



Virtual University of Pakistan
FEDERAL GOVERNMENT UNIVERSITY

COMPLETE SLIDES

Information Technology Infrastructure

IT602

Order Your VU Books Now

Get all Virtual University books in high-quality hard copy,
delivered right to your doorstep. Visit vubookshoppk.com

 **Order Now**

Information Technology Infrastructure

Information Technology Infrastructure

- **Credit Hours:** 3
- **Lecturer:** Shafaq Nisar
- **Lecturing Style:** Video Lectures of short duration (5-7 minutes)

The definition of IT infrastructure

1. Introduction
2. IT Infrastructure
3. The Infrastructure Model
4. Non-functional Attributes
5. Non-functional Requirements

Topic 1

Introduction of IT Infrastructure

What is IT Infratsructure?

Infrastructure



IT Infrastructure



Change is inevitable because

Rapid
development of
new types of
applications

Big data

The Internet of
Thing

Mobility

Cloud
computing

Various Kinds of Architecture

- ❖ Business architecture
- ❖ Enterprise architecture
- ❖ Data architecture,
- ❖ Application architecture and
- ❖ Infrastructure architecture

Is there a General Definition of IT Infrastructure?

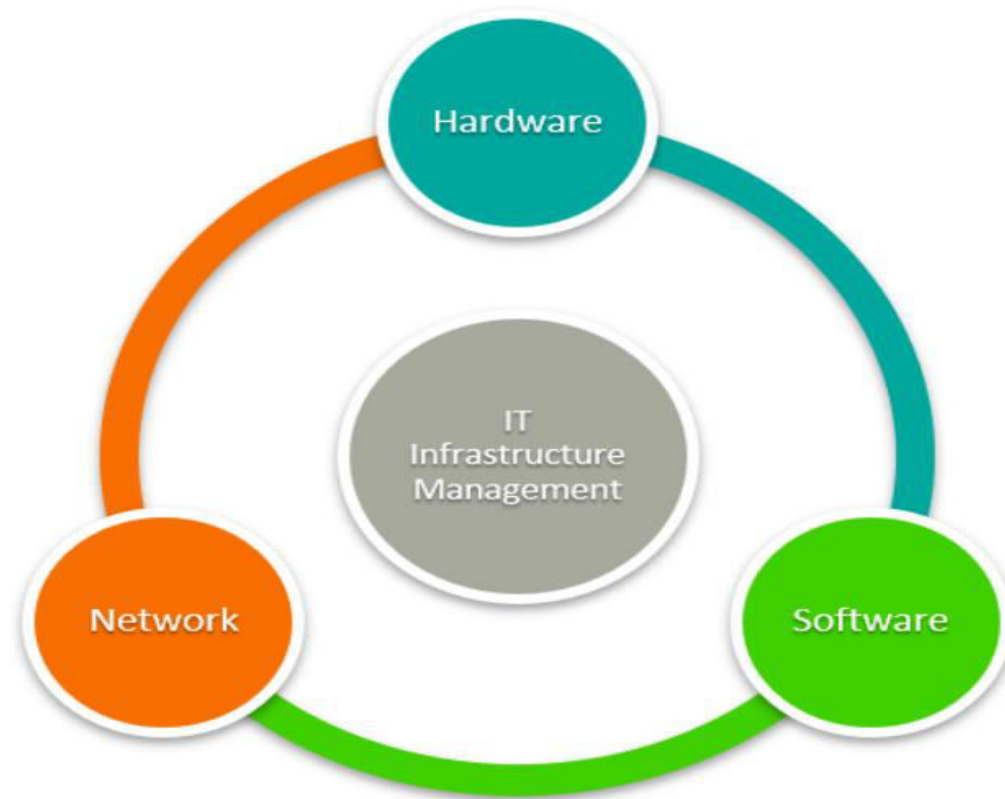
No generally accepted definition of IT infrastructure seems to exist

In literature, many definitions of IT infrastructure are described. Some of them are:

IT Infrastructure

IT infrastructure consists of the equipment, systems, software, and services used in common across an organization, regardless of mission/program/project. IT Infrastructure also serves as the foundation upon which mission/program/project-specific systems and capabilities are built.

