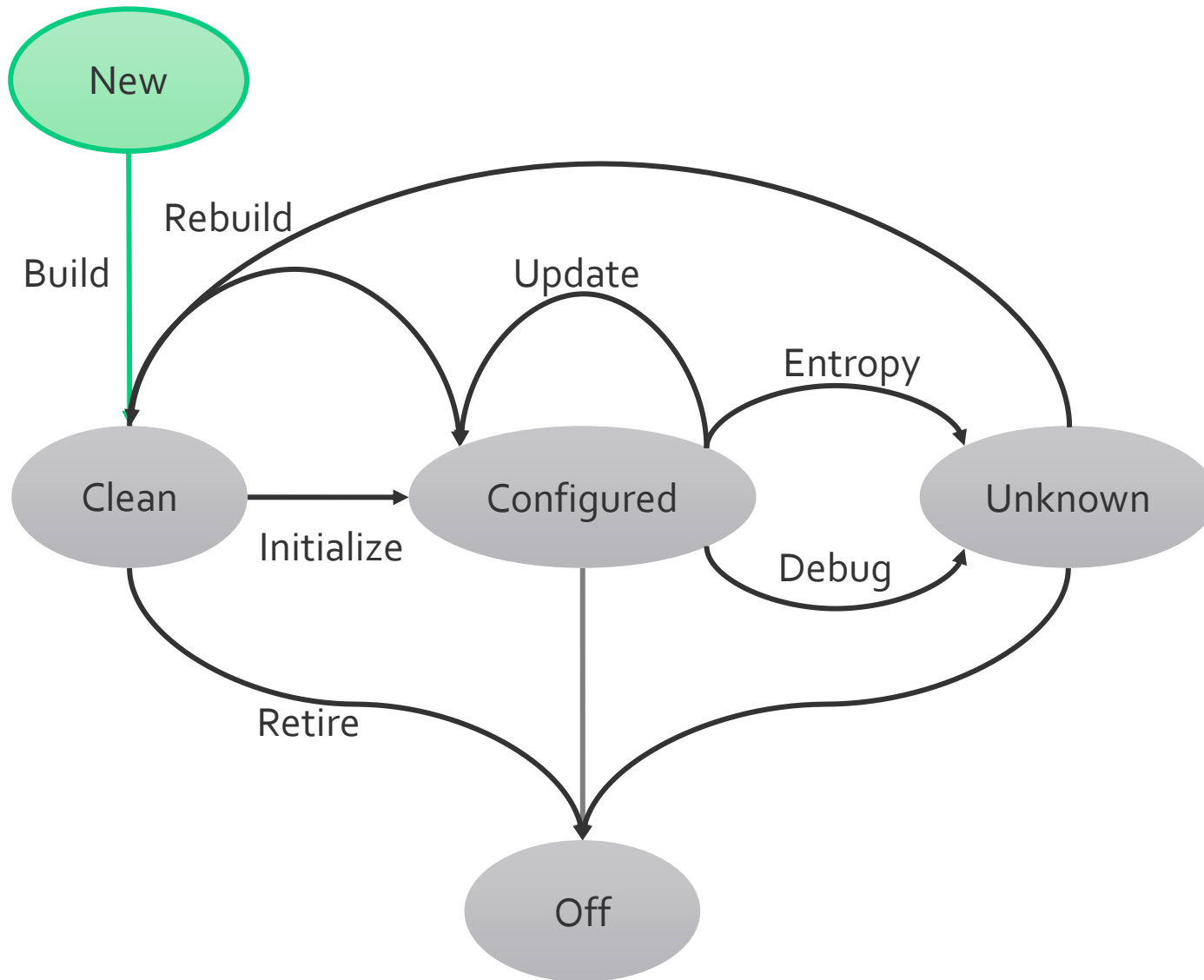
• Five States

- 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

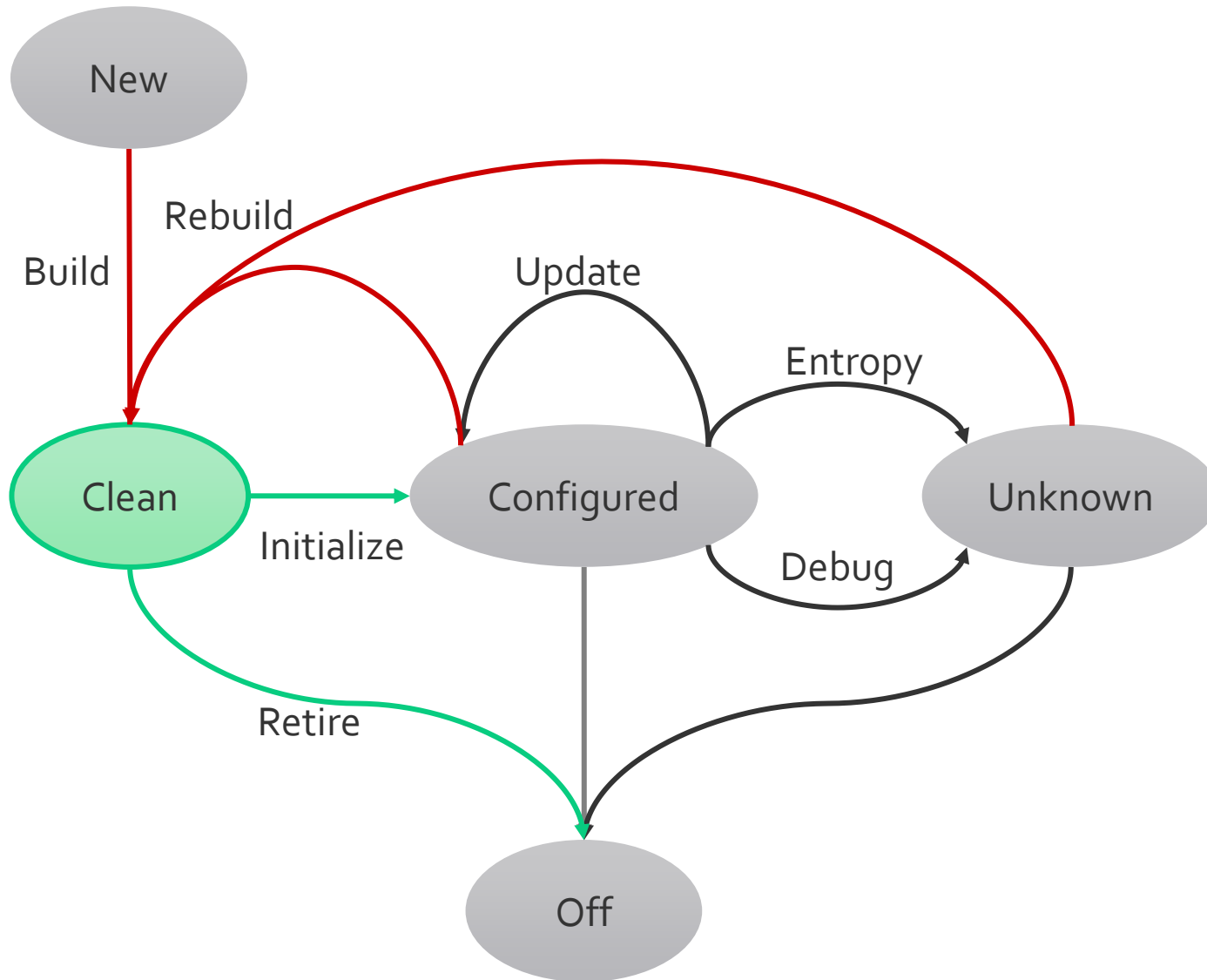
Unpacking

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.

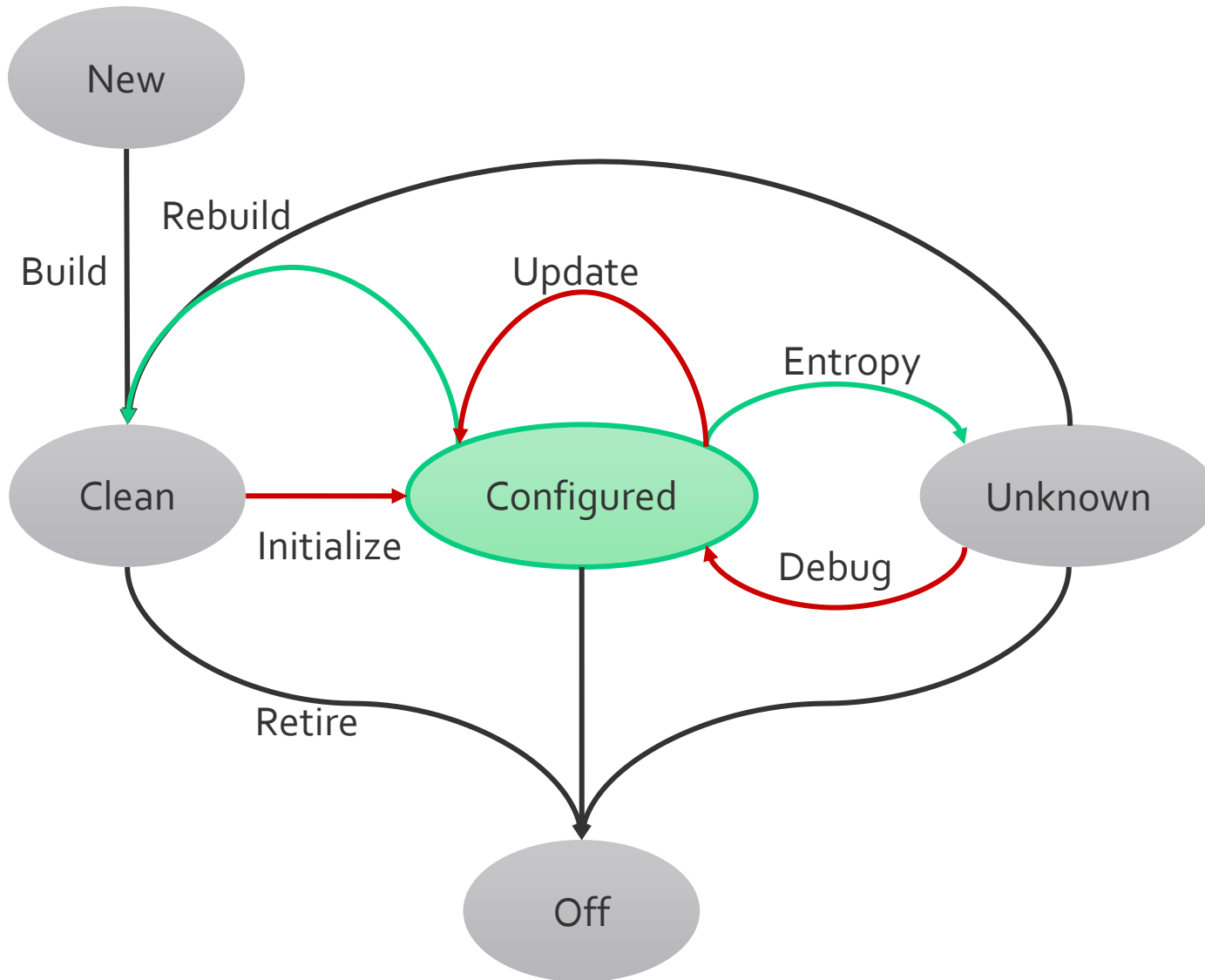
Build Process

- Select OS
- Install OS
- Drivers and Softwares

Rebuild Process

- Recovered Data
- Reformat Disks
- Reinstall OS/Software

Evards 'S Life Cycle Of Computer Machine



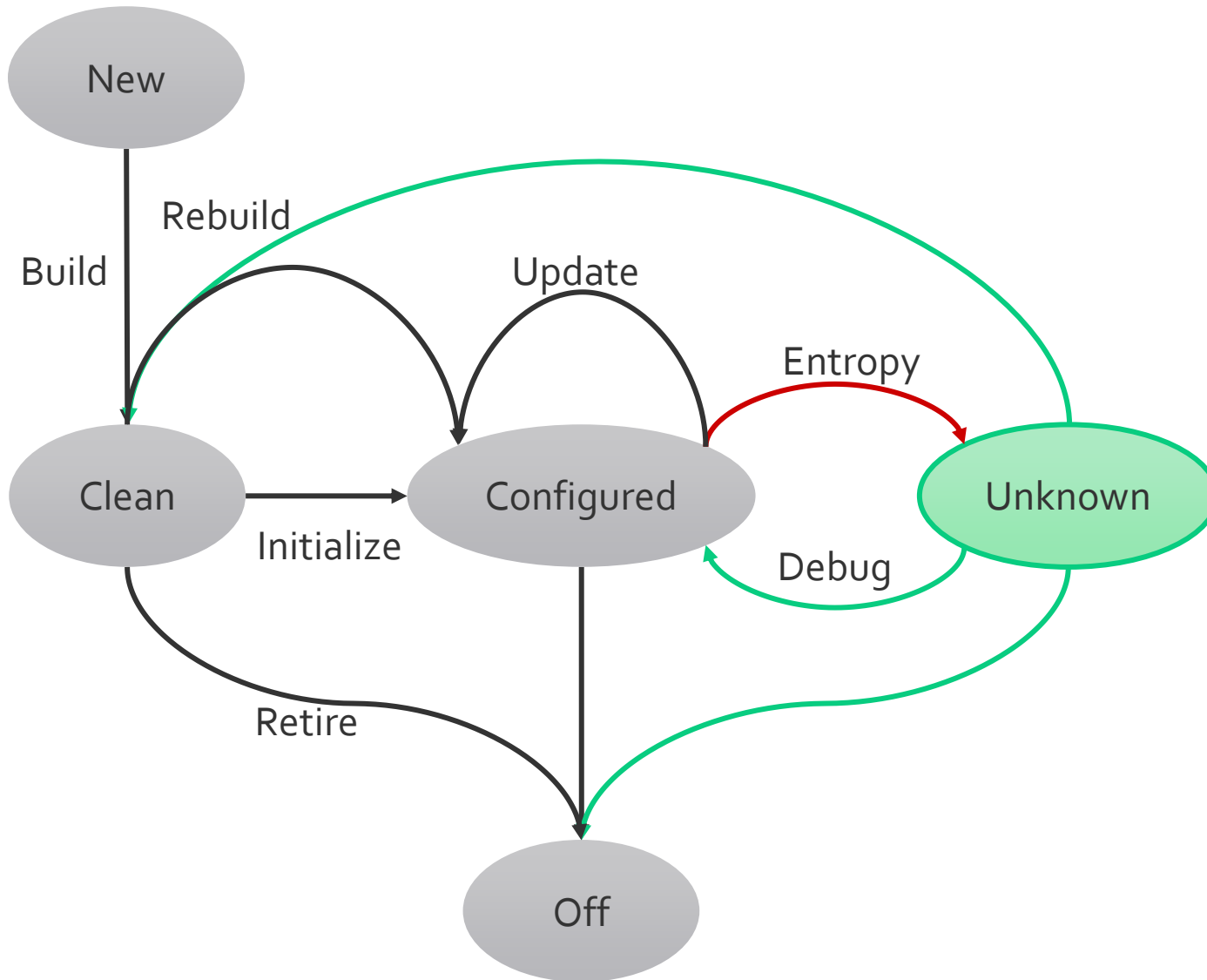
Configured State

- 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

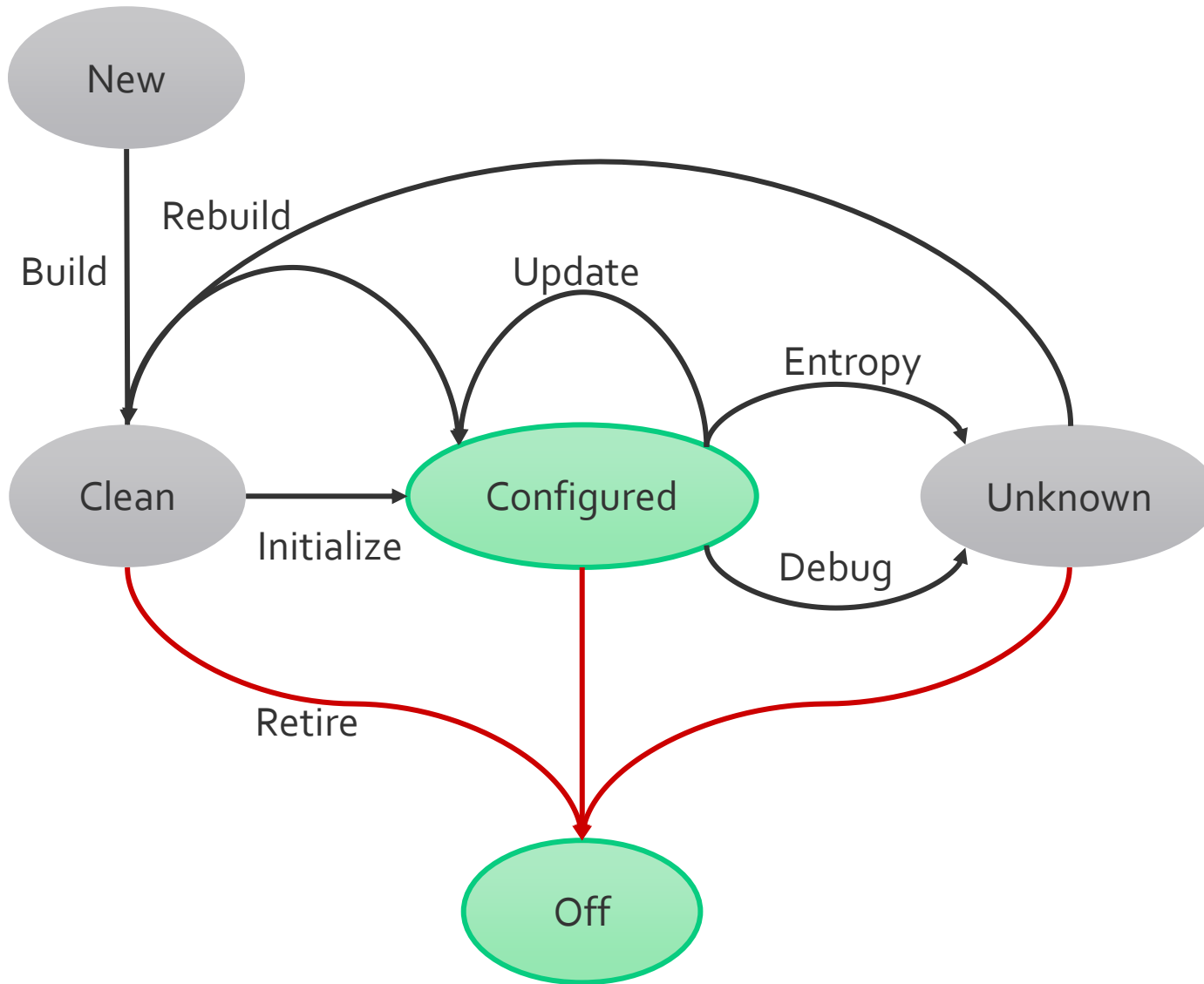
Entropy Process

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 Infrastructure

Retire Process

- Recover User Data
- Flush the Hard Disks
- Isolate from Infrastructure

- 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

- What is System Administration?, its Importance?
- Organization 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.



Computers



Networks



People

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



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

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**

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

Business Operations

Collaborations

Solve Many Problems

- **Business Depends on**

Internet

Intranet

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**

- **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

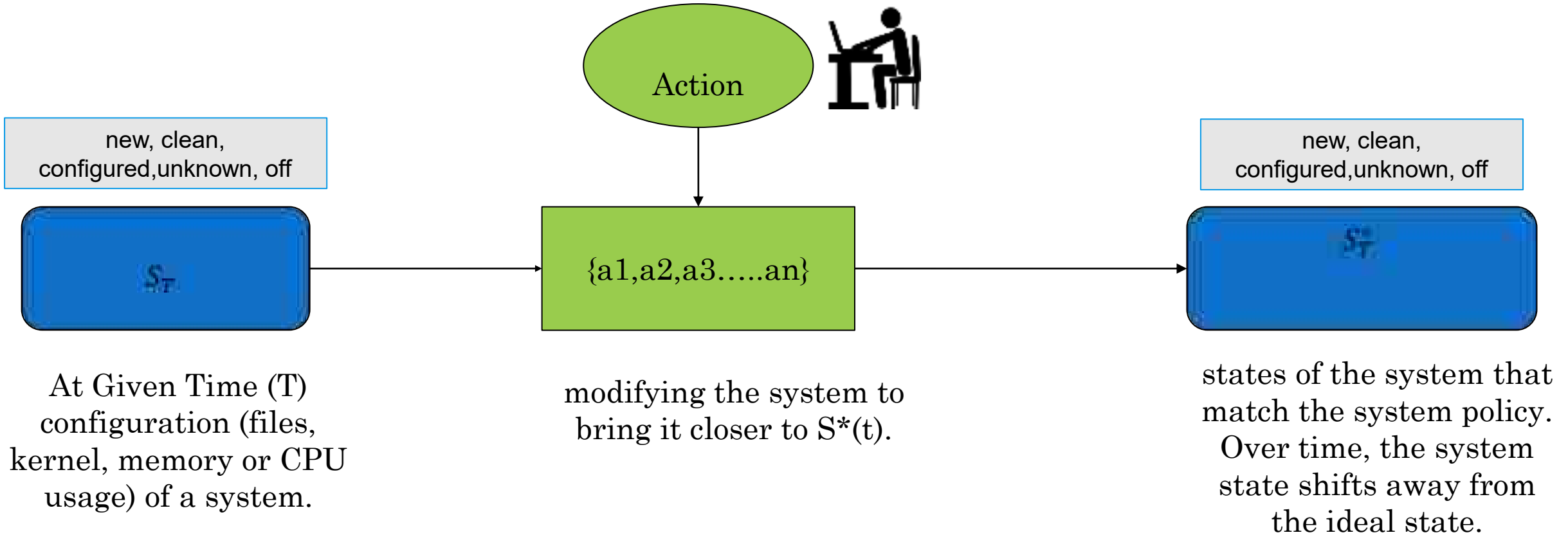
- Installation and Configuration
- Monitoring, Logging and Reporting
- Changing
- Supporting
- Provisioning
- Troubleshooting



Changes
System State

Formal Definition of System Administration

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



IT601 – System and Network Administration

Week # 2

System Administration Maturity

Arif Husen

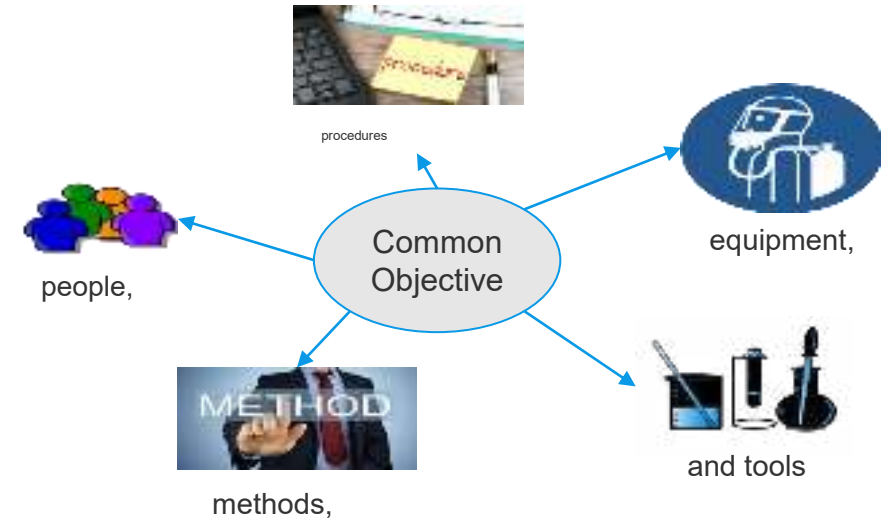
Department of Computer Science and Information Technology,

Virtual University of Pakistan

What is a process?

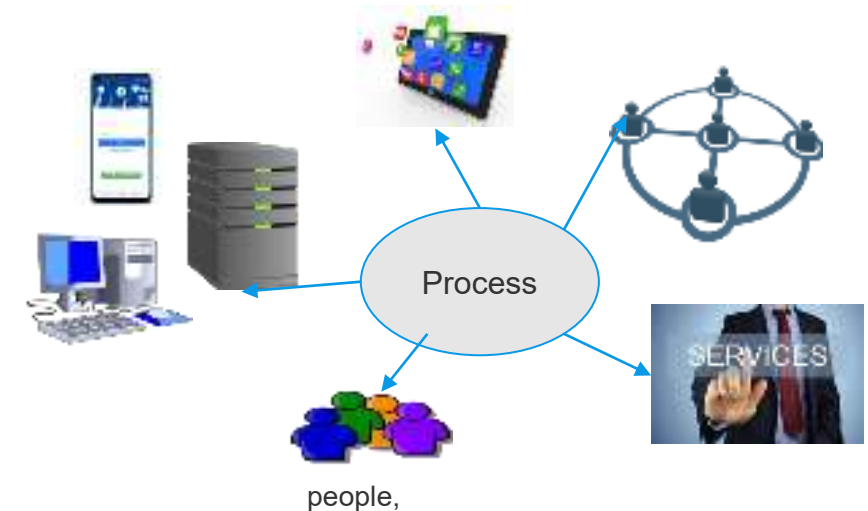
- Maturity model in IT?
- System Administration Maturity Model

- 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.

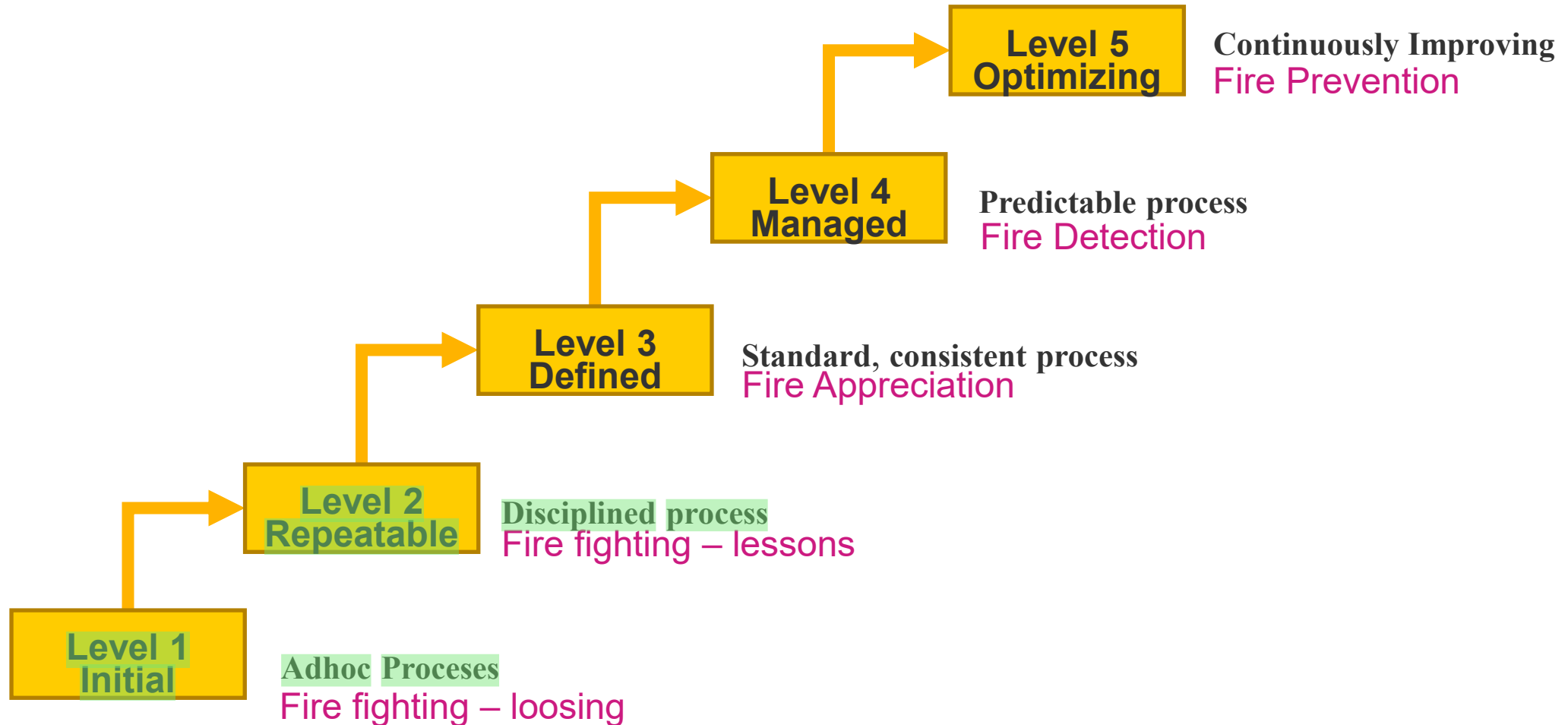


Several maturity models exists for IT and Business, Examples are....

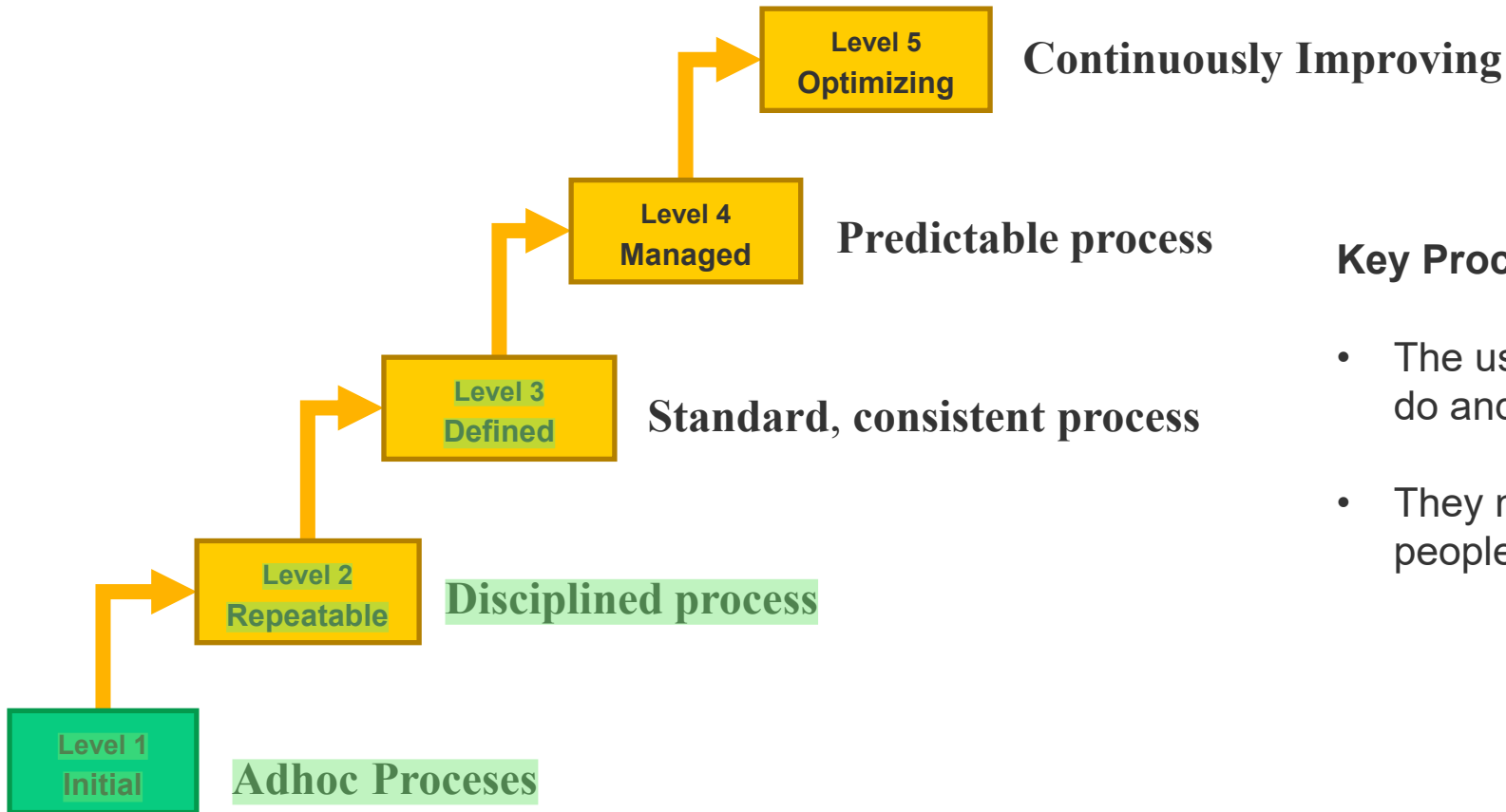
- CMMI
- ITIL Maturity Model
- Agile Maturity Model
- DevOps Maturity Model
- MDM Maturity Model

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

SAMM Maturity Levels



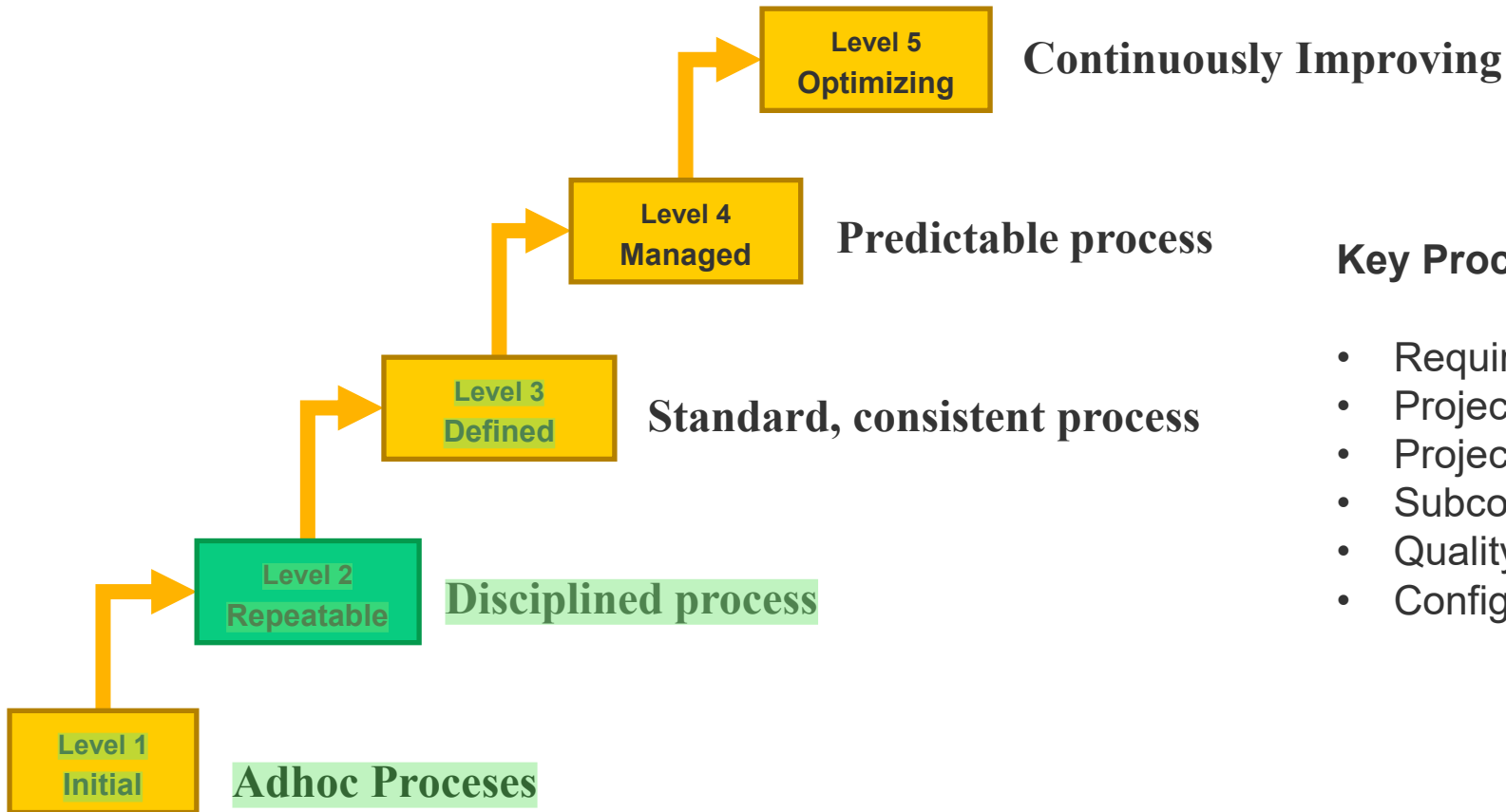
SAMM Maturity Levels



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.

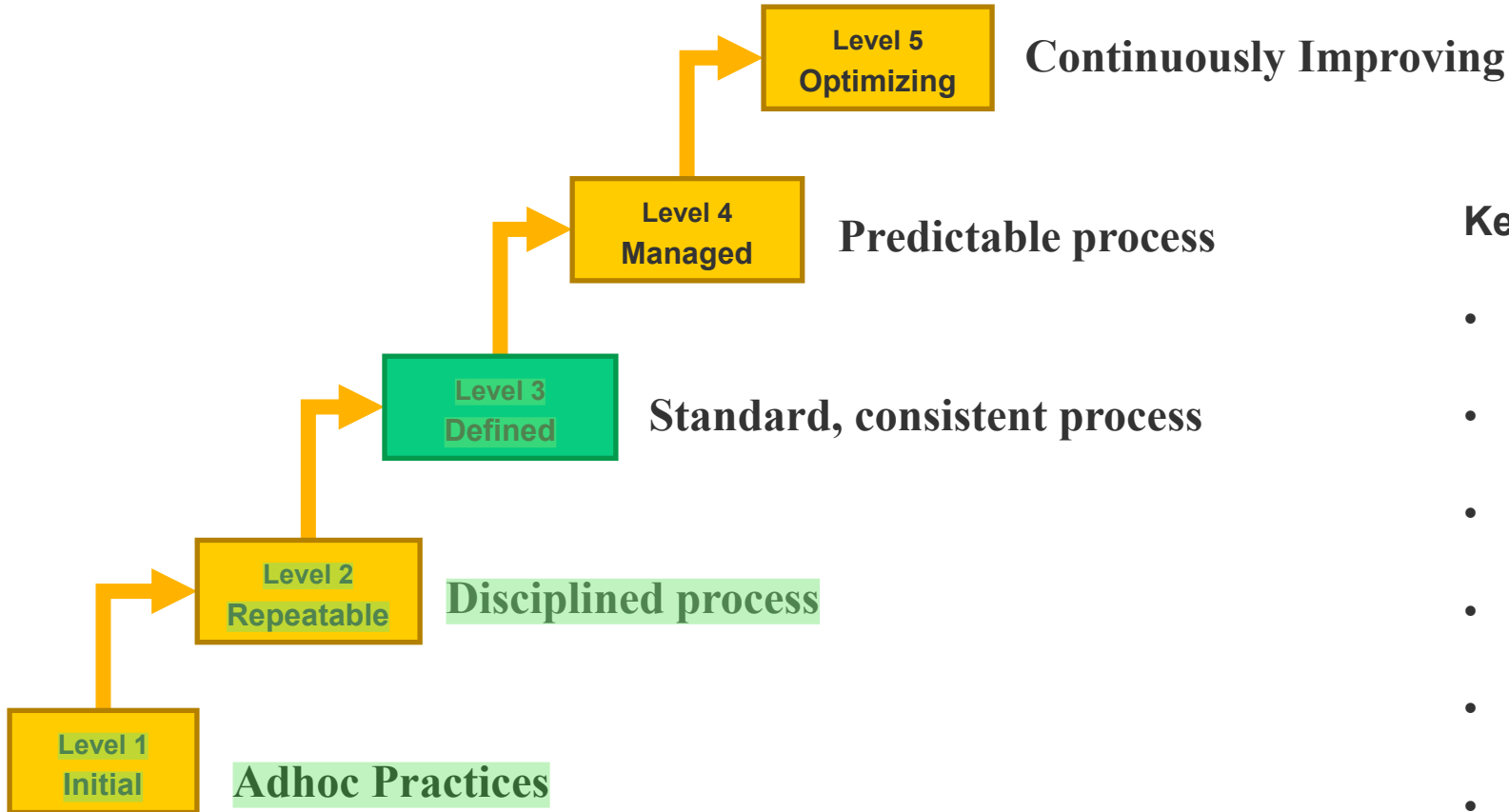
SAMM Maturity Levels



Key Process Area

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

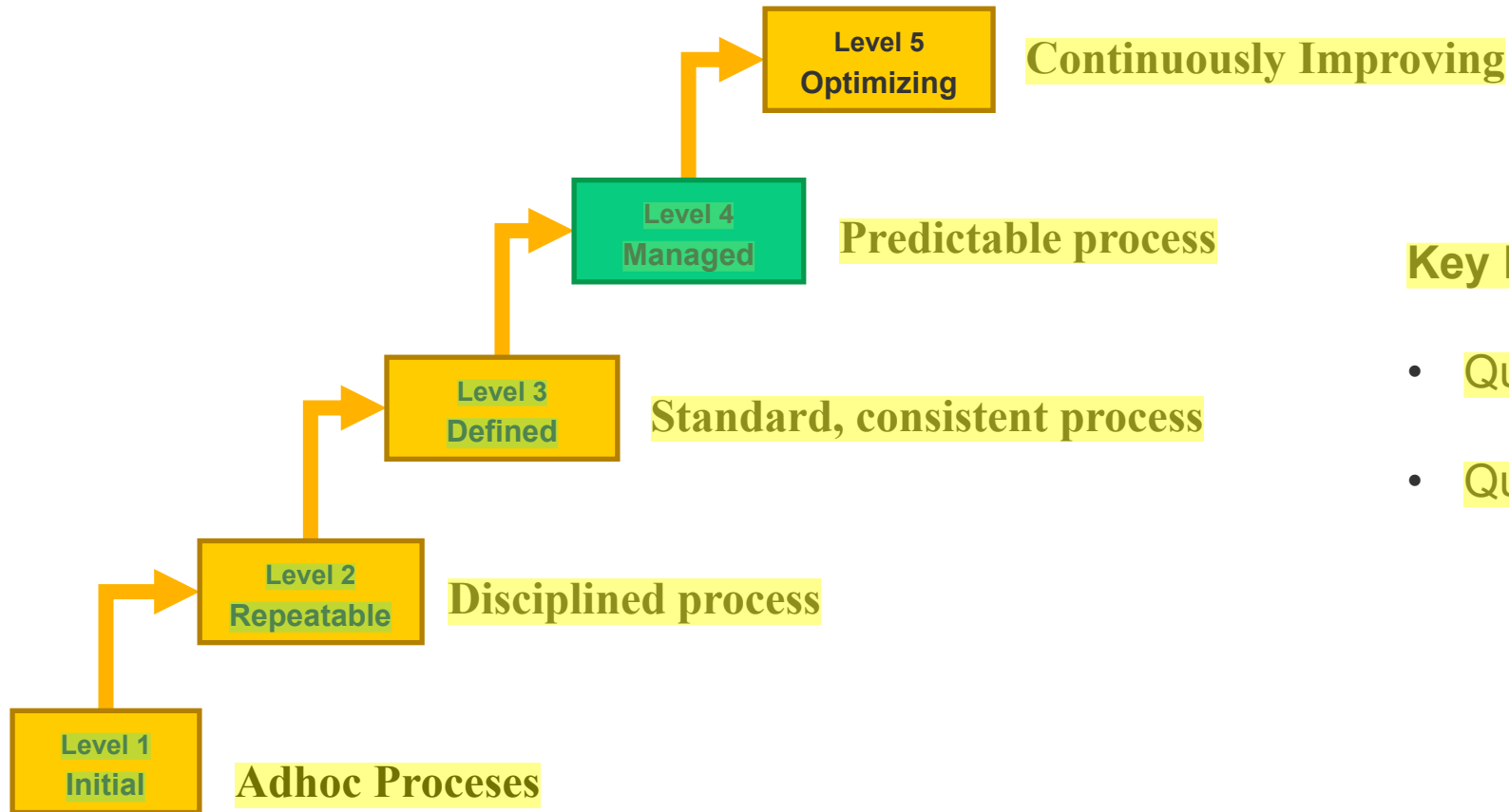
SAMM Maturity Levels



Key Process Area

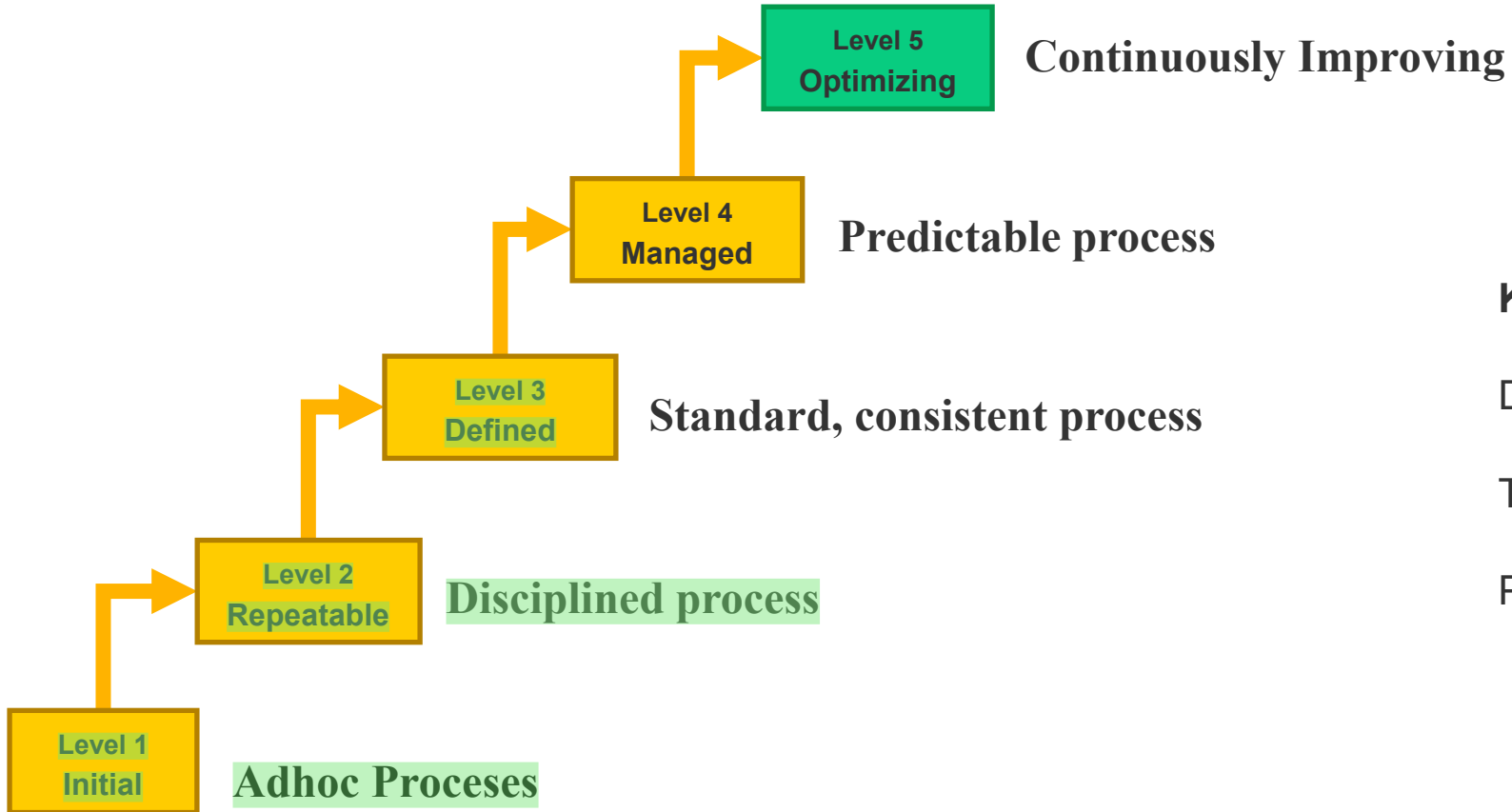
- Process Focus
- Process Definition
- Training
- Integrated Management
- System & Network Engineering
- Intergroup Coordination

SAMM Maturity Levels



Key Process Area

- Quantitative Process Management
- Quality Management



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

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

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

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 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 Positive measurable results
Qualified	8	Practice is integral part of process Consistent use across organization Positive long term results
Outstanding	10	Excellence in practice well recognized Consistent long term use 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

- 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

- The objective of studying reliability

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

To understand the causes for poor reliability,

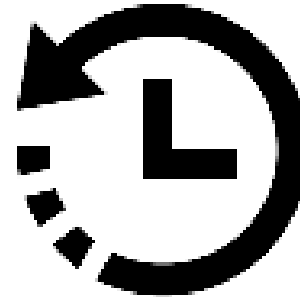
To deploy actions to improve the reliability of Organizations.

- Unreliable Systems contributes

Increase cost of ownership of the systems.

- 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



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.