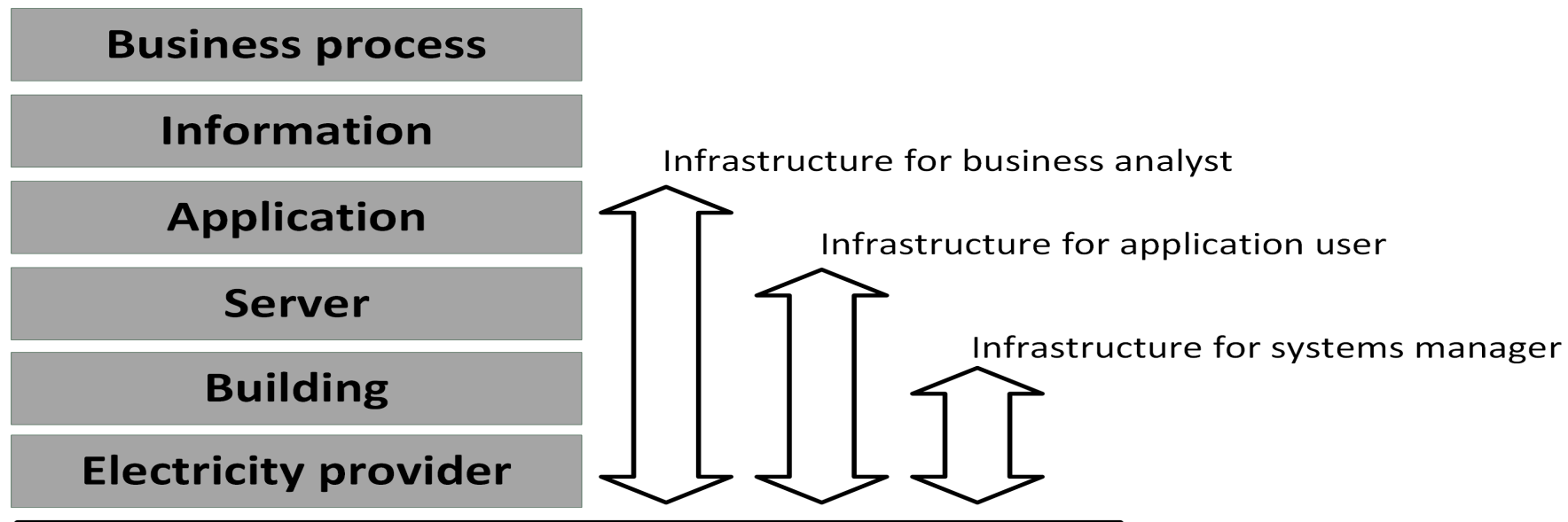
Traditional IT Infrastructure



The definition of IT Infrastructure

What infrastructure comprises depends on:

- Who you ask
- What their point of view is



For most people, infrastructure is invisible and taken for granted

Information Technology Infrastructure

Information Technology Infrastructure

- **Credit Hours:** 3
- **Lecturer:** Shafaq Nisar
- **Lecturing Style:** Video Lectures of short duration (5-7 minutes)

Introduction to Non-Functional Attributes

- IT infrastructure provides services to applications
- Many of these services can be defined as functions such as
 - Disk space,
 - Processing,
 - Connectivity

However most of these services are non functional in nature

Non-Functional Attributes

Non functional attributes describe the qualitative behavior of the system rather than its specific functionality and these include

- Availability
- Security
- Performnace
- Recoverability
- Testtability
- Scalability

Handling Conflicting NFRs

It is unusual to encounter conflicting NFRs for instance users may want a system that is secure but not want to be bothered by passwords

- It is the task of the infrastructure architect to balance these NFRs, in some cases some NFRs may take priority over others and the architect must involve the relevant stakeholders

Information Technology Infrastructure

Information Technology Infrastructure

- **Credit Hours:** 3
- **Lecturer:** Shafaq Nisar
- **Lecturing Style:** Video Lectures of short duration (5-7 minutes)

AVAILABILITY CONCEPTS

1. Introduction
2. Calculation of Availability
3. Mean Time Between Failures (MTBF)
4. Sources of Unavailability
5. Types of errors
6. Environmental issues
7. Complexity of the infrastructure
8. Availability Patterns
9. Redundancy, Failover
10. Business Continuity