The Reliability is calculated

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

$r(T)$ Reliability at time T

$N_w(T)$ Working Systems at time T

$N(T_0)$ Total Systems at time T0

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

- 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

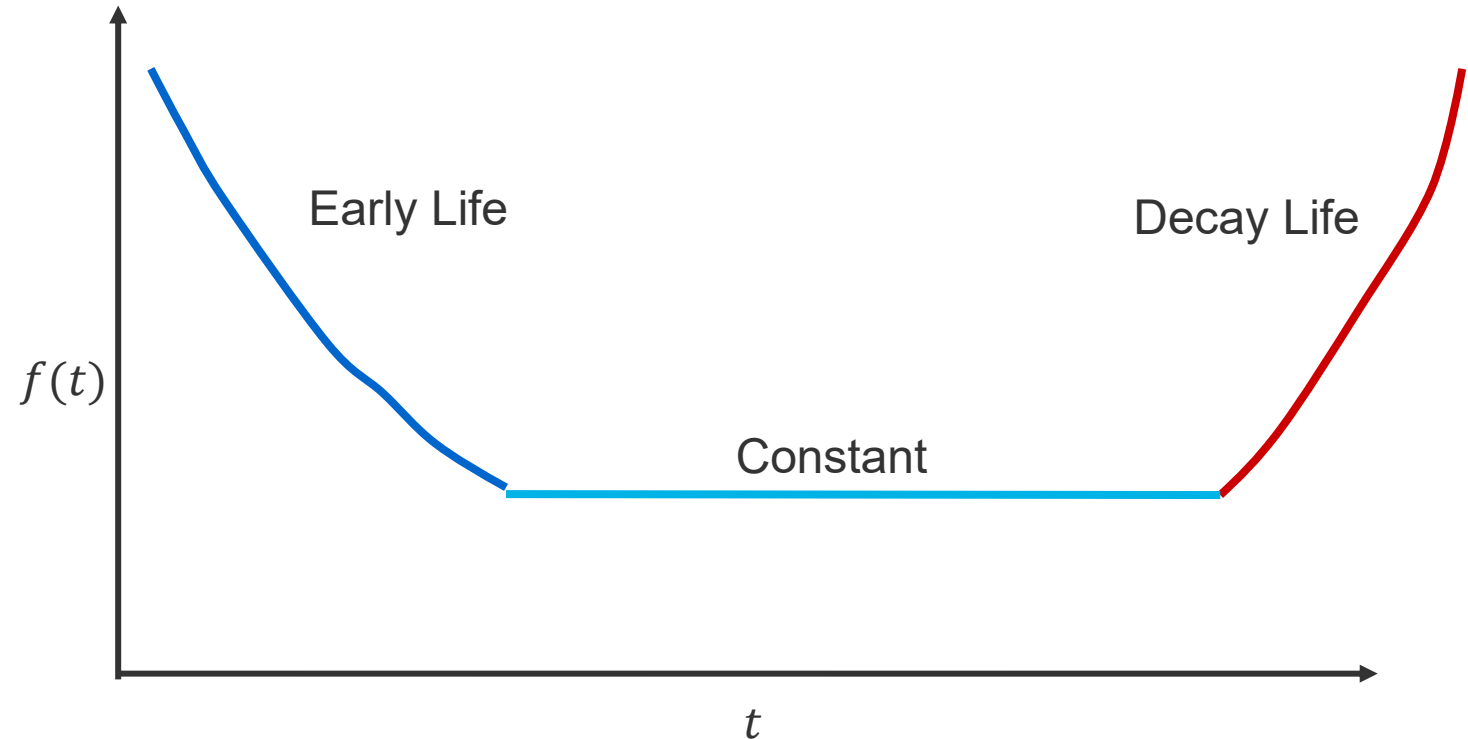
$$r(t) = e^{-\left(\frac{t}{\alpha}\right)^\beta}$$

α is called the scale factor which represents the characteristic life of the product

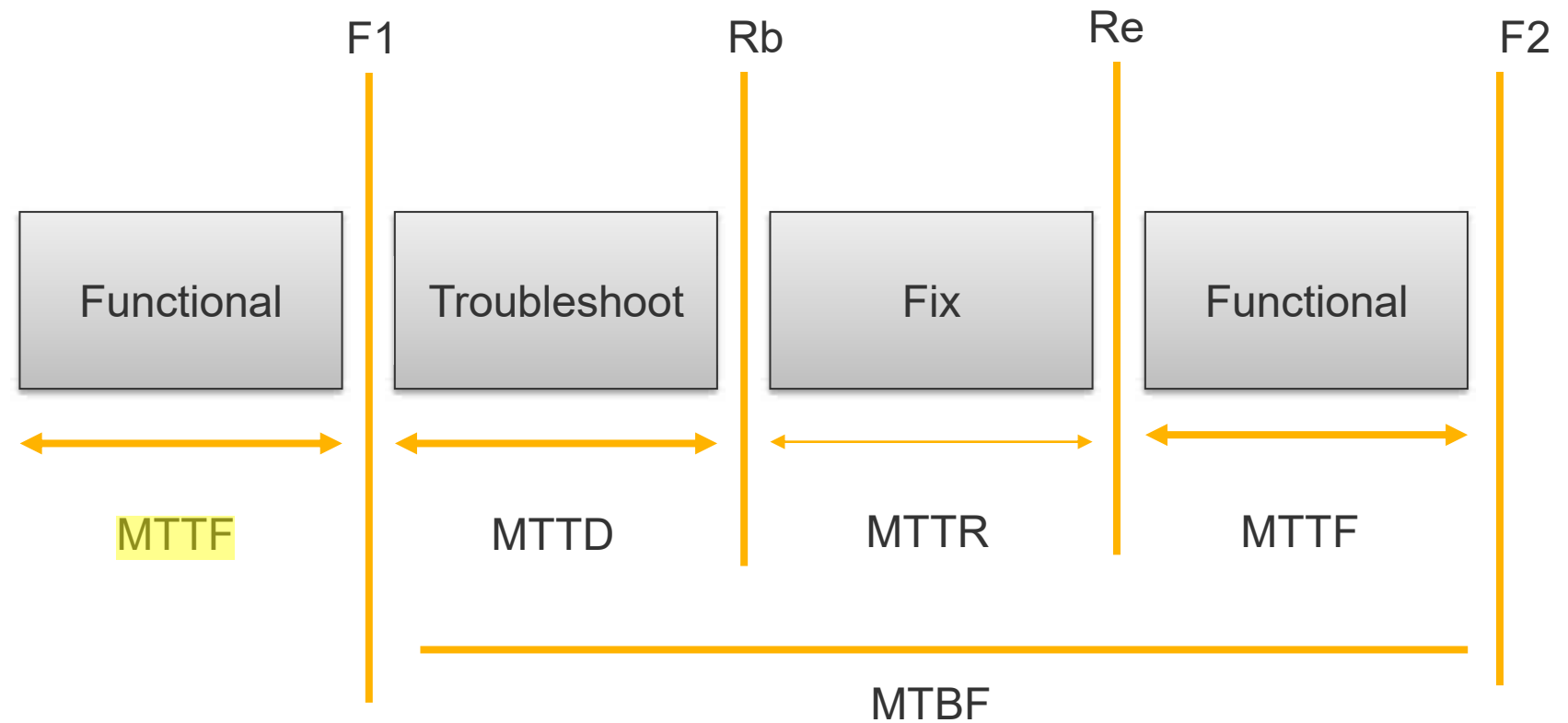
- which is the time at which 63% of the products have failed.

β 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



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



- 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.

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

- The frequency of successful repair operations performed on a failed component per unit time and denoted μ .
- Repair rate is defined mathematically as follows:

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

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

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

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

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

$$MTBD = MTTF + MTTR$$

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.

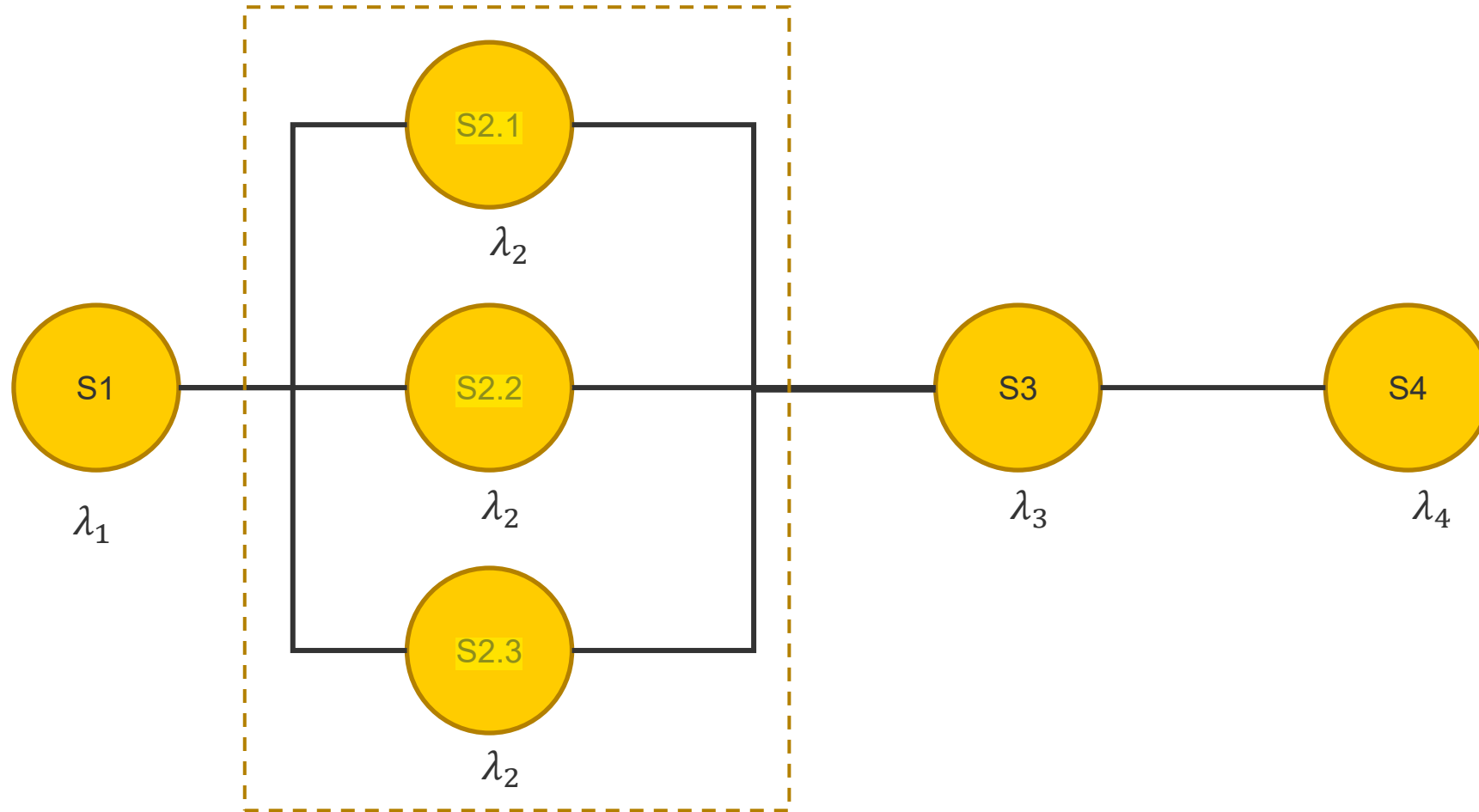
$$MTTR = \frac{\textit{Total Maintenance Hours}}{\textit{Total Repairs}} = \frac{1}{\mu}$$

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

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

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

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



- 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.

- 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.

- 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.

- <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

1. Services

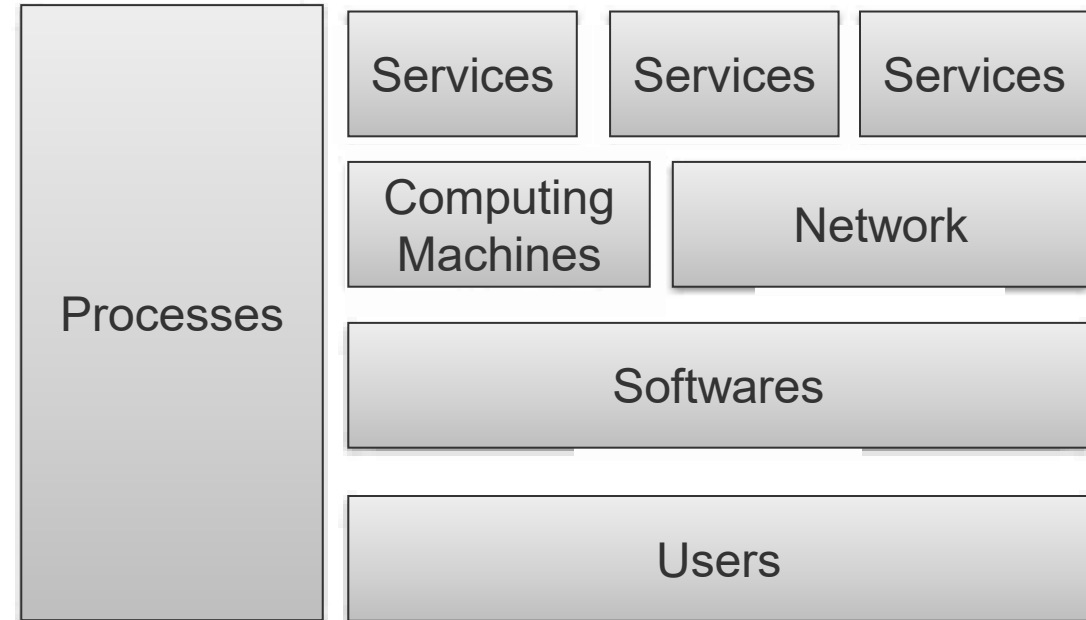
2. Computing Machines

3. Network

4. Processes

5. Softwares

6. Users



Planning and Design

- 1) Architecture
- 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.

Adding User
Accounts

Namespace
management

Removing user
accounts

Consistency

automation

Startup (dot) files

Username

UIDs

SSI

Multiple namespaces

Consistency

automation

Many accounts

different systems

4.2 - Hardware Management

Adding and removing hardware

Configuration, cabling, etc.

Purchase

Evaluate and purchase servers + other hardware

Capacity planning

How many servers? How much bandwidth, storage?

Data Center management

Power, racks, environment (cooling, fire alarm)

Virtualization

When can virtual servers be used vs. physical?

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

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

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

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

Week # 3

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





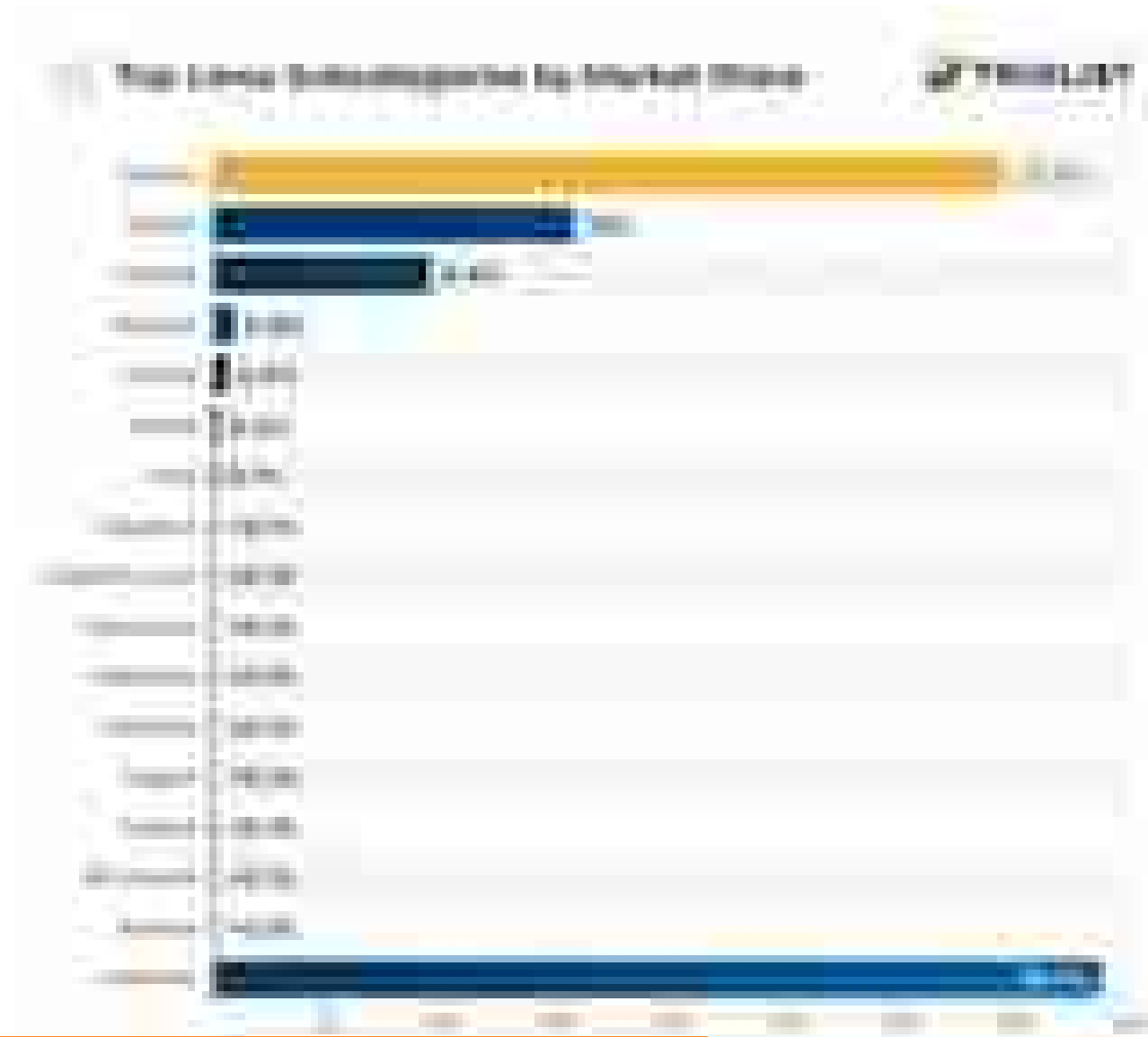
Linux Usage Worldwide on Servers







Usage of Linux Distributions



Quick Comparison

OS



Architecture

Centered around
the Linux kernel

Cost

Free
open-source
software

Security

- Highly secure against malware cyber threats

Support

Large comm. supports
Answer commonly asked questions



- Based on the Windows NT architecture

- Owned by Microsoft,
- a licensing fee per user

- More prone to
 - hacking attempts
 - cyber threats

- Community and long-term customer support
- Great documentation

Quick Comparison

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

➤ 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

- 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?

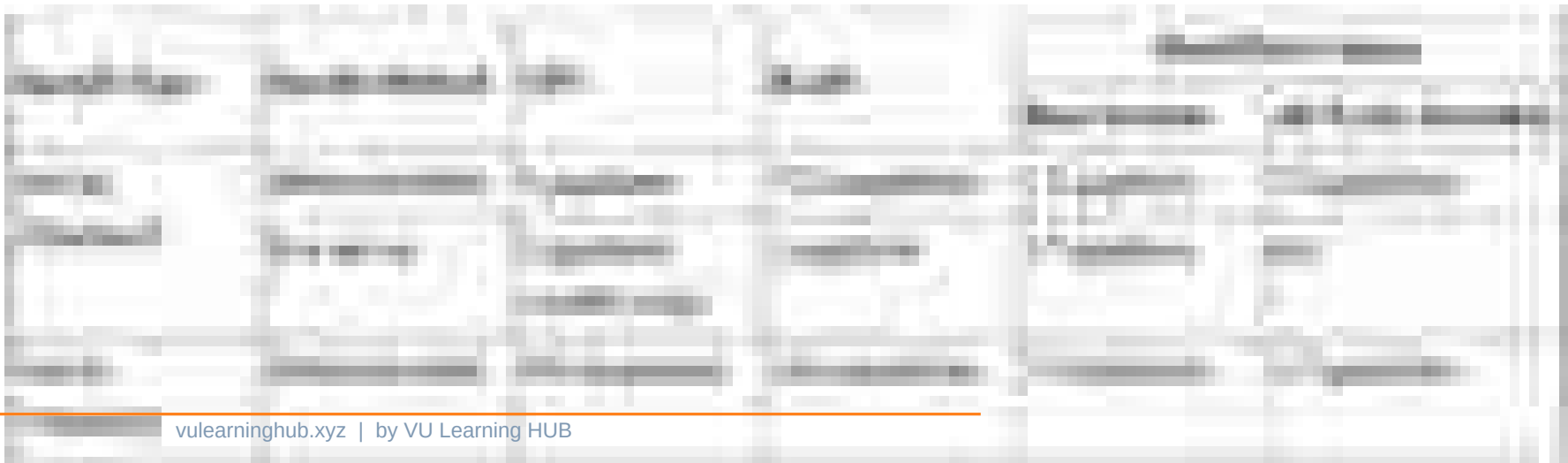
- Ubuntu Server is a server operating system developed by Canonical that runs on all major architectures



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



- Ubuntu Server has these minimum requirements:



Component	Minimum Requirement
Processor	1 GHz or faster, 64-bit
Memory	2 GB
Disk Space	25 GB
Network	100 Mbps or faster
Operating System	Ubuntu Server 18.04 LTS
Kernel	4.15 or later
Libraries	glibc 2.27 or later
Tools	gcc 7.4 or later, make 3.81 or later

- 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.

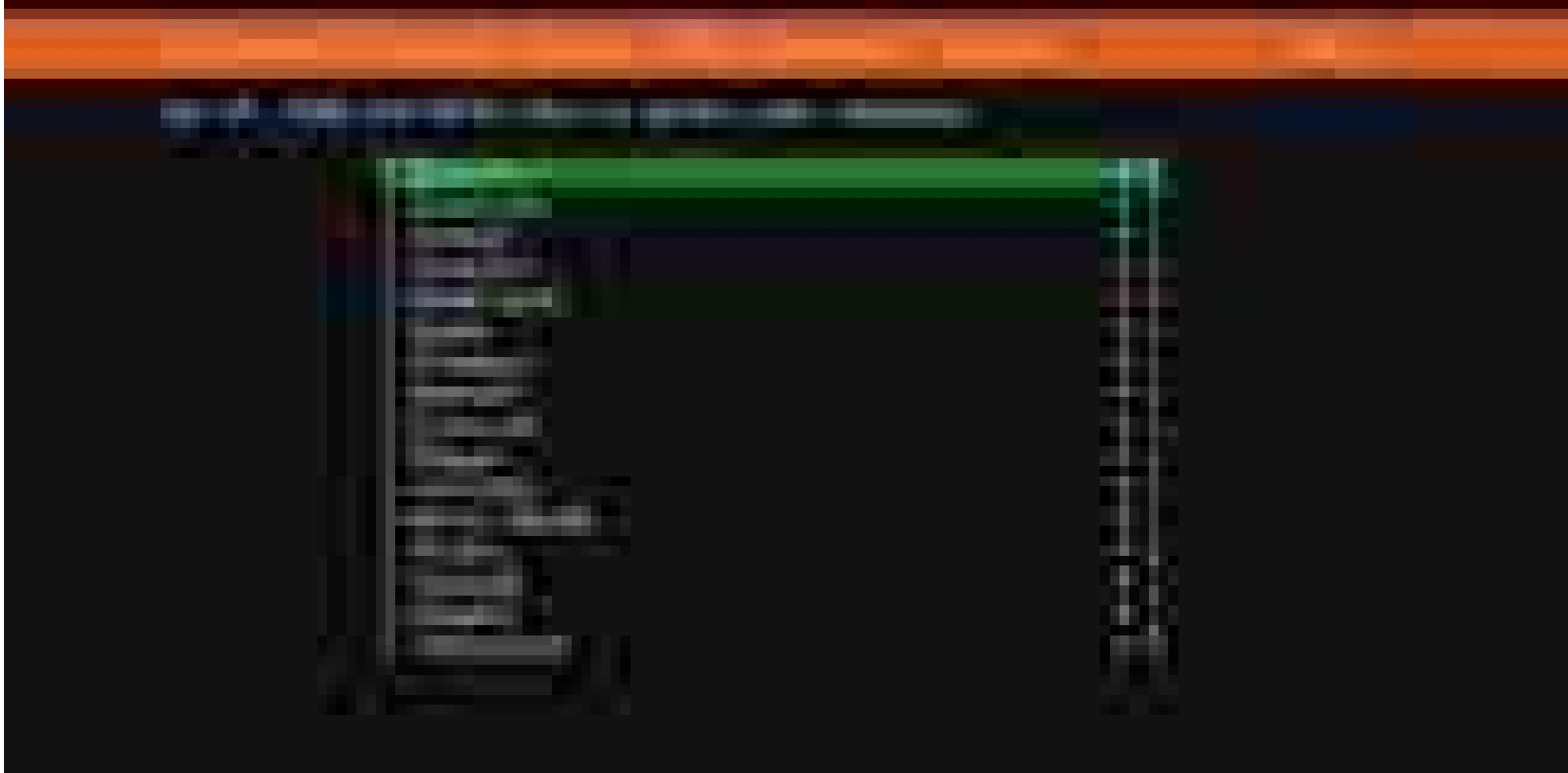
➤ Preparing install media

- 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

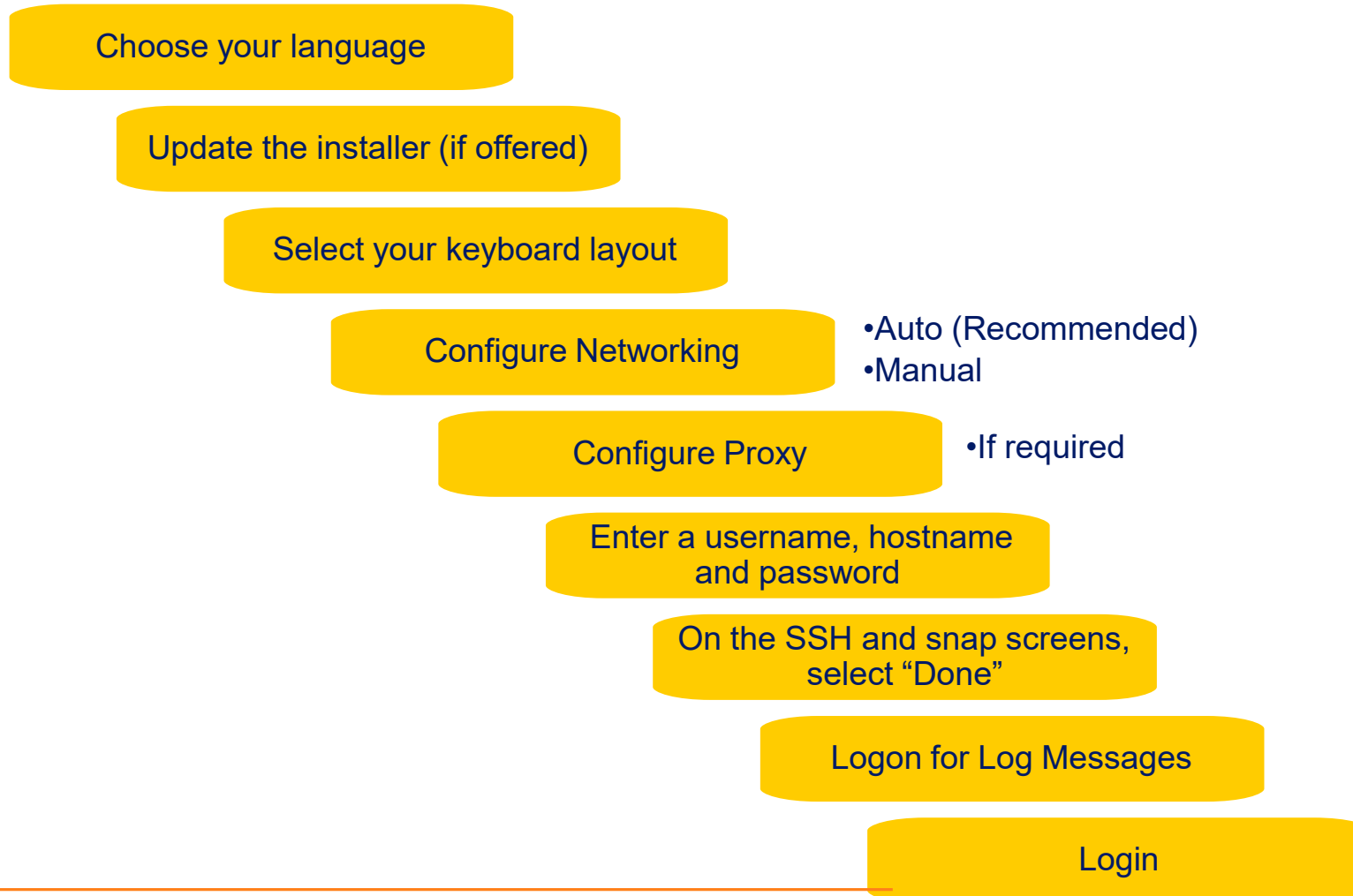
- Plug the USB stick into the system to be installed and start it.
- Most computers will automatically boot from USB or DVD, though in some cases this is disabled to improve boot times. If you don't see the boot message and the "Welcome" screen which should appear after it, you will need to set your 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

- **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 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

- 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
```

- **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.

➤ 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.

- 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

- 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*

- 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";  
};
```

- 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.

➤ 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 `apicron`. `apicron` 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 `apicron` package, in a terminal enter:

```
sudo apt install apicron
```

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

```
EMAIL="root@example.com"
```

- 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

- 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 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.

ip a



sudo lshw -class network



- 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 logical name.



- ```
sudo ethtool eth4
```

[illegible]

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

- 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
```

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

```
ip address show dev enp0s25
```



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
```



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

`ip route show`



- 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 recommended, 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`

- 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
```

- 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
```

- 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



**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 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

- Generally, the file `/etc/resolv.conf` is a static configuration file that rarely needed to be changed or automatically changed via DHCP client hooks.
- **Systemd-resolved** handles name server configuration, and it should be interacted with through the `systemd-resolve` command.
  - **systemd** 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:





- 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:



- 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  |

- 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](#).

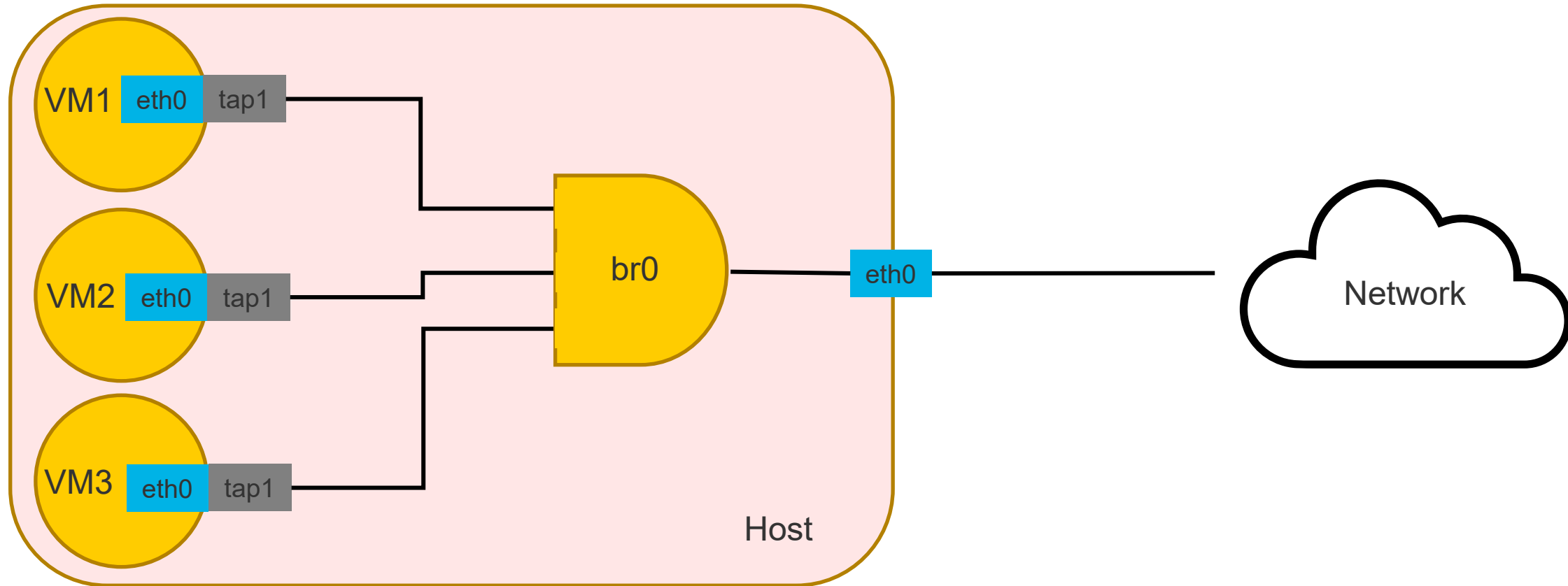
**hosts: files mdns4\_minimal [NOTFOUND=return] dns mdns4**

- **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 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.

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

```
network:
 version: 4
 renderer: networkd
 ethernets:
 eth0:
 dhcp4: no
 bridge:
 name: br0
 interfaces:
 - eth0
 dhcp4: no
 dhcp6: no
```

- 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
```

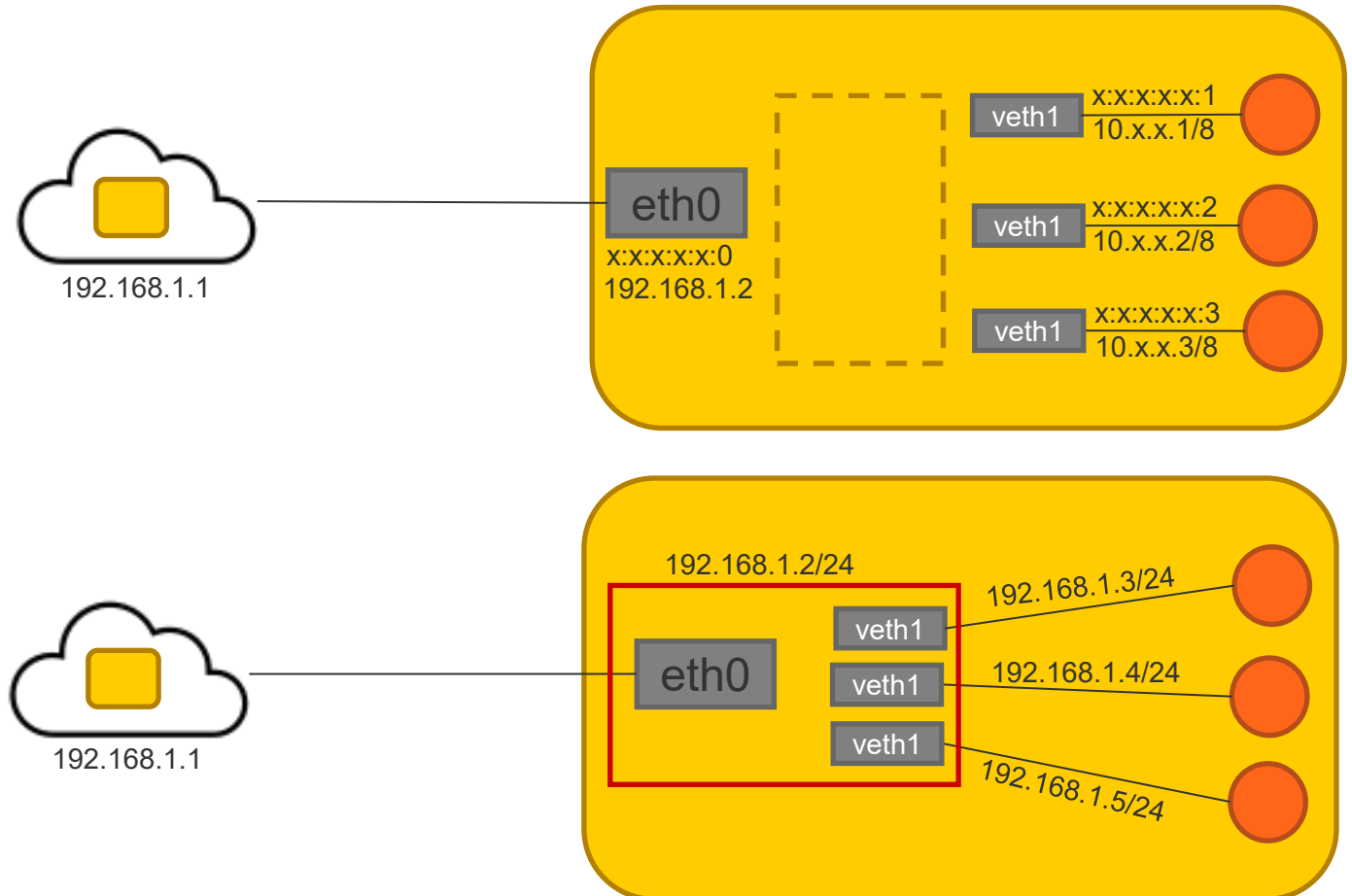
- creating a new bridge device, we'll create one called br-sna1

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

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

```
brctl addif br-sna1 NetDeviceName
```

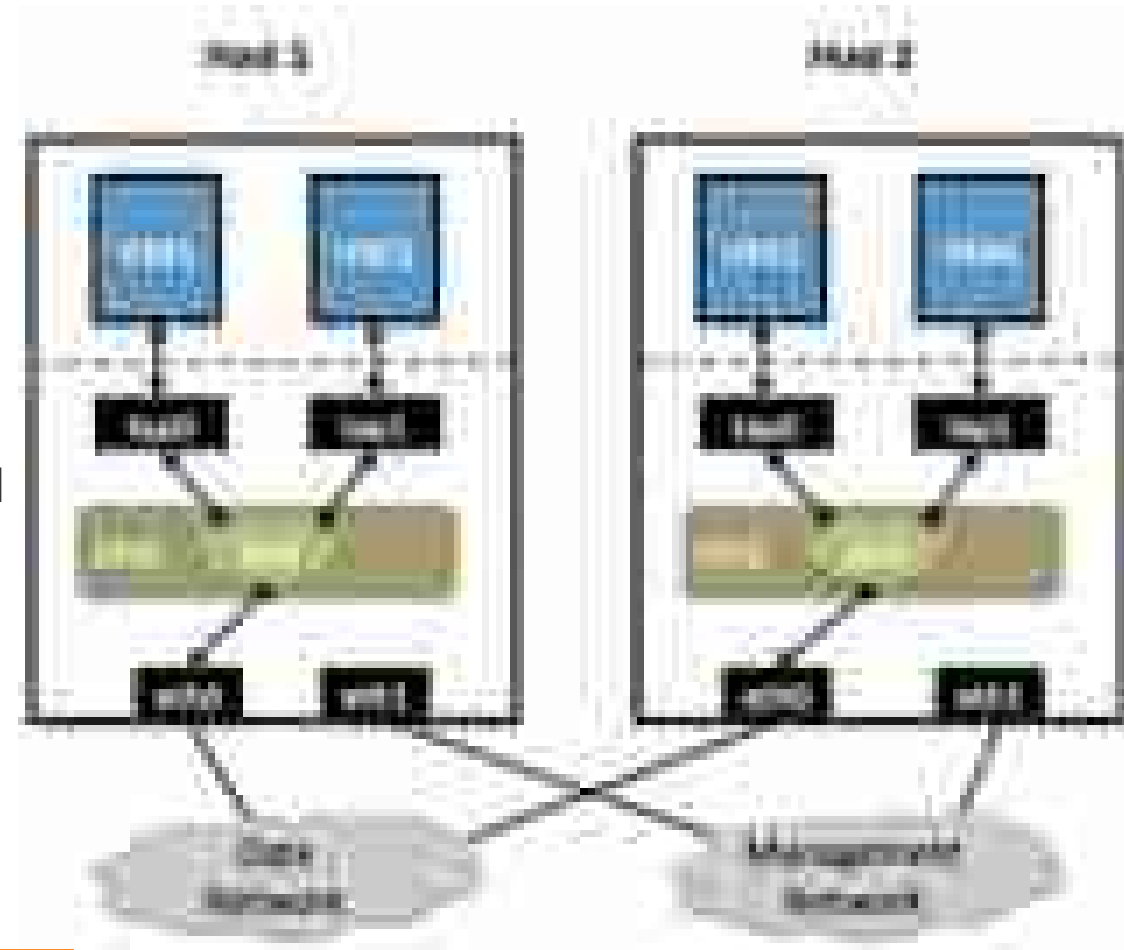
- See man `brctl` for more information



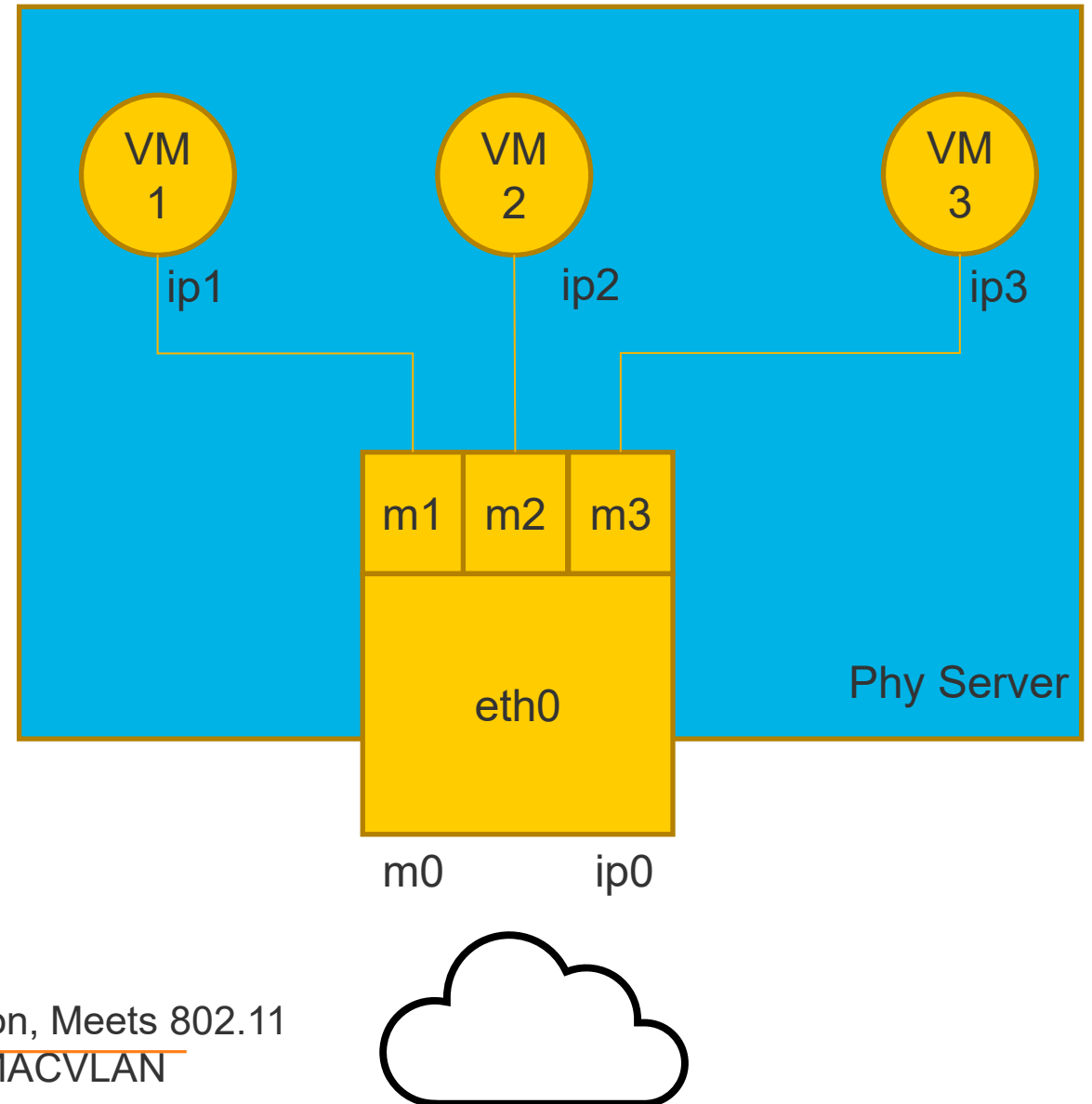


# Open vSwitch Bridge devices

- 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 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**

**Week # 4**

# **User Management**

Arif Husen

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

**Virtual University of Pakistan**

- 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.

### sudoer rights

root

- Highest Privileges

admin

- Middle Privileges

### No sudoer rights

limuser

- Lowest Privileges

- 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.

- 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.



- 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`

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**

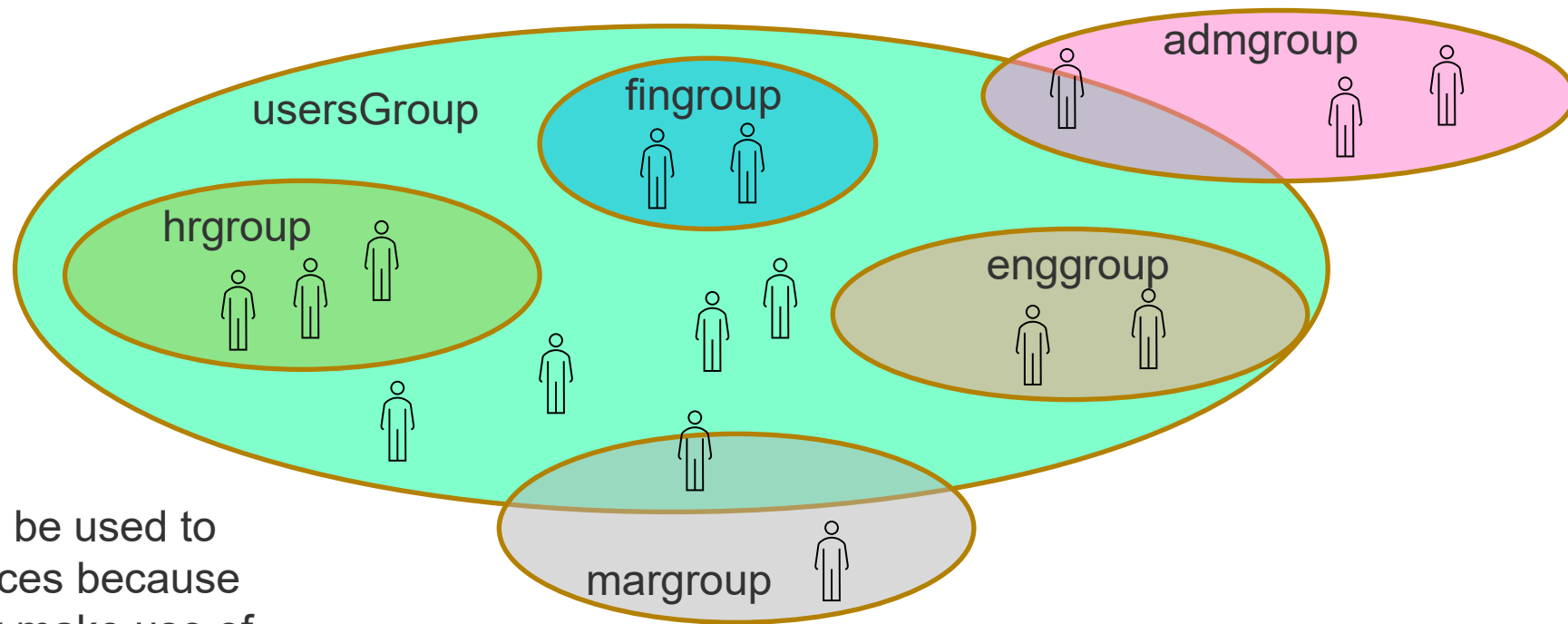
- Deleting an account does not remove their respective home folder. It is up to you whether or not you wish to delete the folder manually or keep it according to your desired retention policies.

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.

➤ 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.





- 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 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.                                                                 |

- 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**

- 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`

|           |      |   |   |       |   |   |       |   |   |              |            |             |           |                |  |  |           |  |  |  |  |  |  |  |  |  |  |
|-----------|------|---|---|-------|---|---|-------|---|---|--------------|------------|-------------|-----------|----------------|--|--|-----------|--|--|--|--|--|--|--|--|--|--|
| Numeric   | 7    |   |   | 5     |   |   | 5     |   |   |              |            |             |           |                |  |  |           |  |  |  |  |  |  |  |  |  |  |
| Binary    | 1    | 1 | 1 | 1     | 0 | 1 | 1     | 0 | 1 |              |            |             |           |                |  |  |           |  |  |  |  |  |  |  |  |  |  |
| d         | r    | w | x | r     | - | x | r     | - | x | 2            | userid     | groupid     | 4096      | 2023-1-1 20:03 |  |  | username  |  |  |  |  |  |  |  |  |  |  |
| File Type | User |   |   | Group |   |   | Other |   |   | No. of Links | Owner User | Owner Group | File Size | Last Modified  |  |  | File name |  |  |  |  |  |  |  |  |  |  |

- 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
```

- 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



- 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**~~

- 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`

# IT601 – System and Network Administration

< End >

Arif Husen

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

**IT601 – System and Network Administration**

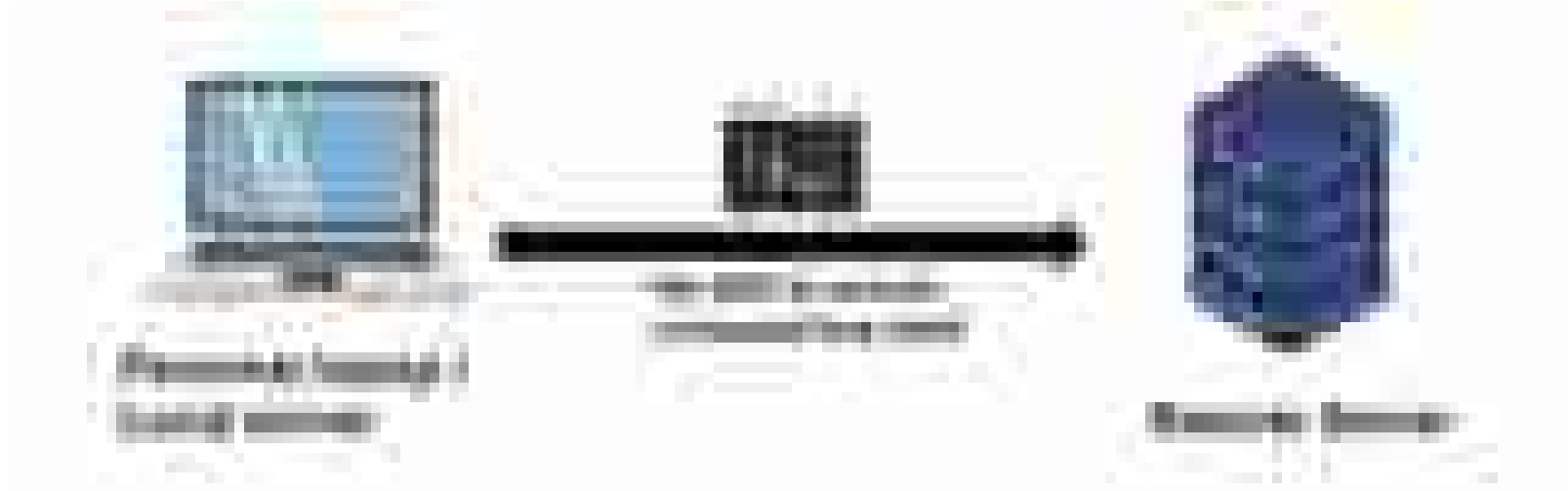
# **Remote Administration**

Arif Husen

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

**Virtual University of Pakistan**

- 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.

- 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.

➤ 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
```

```
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 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
```

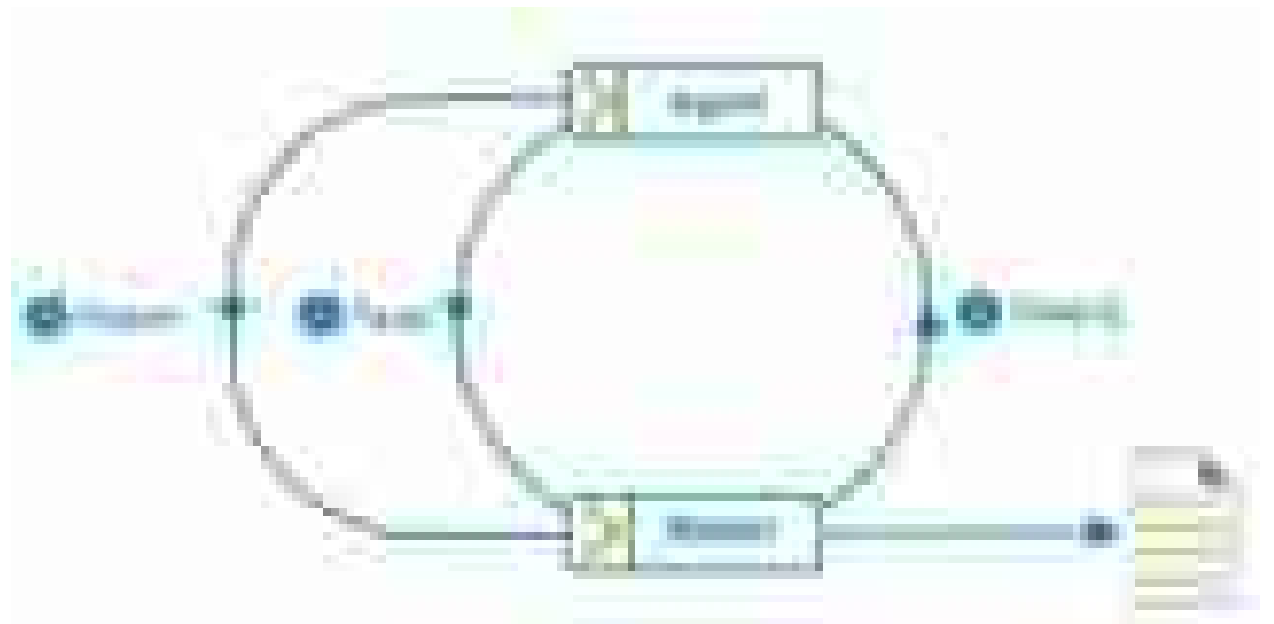
- Finally, double check the permissions on the `authorized_keys` file, only the authenticated user should have read and write permissions. If the permissions are not correct change them by:

```
chmod 600 ~/.ssh/authorized_keys
```

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

- Puppet 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 utilizes a client/server architecture consisting of the Puppet Master and Puppet Agents. Puppet Agents use 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.



- 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 'puppetserver1.example.com' {
 include apache2;
}
```

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

```
sudo systemctl restart puppetmaster.service
```

- 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 its dependencies will be installed on the Puppet client.

- 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.

- 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.



- 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.

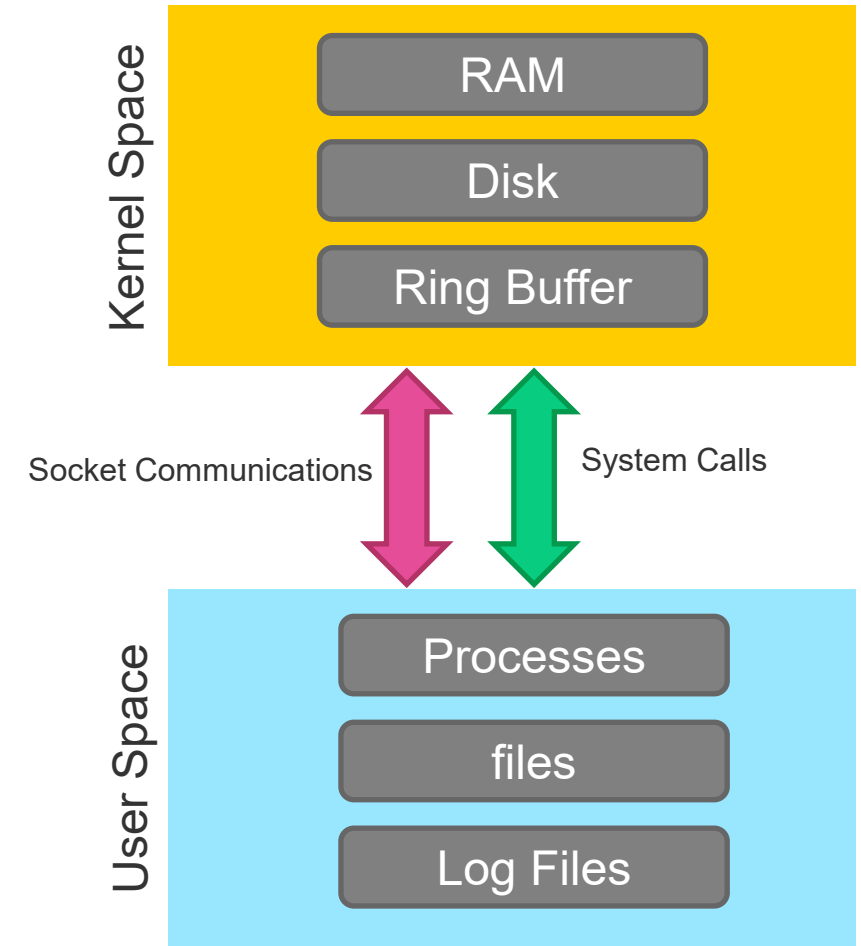
# IT601 – System and Network Administration

## Logging

Arif Husen

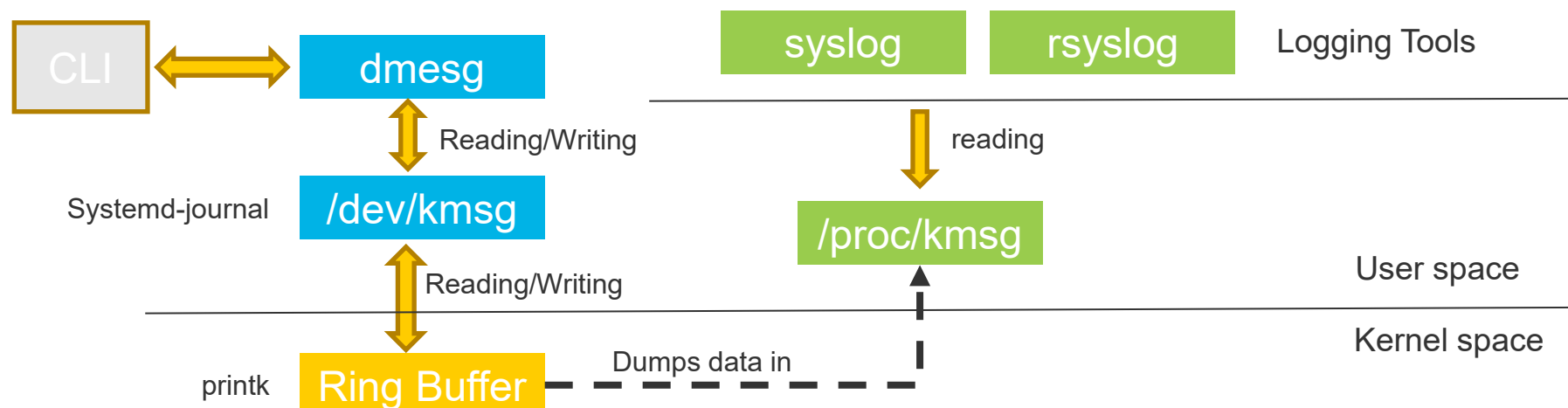
Department of Computer Science and Information Technology,  
Virtual University of Pakistan

- Logging 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.



- 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



- 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**

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

- **Application logs**

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

- **Non-human-readable logs**

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

- 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`.

- 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.

- 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 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**

- 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
```

- 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

- Default mode, also called Normal mode.
- To switch from one mode to another, you have to come to Command Mode
- The commands that you run without any prefix (colon) indicate that you're running the command in command mode.

## Insert Mode

- to edit the contents of the file.
- You can switch to insert mode by pressing i from command mode.
- You can use the Esc key to switch back to command mode.

## Command-Line Mode

- To execute commands
- the commands in this mode are prefixed with a colon (:)
- Switch to this mode by pressing : (colon) in command mode

## Visual Mode

- to visually select some text and run commands over that section of code
- switch to this mode by pressing v from the command mode.

## ➤ 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

## ➤ Editing Commands

- d ...delete the characters from the cursor position up to 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



- 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**

## ➤ 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

- `$0` - The filename of the current script.
- `$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.

- **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]}`

## ➤ Arithmetic Operators

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

## ➤ Relational Operators

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

## ➤ 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.      |

---

## ➤ File Test Operators

---

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

---

## ➤ 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
```



## 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
```

## ➤ 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 returned 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**

**Week # 5**

**Servers**

Arif Husen

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

**Virtual University of Pakistan**

A group of computer machines in an organization general refer 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**



- **Operating System**



- **Software packages**





# 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

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.

- 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
- ability 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

## 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

## Budget

- key factors to consider is your budget

## Workload

- The server load

## 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

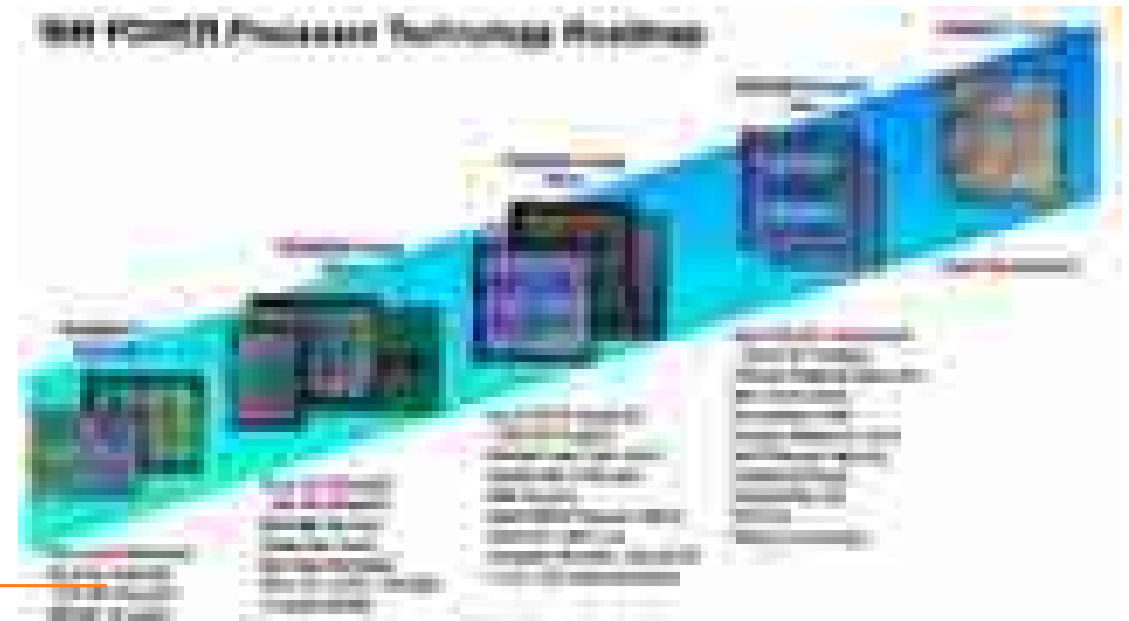
- (a) - 4, 6, 8, or 12 cores @ 1.4 to 3.2 GHz
- (b) - Up to 12 MB L3 cache

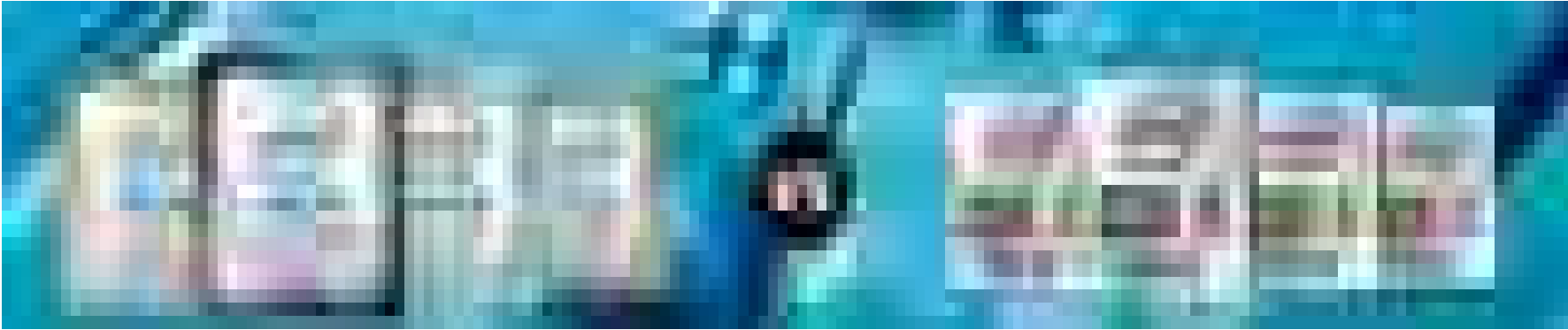
## 3 - IBM Power 7/8/9/10...

- (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 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



## 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**

Server memories are concerned with

- RAM
- CPU Cache

Two Major aspects of server memory are

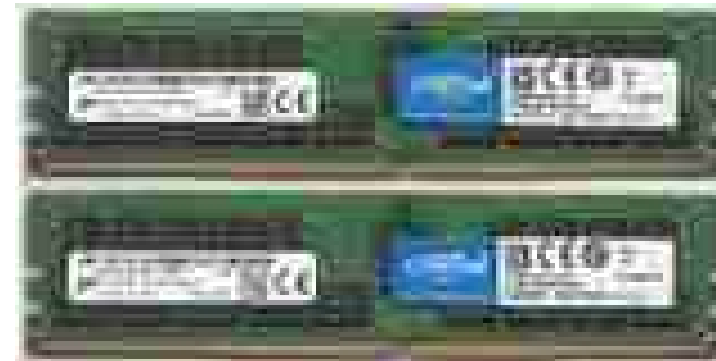
- Large Capacity
- Higher speed

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

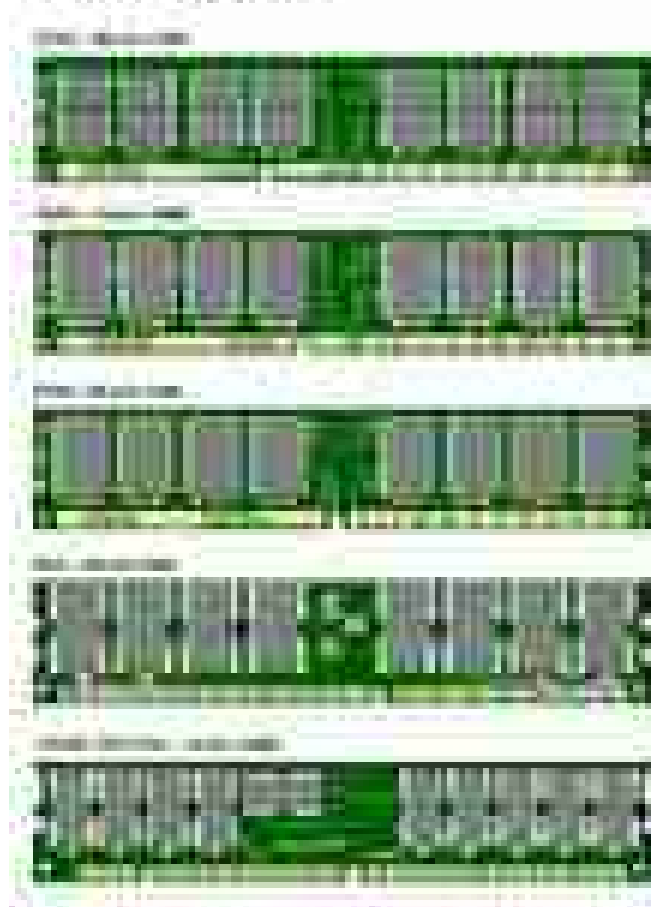
- Load Reduced dual in-line memory module

## SODIMM

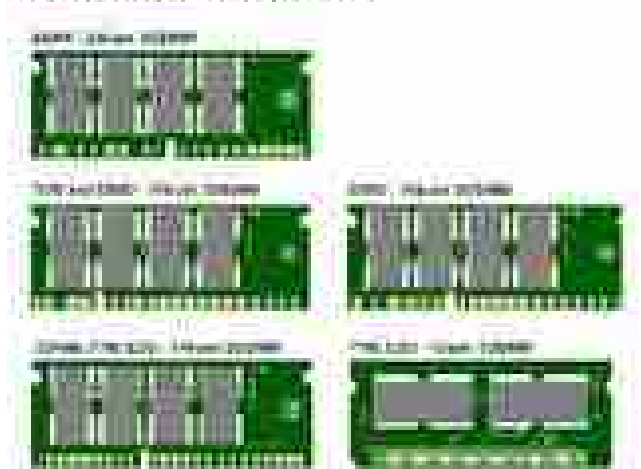
- Used with laptops

## MicroDIMM

### DIMM MODULES



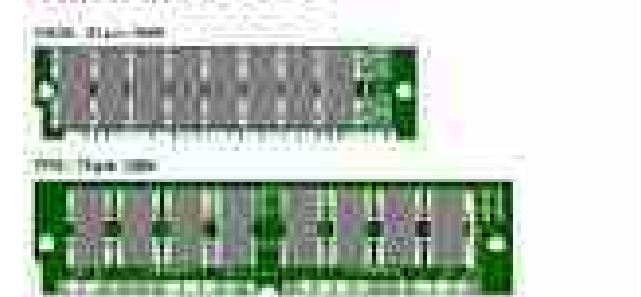
### SODIMM MODULES



### RIMM MODULE



### SIMM MODULES



# 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



**IT601 – System and Network Administration**

# Cache Memory

Arif Husen

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

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

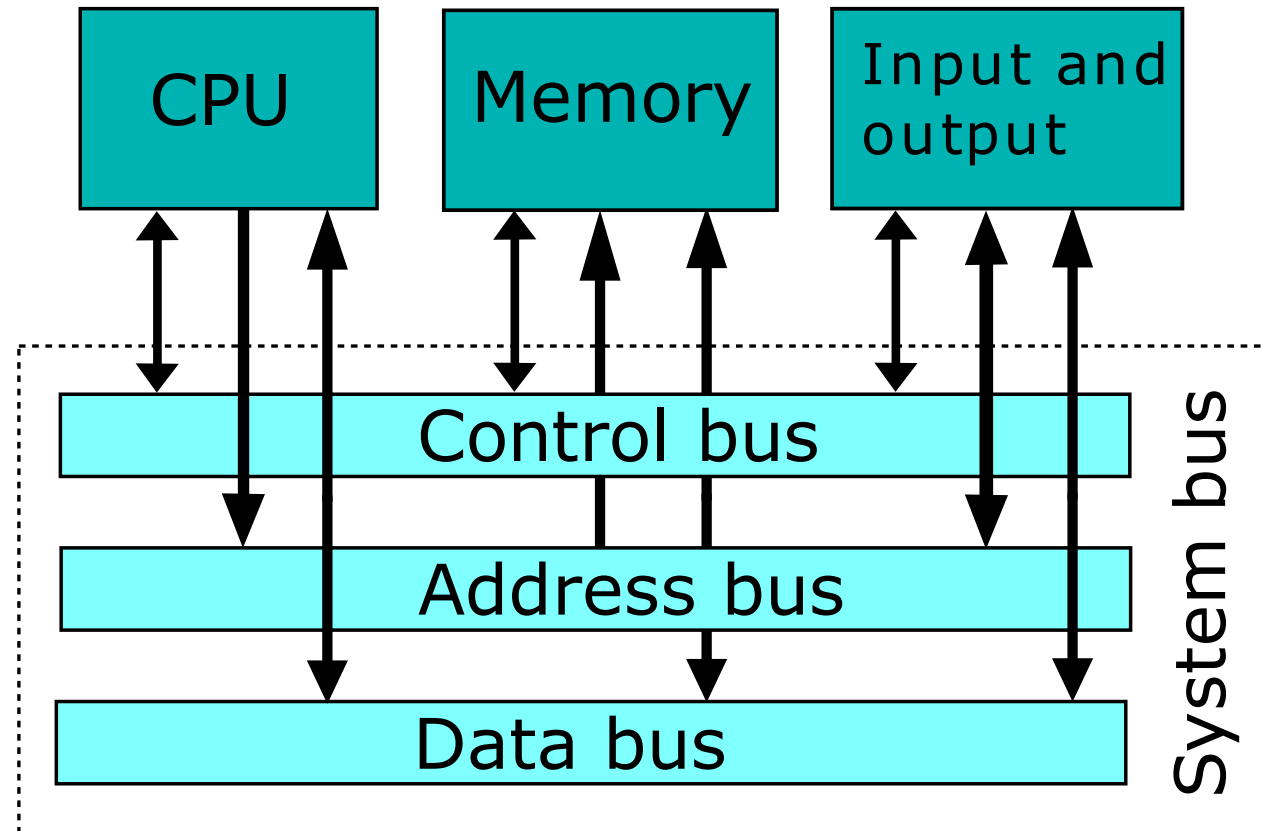
## System Bus

Arif Husen

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



- Data sharing
- Addressing
- Power
- Timing



## 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

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)**

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

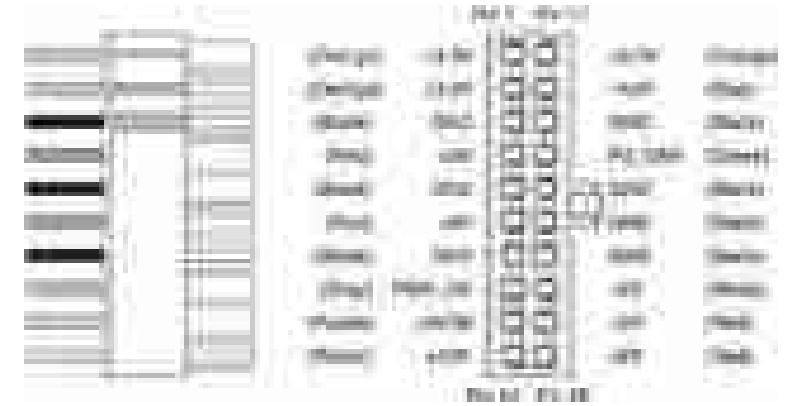
- Servers based on the ATX or microATX form factors generally use an ATX power supply
- Pedestal servers based on one of the Server System Infrastructure (SSI) form factors generally use an EPS12V power supply



# 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



# 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





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

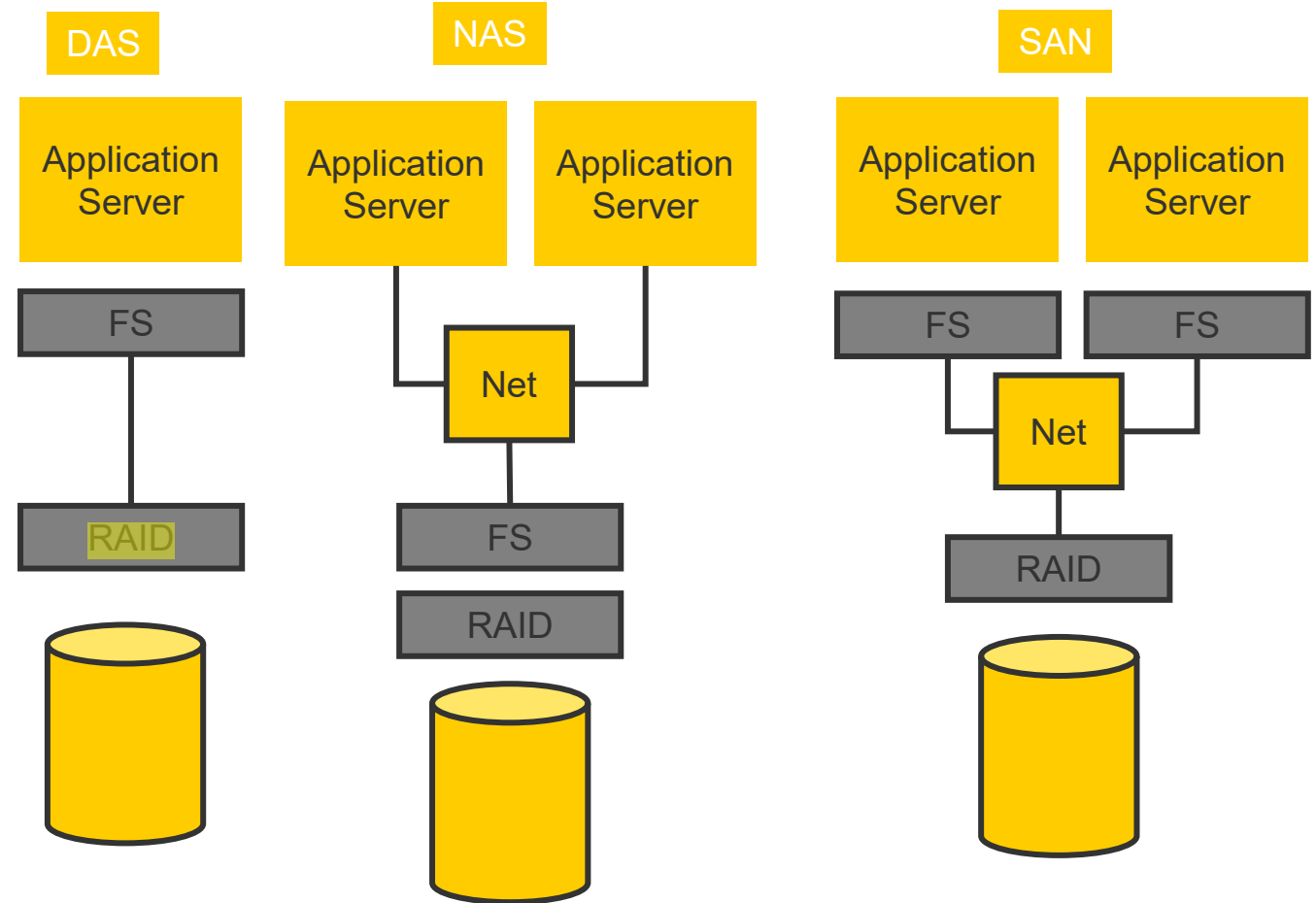
Arif Husen

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

# Storage Types

Four Types of Storage Solutions are :

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



## ➤ Direct-attached storage

### ▪ HDD

- Applications : Suitable for Write Heavy Applications, Backups
- Advantages : Cheap, High Capacity, Unlimited number of writes , Low Cost
- Disadvantages : Frigile, 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

## ➤ Direct-attached storage

### ▪ HDD

- Applications : Suitable for Write Heavy Applications, Backups
- Advantages : Cheap, High Capacity, Unlimited number of writes , Low Cost
- Disadvantages : Frigile, 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

## ➤ 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

## ➤ 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

## ➤ 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



## ➤ 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 (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

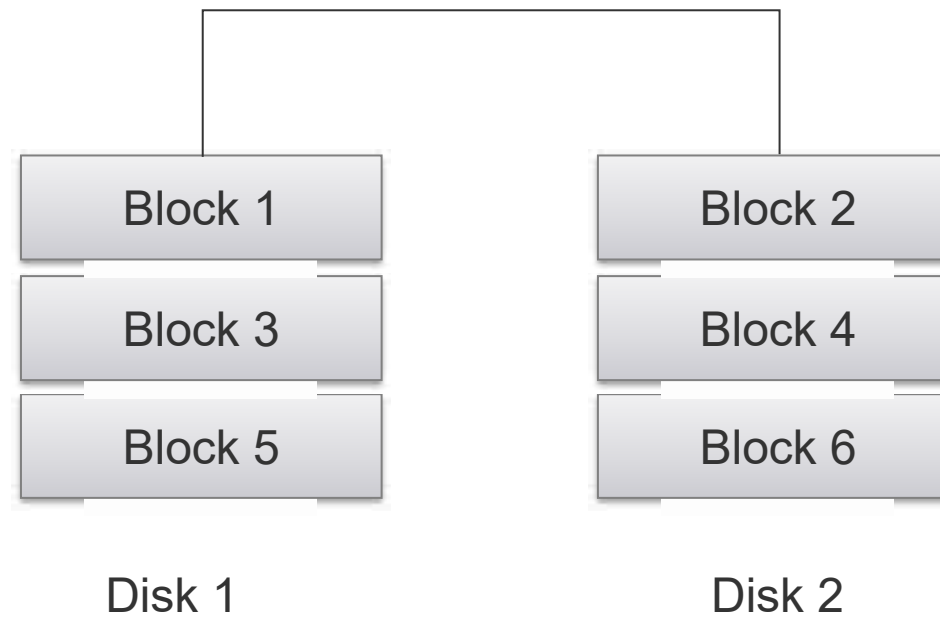
RAID 6

RAID 10 (RAID 1 + 0)

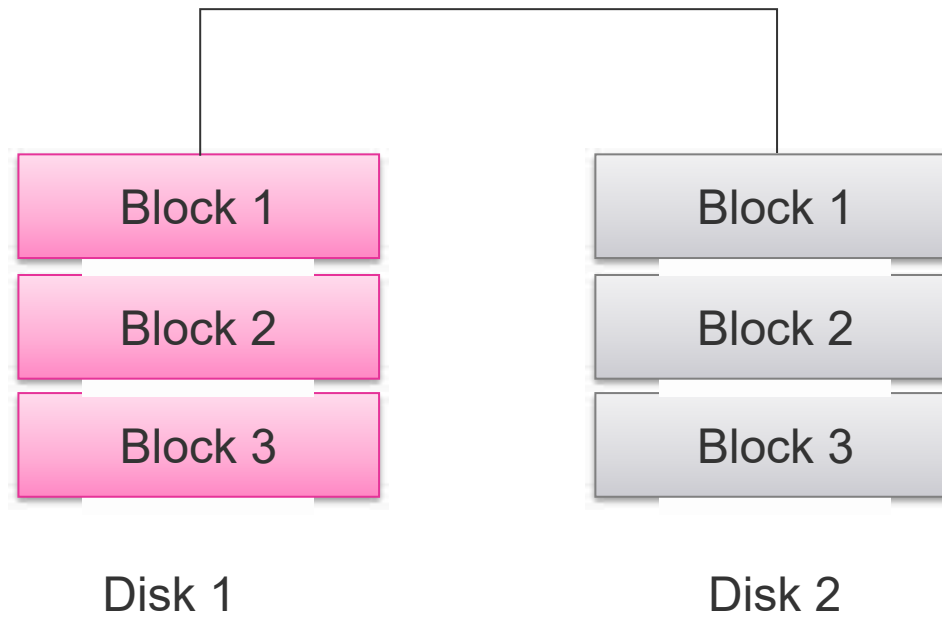
RAID 50 (RAID 5 + 0)

RAID 60 (RAID 6 + 0)

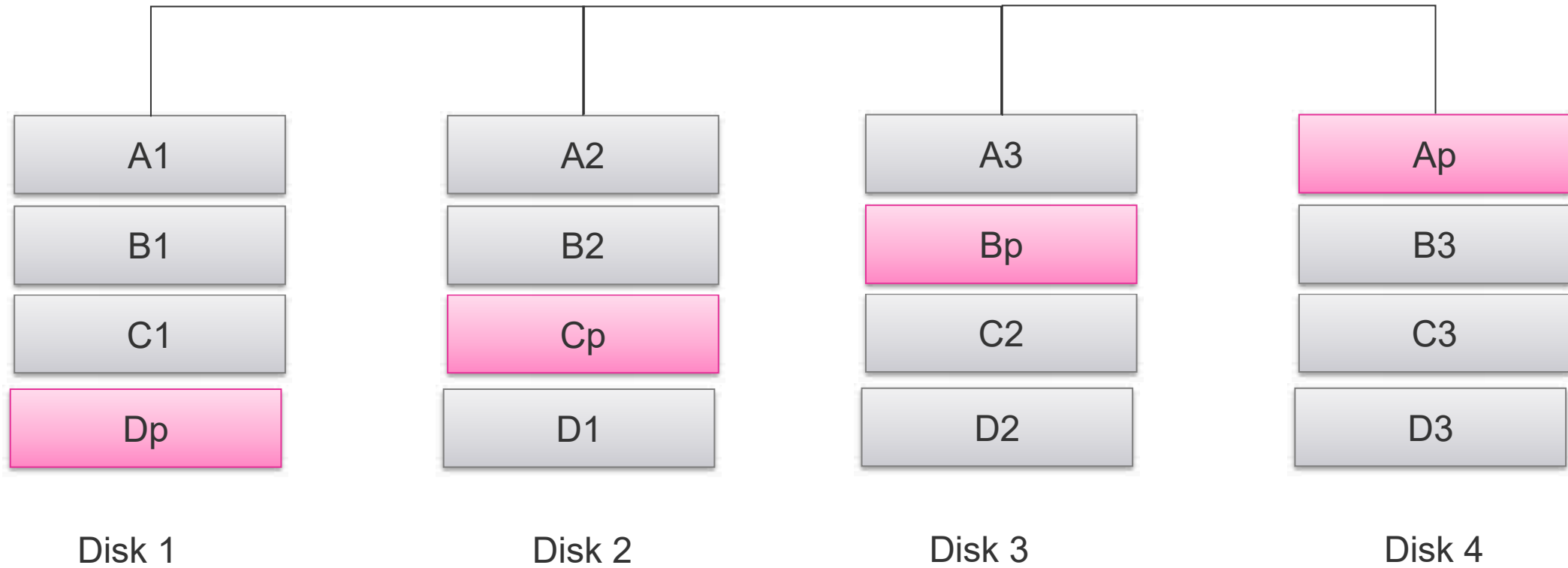
# RAID 0



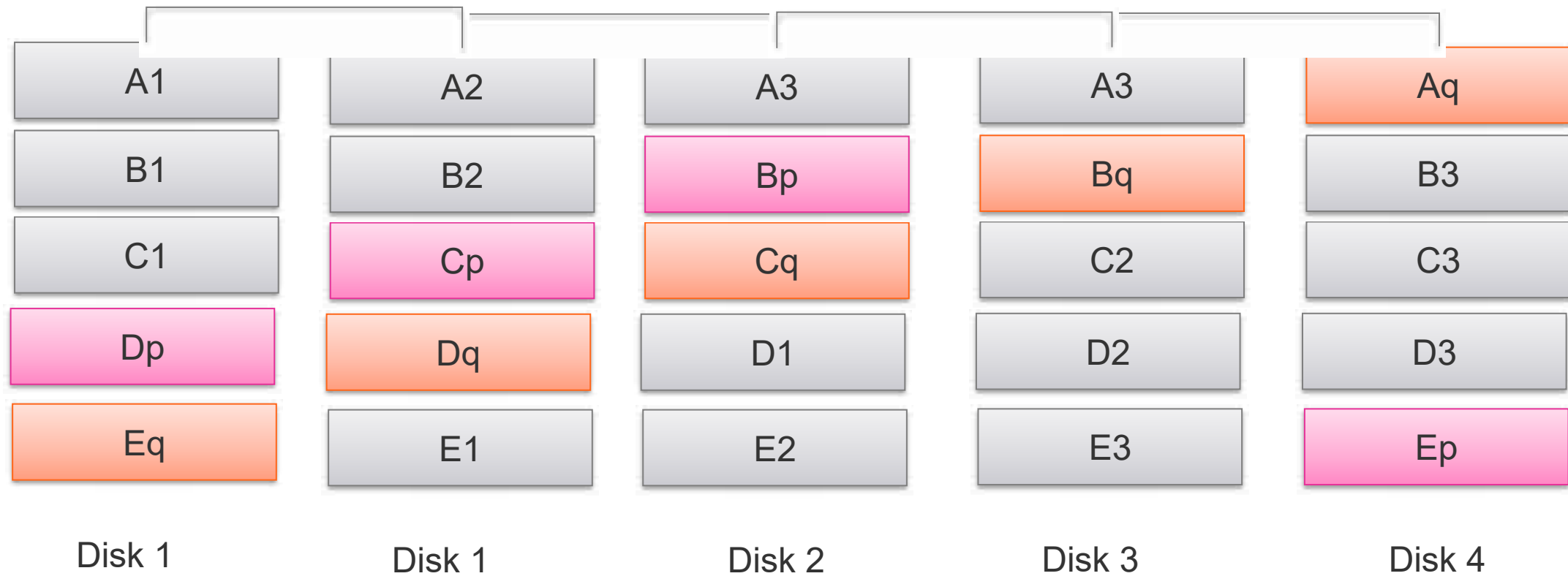
# RAID 1



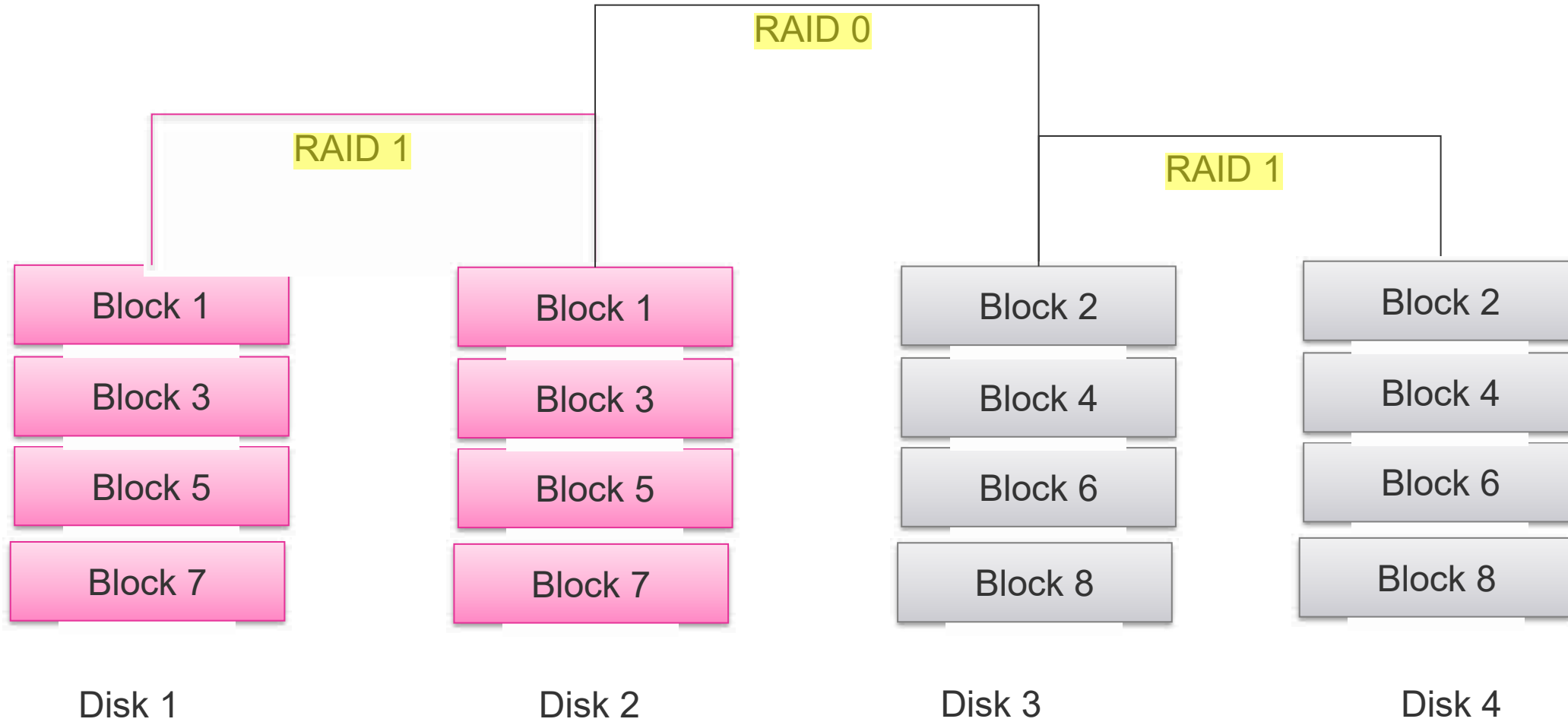
# RAID 5



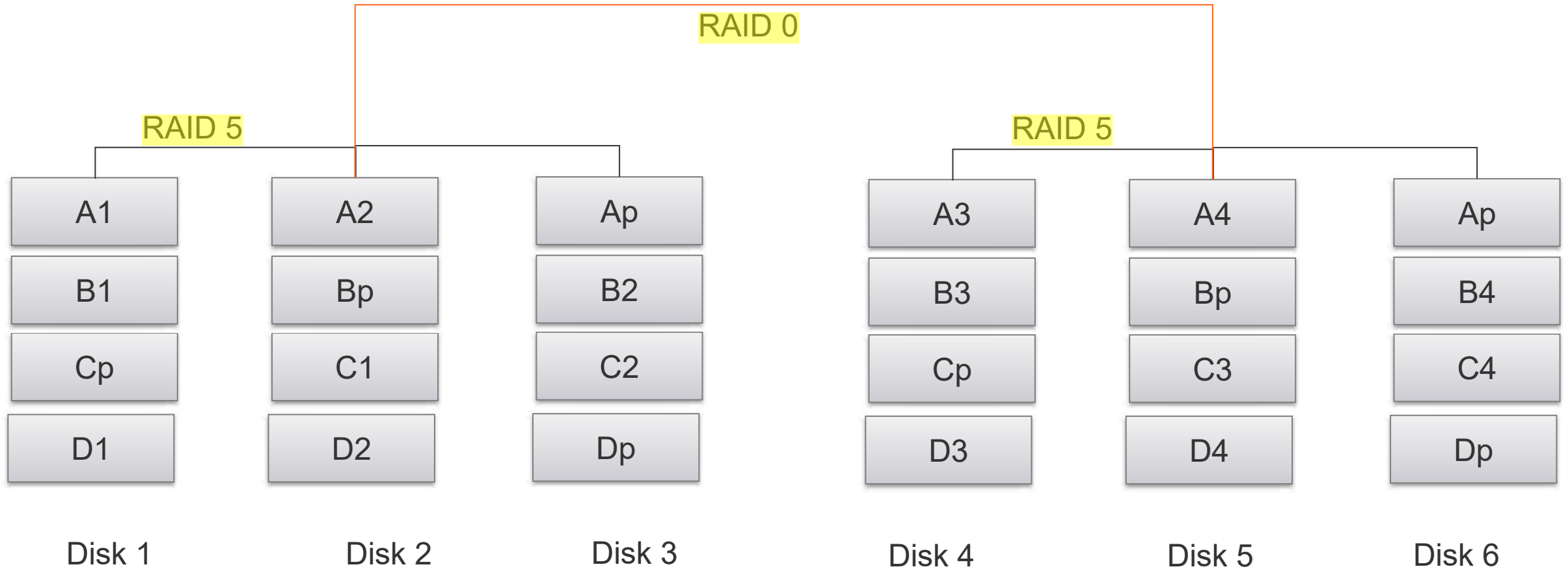
# RAID 6



# RAID 10

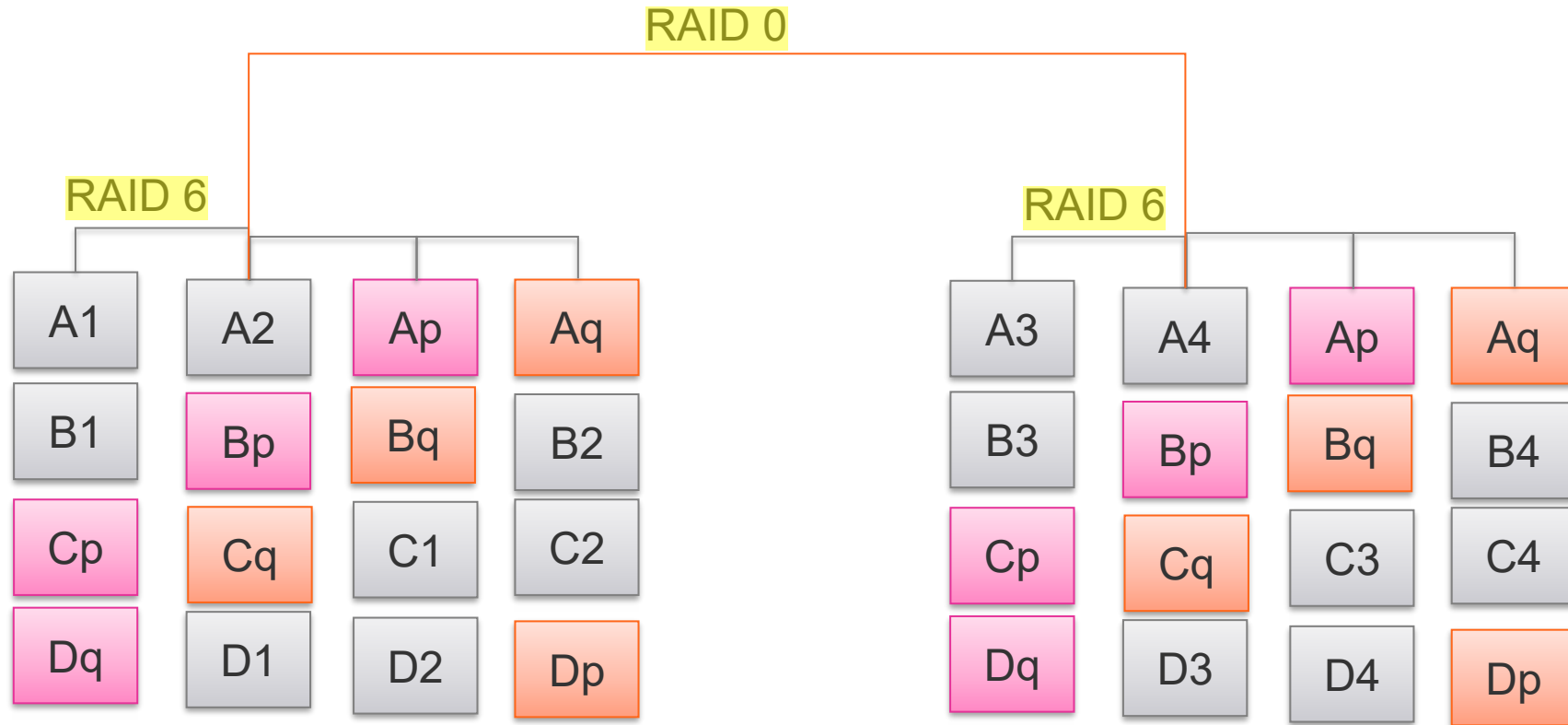


# RAID 50





# RAID 60



**IT601 – System and Network Administration**

**Week # 6**

# Server Racks

Arif Husen

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

**Virtual University of Pakistan**

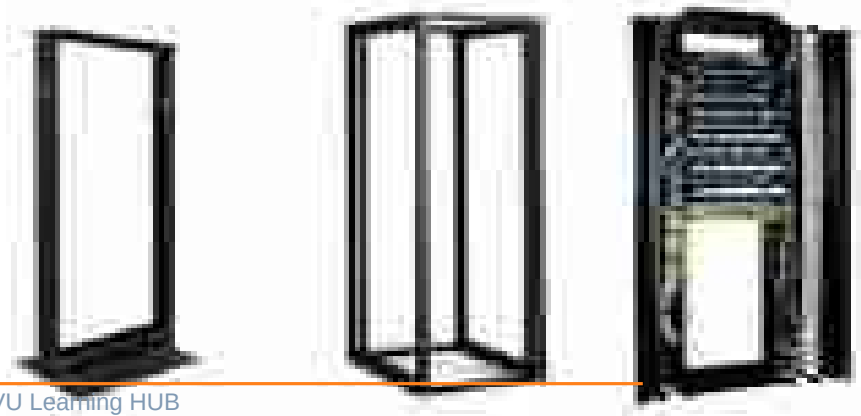
- Racks organize IT equipment into standardized assemblies that make efficient use of space and other resources
- At the most basic level, a rack consists of two or four vertical mounting rails and the supporting framework required to keep the rails in place.
- The rails and framework are typically made 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. 

## 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.



## ➤ 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

## 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**



- Extensibility
- More CPU performance
- High-performance I/O Rack 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**

## 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

- 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 I meets the requirements.
  - The downside is that the result is a fleet of unique machines.
  - Each is a beautiful, special little snowflake



- 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.



- **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.

- 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.

- 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

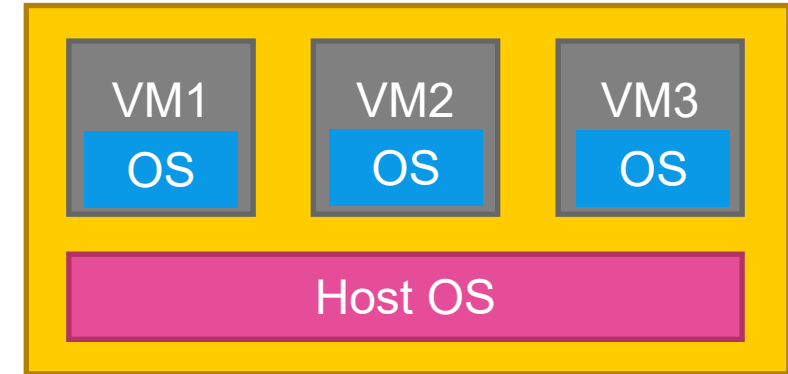
➤ **The next strategy is to buy computing resources in bulk and allocate fractions of it as needed**

➤ **One way to do this is through virtualization.**

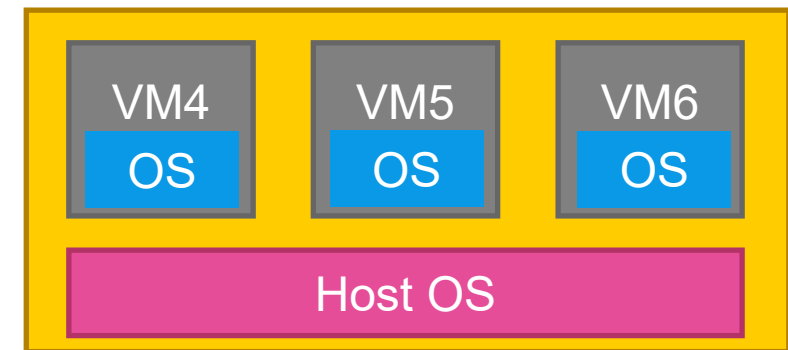
- That is, an organization purchases large physical servers and divides them up for use by customers by creating individual virtual machines (VMs).
- A virtualization cluster can grow by adding more physical hardware as more capacity is needed.

➤ **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 1



Physical Server 2

## ➤ 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.

- 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.

## ➤ 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.



- 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

- 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.

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.

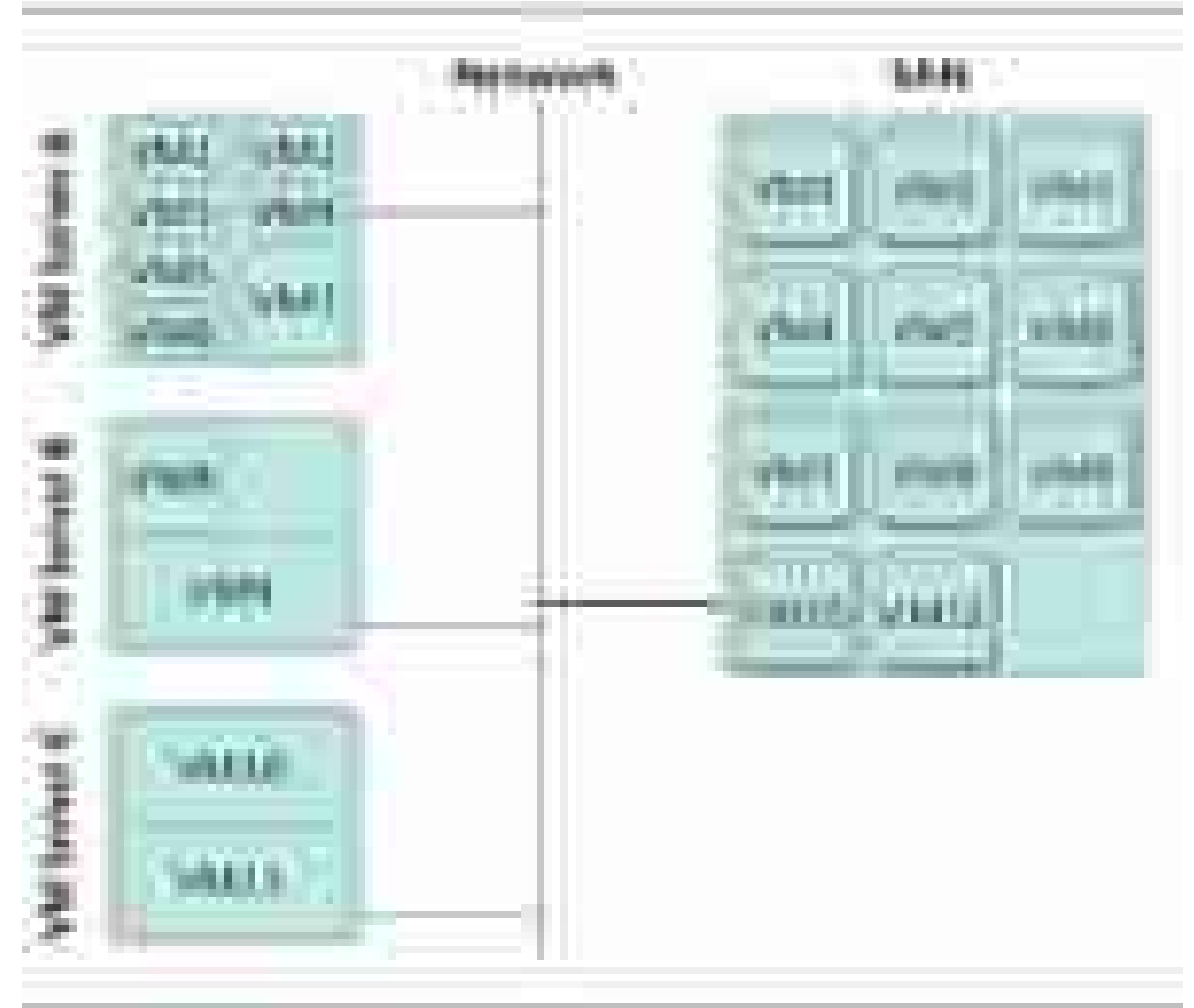
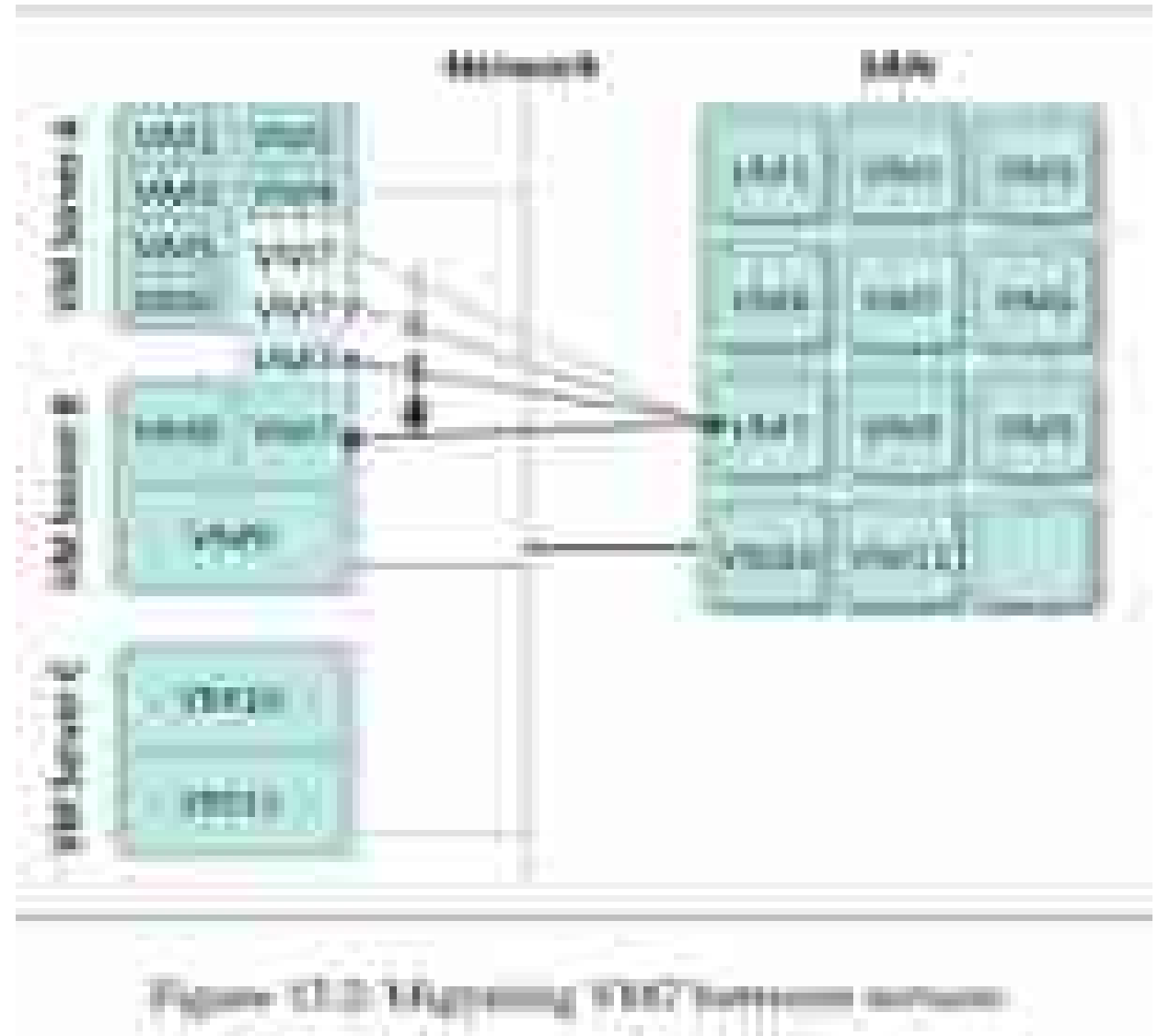


Figure 12.1 VM servers with shared storage

# Buy in bulk, allocate fractions – Live Migration



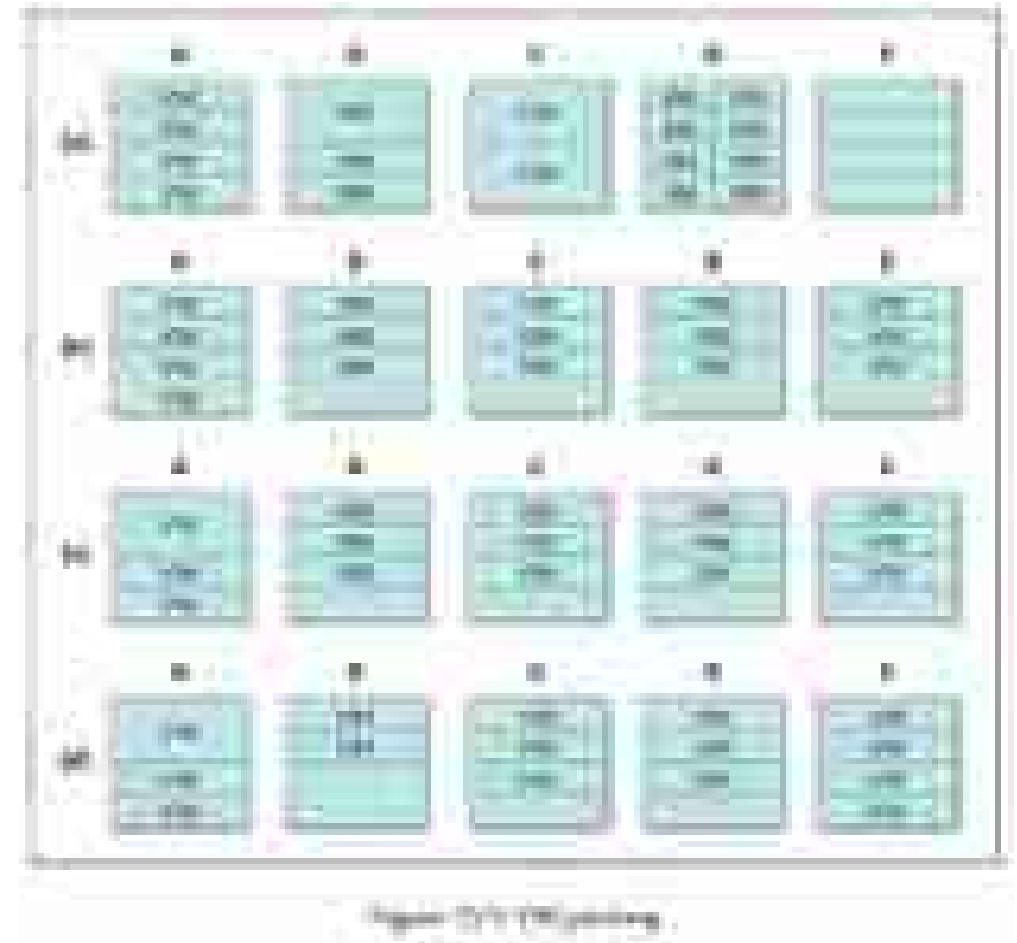
**IT601 – System and Network Administration**

**Buy in bulk, allocate fractions  
Other Aspects**

Arif Husen

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

- 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



- 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



- 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 from the 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 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**

**Week # 7**

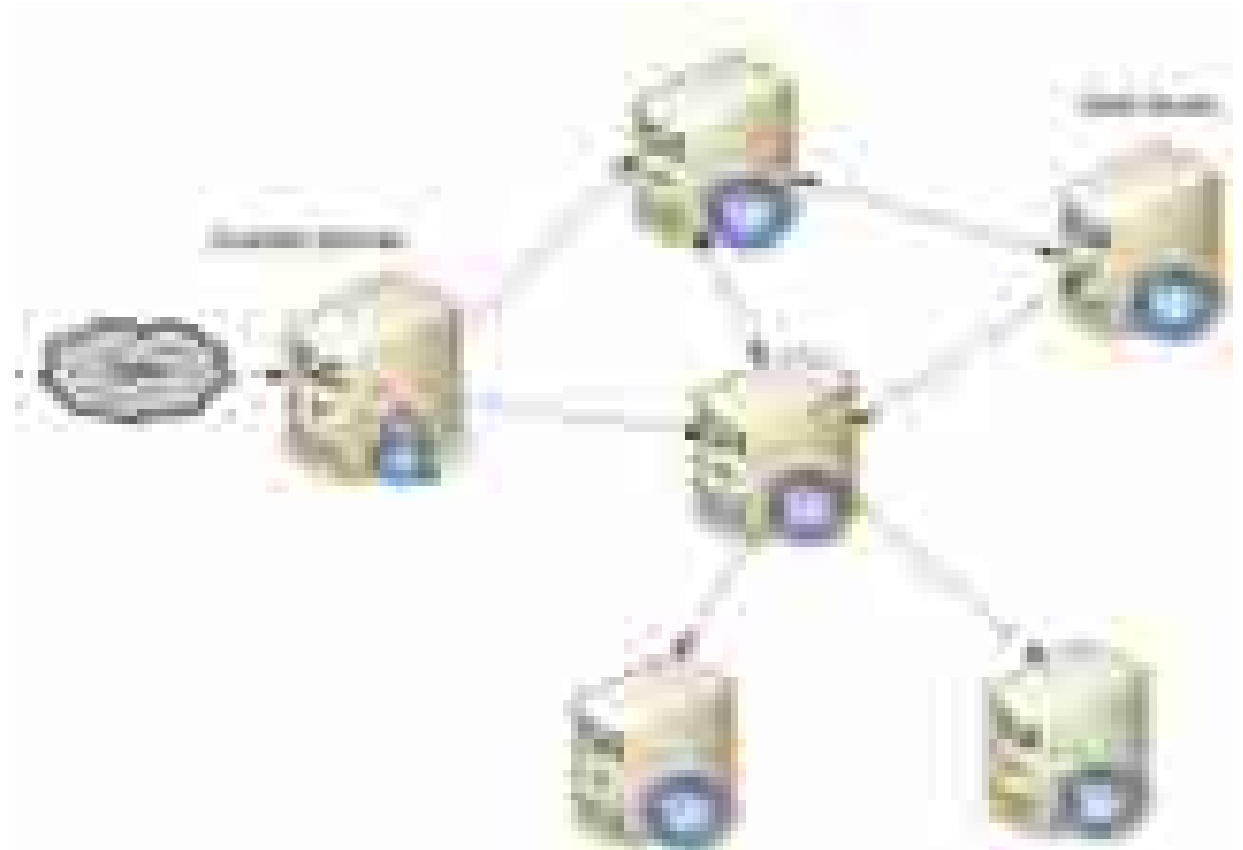
# **Grid Computing Approach**

Arif Husen

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

**Virtual University of Pakistan**

- 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 through the grid management and scheduling system



- 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**

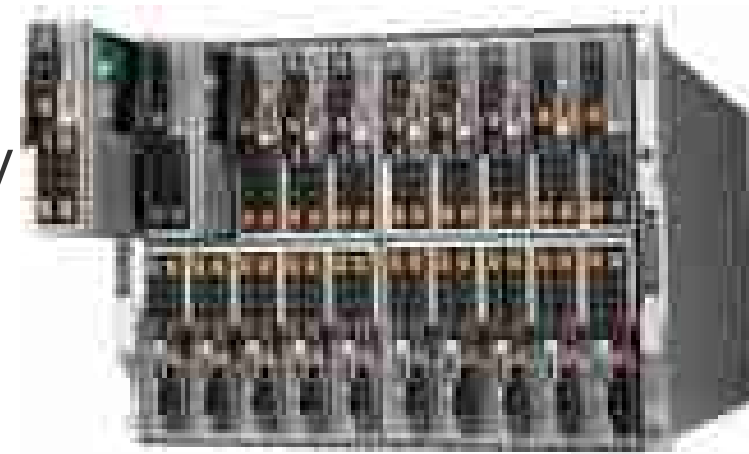
# Blade Server Approach

Arif Husen

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

**Virtual University of Pakistan**

- 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**

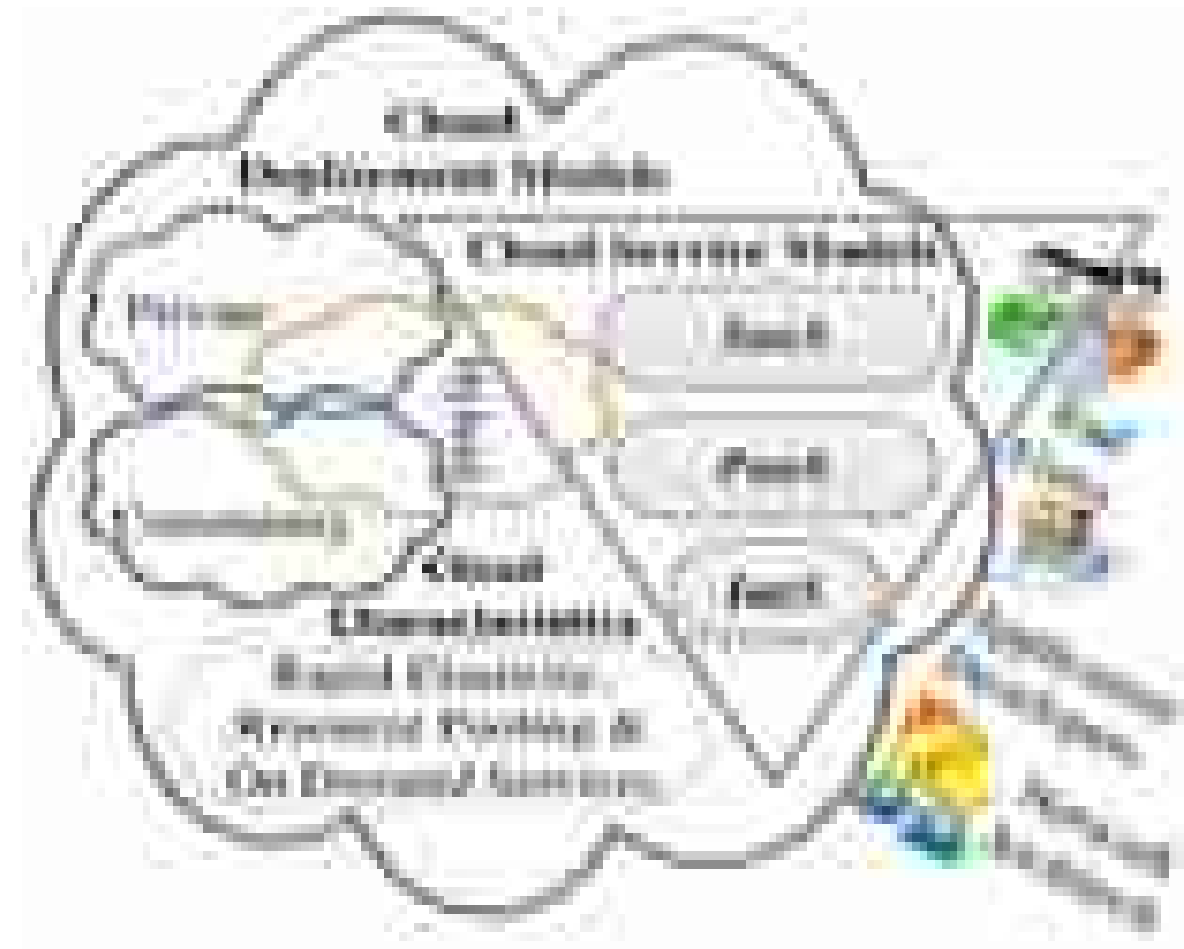
# Cloud Computing Approach

Arif Husen

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



- 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



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

- 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 pay-as-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.

- 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-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) 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.



- 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 web browsing 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.



- 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.

Integrator

Focus on Business

Reduced Complexity

Simplified Operations

Pay as you go

# 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**

➤ 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.



## ➤ 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.

Reduces resources

Time to Service

Less Expertise

Specialized Features

Performance

Reliability

# IT601 – System and Network Administration

<End>

Arif Husen

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

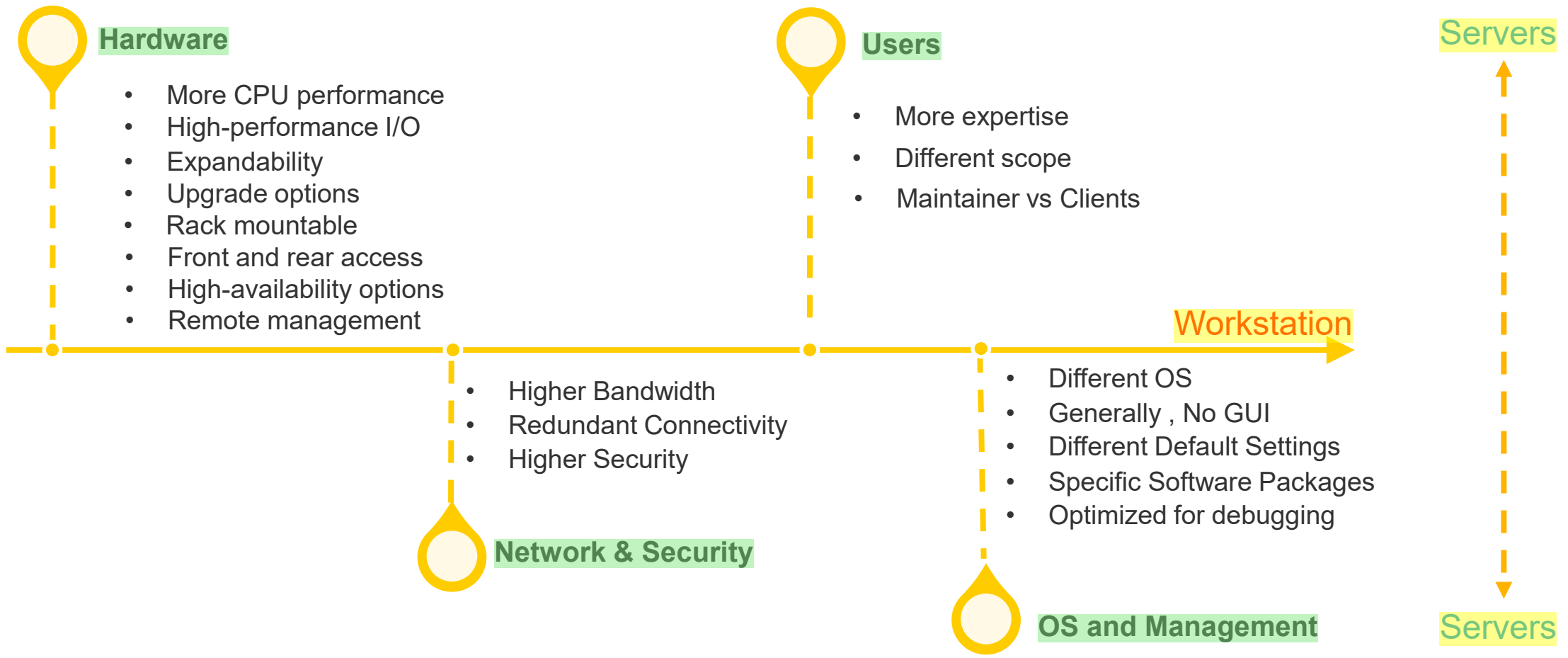


**IT601 – System and Network Administration**

# Server Selection Aspects

Arif Husen

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



## ▪ Levels of Redundancy

- $N+0$ ,  $N+1$ , ...,  $N+r$
- Identify Redundant Units, Power Supplies, CPUs, Hard disks etc

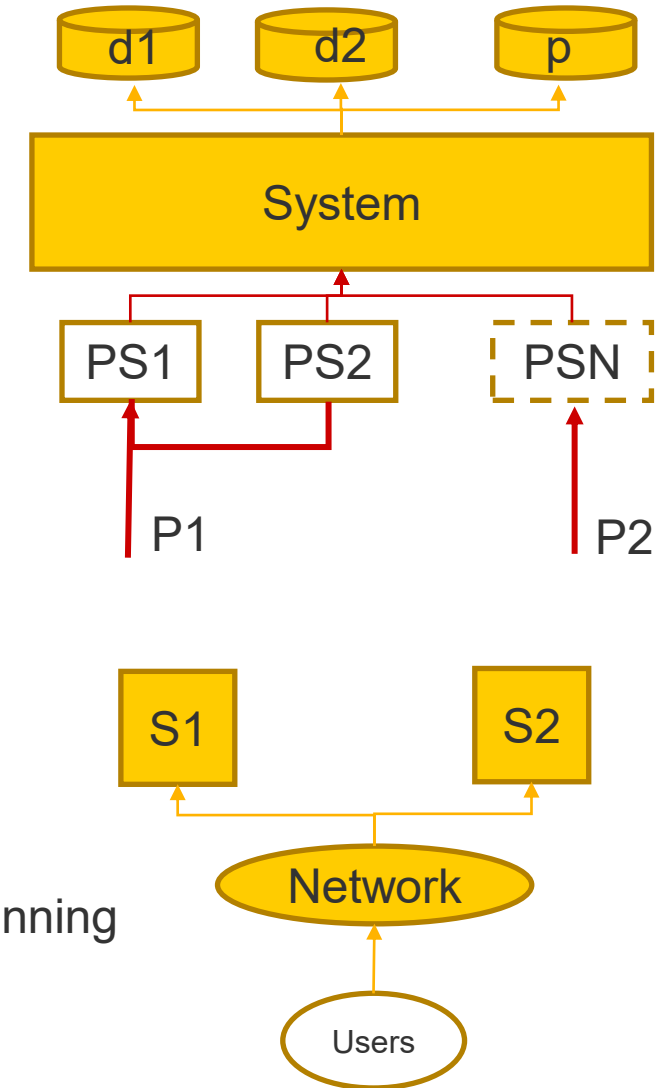
## ▪ Data Integrity

- ❑ 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 swappable

## ▪ Servers Should Be in Computer Rooms



- 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.

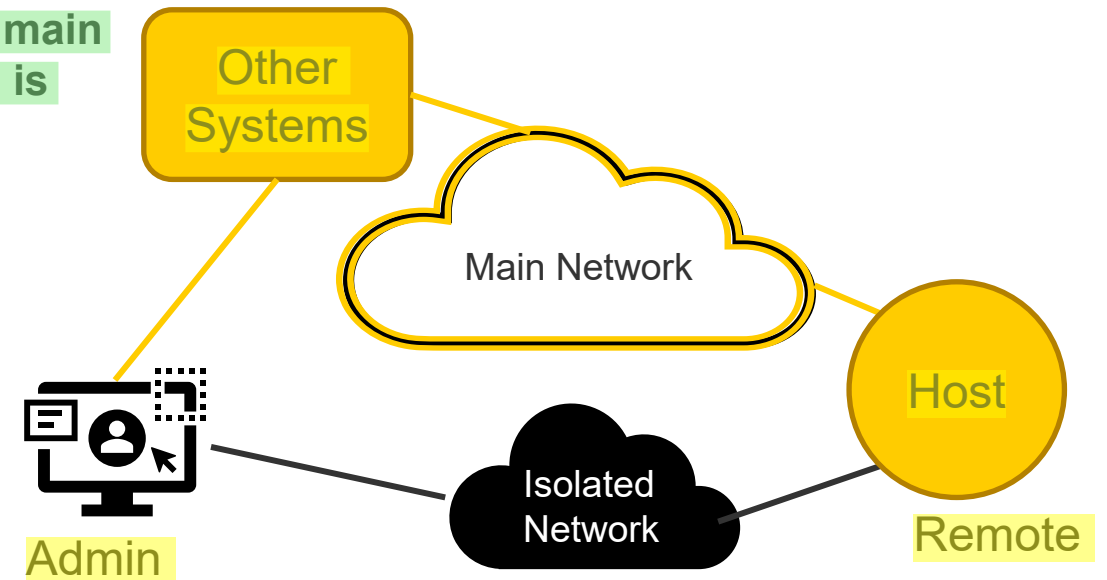
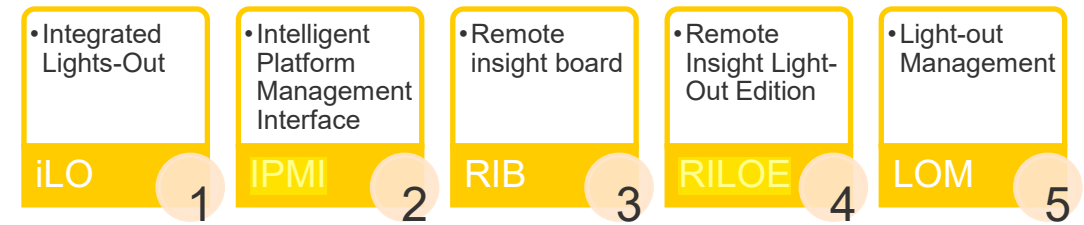
## ➤ OOB interface requires an IP Address, and remote system console is accessed via browser or client.

- 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

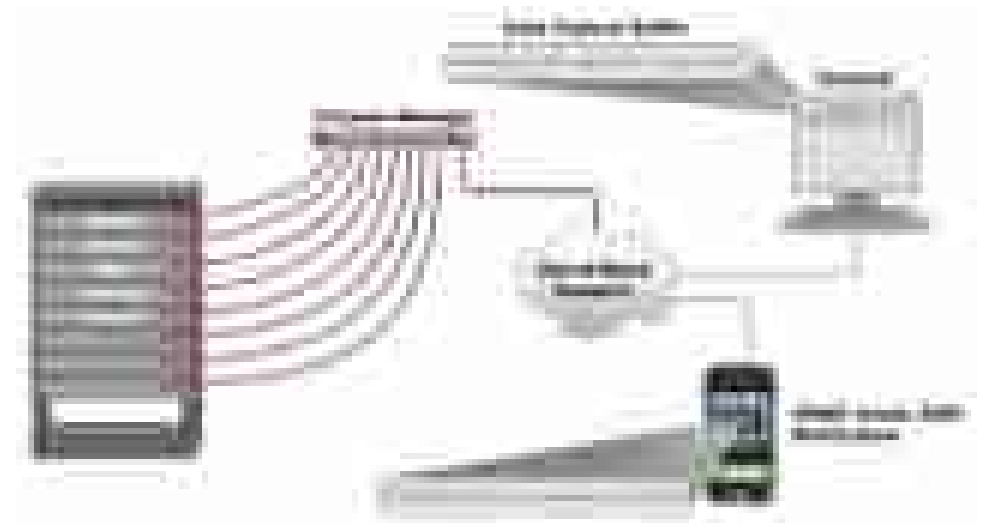
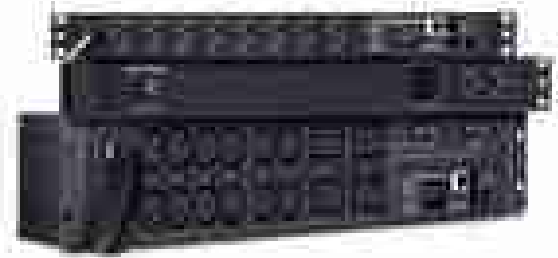
- Switched power distribution unit (PDU) provide the remote power cycling. A PDU is the fancy name for a power strip

## ▪ 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                                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>• Should be more stable</li><li>• More static</li><li>• Service Networks are more dynamic</li></ul> | <ul style="list-style-type: none"><li>• More Restrictive firewall rules</li><li>• Serves more critical systems</li></ul> | <ul style="list-style-type: none"><li>• Only for administrative purpose</li></ul> | <ul style="list-style-type: none"><li>• Should be separated from other service network</li></ul> | <ul style="list-style-type: none"><li>• Simple Engineering</li><li>• No VLANs</li><li>• Simple Topology</li></ul> | <ul style="list-style-type: none"><li>• Should be always available</li><li>• Immune to outages</li></ul> |

## ➤ 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



## ➤ 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

## ➤ 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
- ☐ Track expiry of the initial maintenance contract that came with purchase
  - ☐ Remove dismantled machines
  - ☐ Add new machines

## ➤ 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.

- 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 authenticational 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**

**Week # 8**

**Services**

Arif Husen

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

**Virtual University of Pakistan**

- **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.



- **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

- Create the platform that other services rely on.
- DNS, DHCP, directory services, network access (WAN and LAN), and Internet gateways
- While they are generally invisible to users, failures are highly visible because they affect many services.

## Basic services

- User-visible applications that most people have come to expect in an organization.
- Examples include printing, email, file storage, chat/IM, and VoIP/phone service.
- Because of their visibility and pervasiveness, people have developed expectations that these services will be reliable and always available.

## Primary Applications

- Applications generally drive business functions
- Examples are payroll, inventory management, enterprise resource planning (ERP), supply chain management, and so on

## Back Office Systems

- include the databases and other behind-the-scenes
- services that support applications

- **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**

➤ **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.

## 2 - Gathering Written Requirements

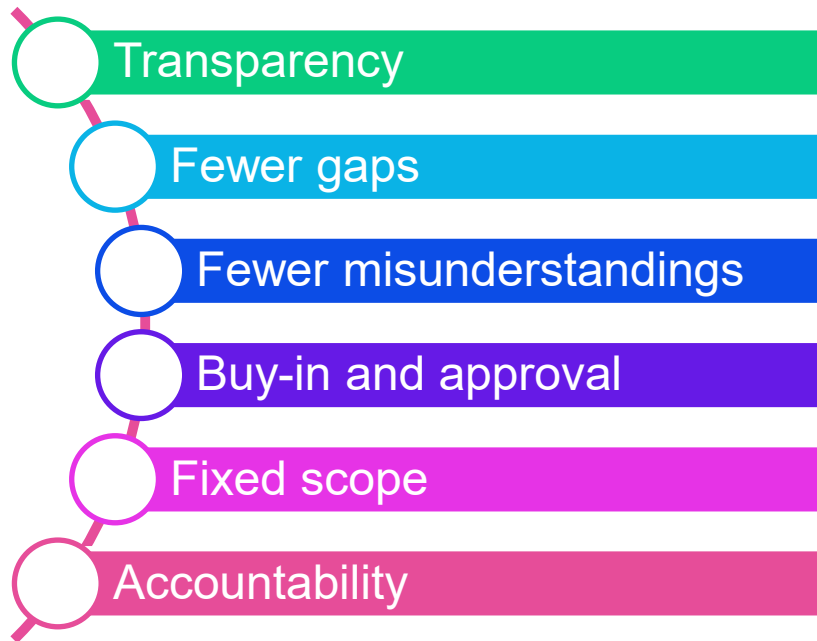
- The next step is gathering requirements.
- **Goal of Requirements**
  - Requirements are a list of what the service will be able to do. Requirements should list desired functionality, features, and capabilities. Focus on the end goal: what the system will enable people to achieve.
- **Serve as checklist 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

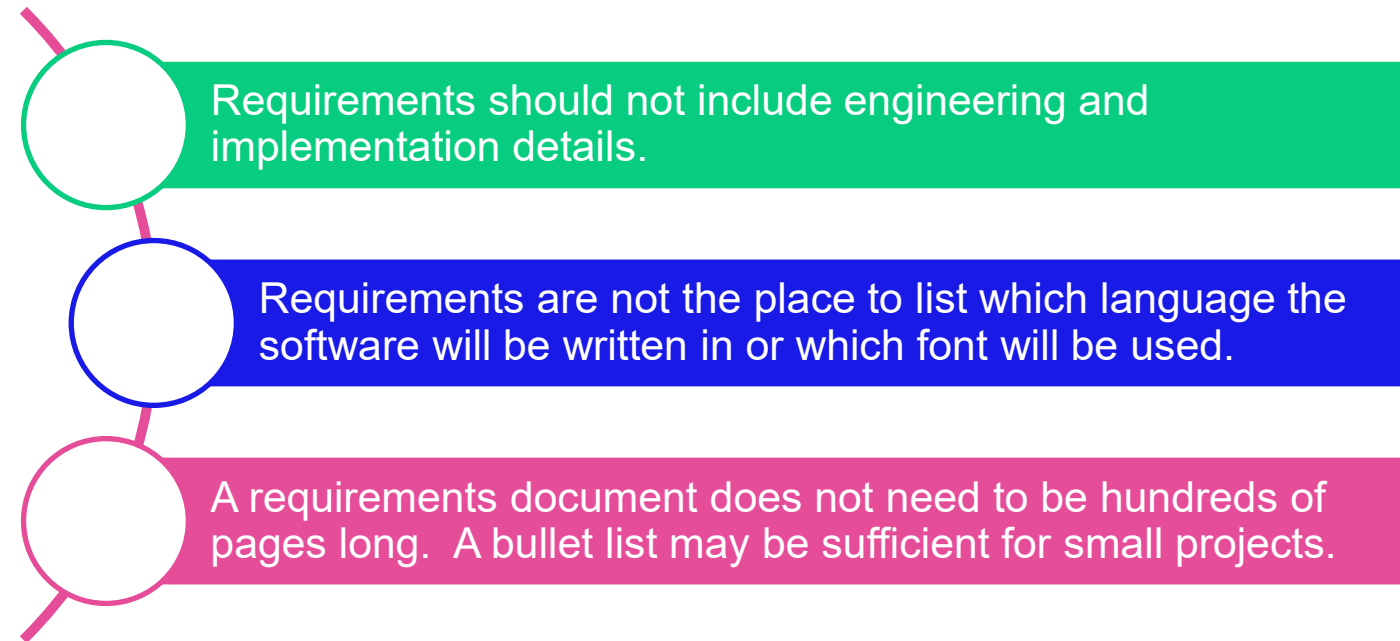
- Requirements are written down. They are not simply agreed to in a verbal discussion, tracked on a dry-erase board in your office, or kept in your head.

- 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.

- 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**



## ➤ 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

- 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

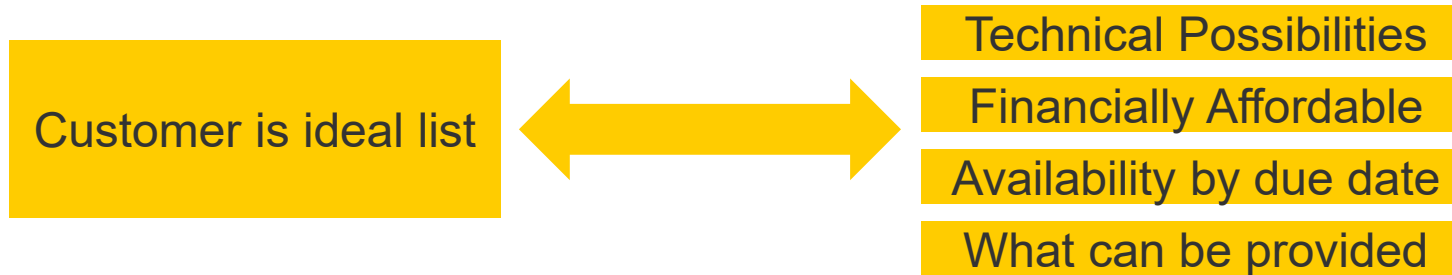
- The requirements document should include the service level agreement (SLA), or at least some general specifications that can be used to build an SLA.
  - What SLA Should Includes.

| 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.

- **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

- **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.



- **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.

- 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.

- **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.

## ➤ 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.

- **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
- Backup/restore facilities
- Monitoring system
- Network equipment vendor and routing protocols
- Security auditing systems

- Trouble-ticket systems
- Dashboards and consoles
- Network directory services
- DNS and other name services
- Printing systems

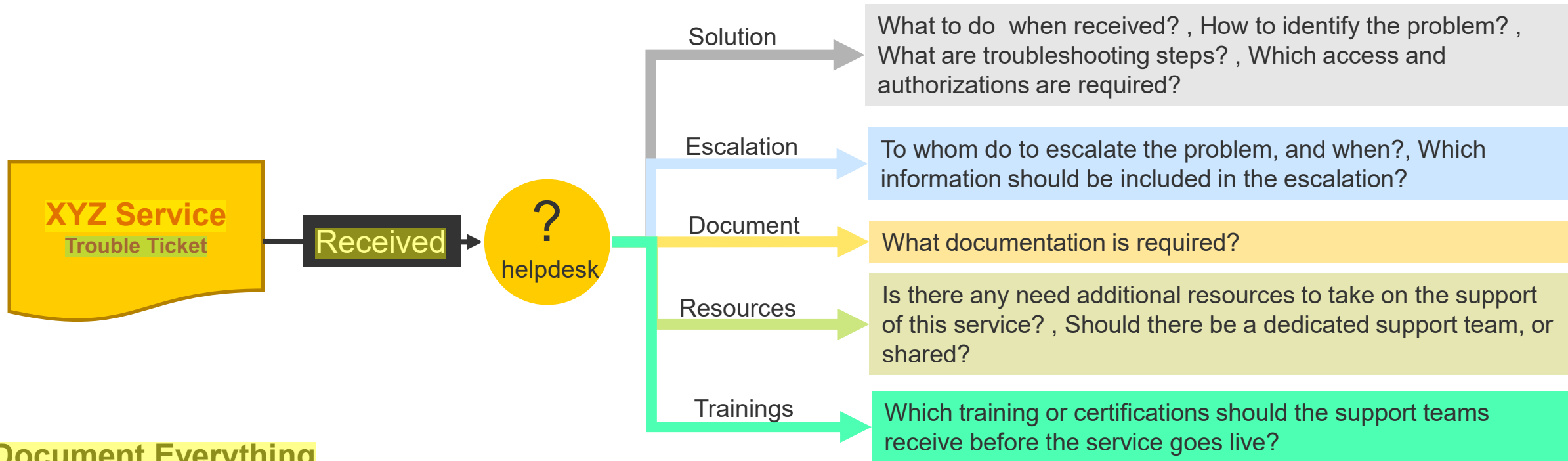
# F - Support Model

➤ **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.~~

➤ **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.

- **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**

**Week # 9 (Part 1)**

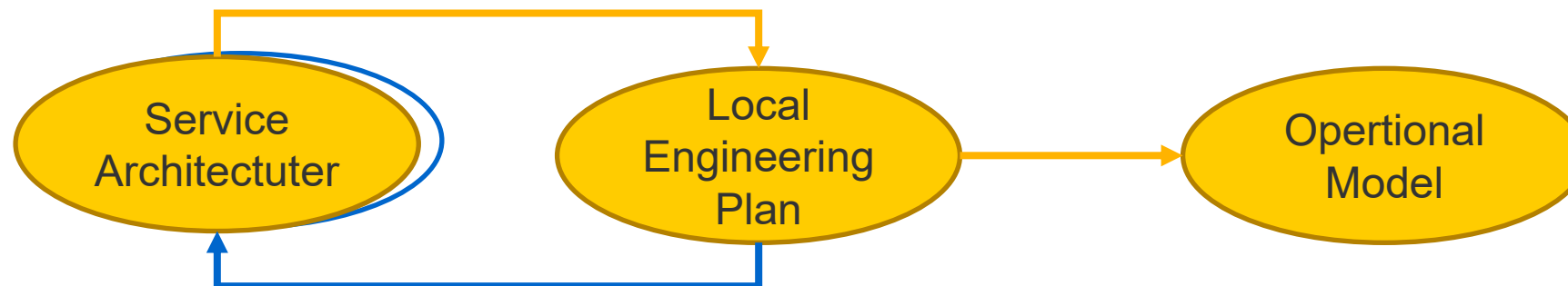
# **Service Planning and Design**

Arif Husen

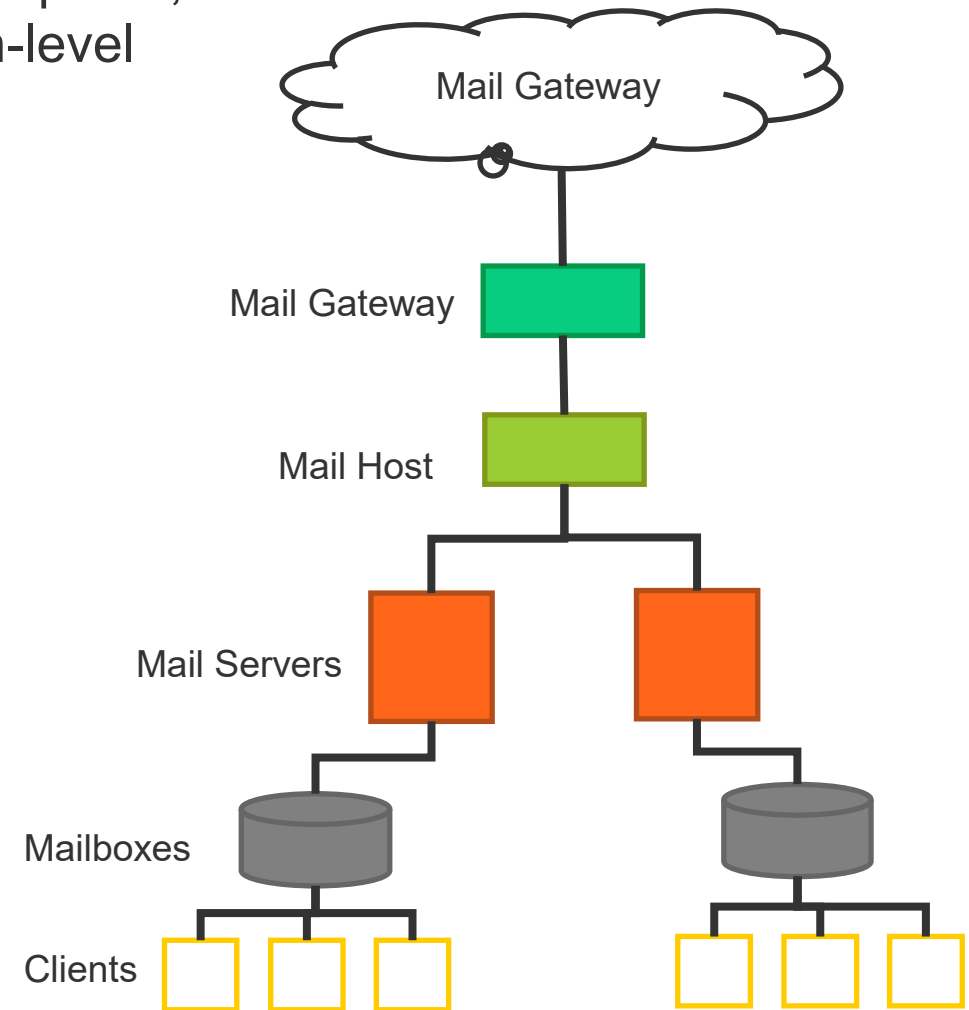
**Department of Computer Science and Information Technology,**

**Virtual University of Pakistan**

- 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



- 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 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 Hierarchy

Operational Plan is also 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





Simplicity



Vendor-Certified Designs



Dependency Engineering



Decoupling Hostname from Service Name



Support

- 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

- **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.

- **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**

- 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

- Make Redundant domains including power strips, CPUs, and storage and VPN devices

## Service Failure Domains

- Align all dependencies on same machine
- All startup scripts should be on the same machine
- Rack aligned dependencies

## Location Failure Domains

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

- 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



- 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

- 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

- 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.

- **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

Week 9 (Part 2)

Primary Services

Arif Husen

Department of Computer Science and Information Technology,

Virtual University of Pakistan

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

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