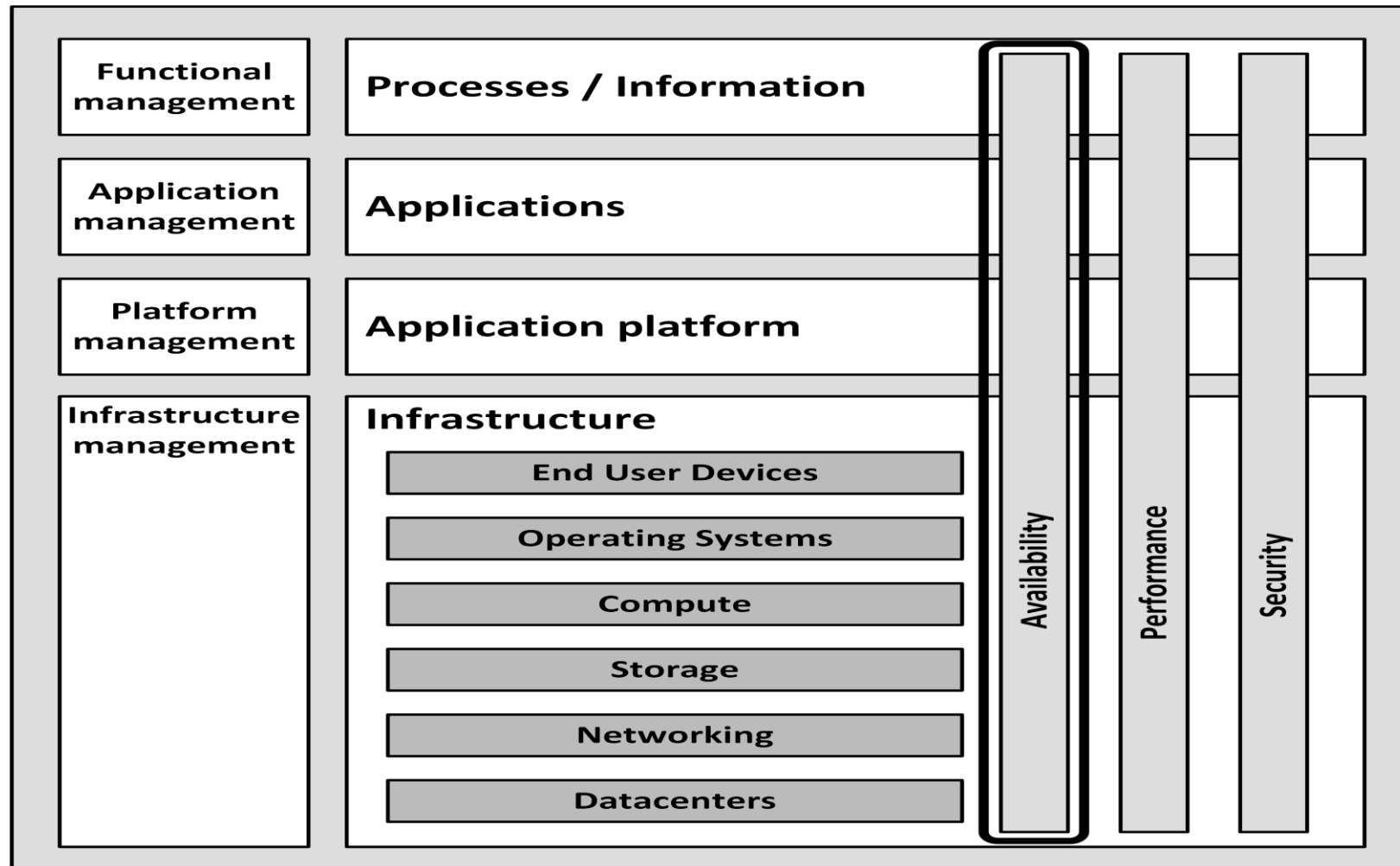
Introduction

AVAILABILITY CONCEPTS

Introduction

- Everyone expects their infrastructure to be available all the time
- A 100% guaranteed availability of an infrastructure is impossible
- **A fact of life**
 - There is always a chance of downtime.

Introduction



Calculation of Availability

- Availability can neither be calculated, nor guaranteed upfront
 - It can only be reported on afterwards, when a system has run for some years
- Over the years, much knowledge and experience is gained on how to design high available systems
 - Failover
 - Redundancy
 - Structured programming
 - Avoiding Single Points of Failures (SPOFs)
 - Implementing systems management

Calculation of Availability

- The availability of a system is usually expressed as a percentage of uptime in a given time period, usually one year or one month
- Example for downtime expressed as a percentage per year

Availability %	Downtime per year	Downtime per month	Downtime per week
99.8%	17.5 hours	86.2 minutes	20.2 minutes
99.9% ("three nines")	8.8 hours	43.2 minutes	10.1 minutes
99.99% ("four nines")	52.6 minutes	4.3 minutes	1.0 minutes
99.999% ("five nines")	5.3 minutes	25.9 seconds	6.1 seconds

Calculation of Availability

Typical requirements used in service level agreements today are 99.8% or 99.9% availability per month for a full IT system

- The availability of the infrastructure must be much higher
 - Typically in the range of 99.99% or higher

Calculation of Availability

- 99.999% uptime is also known as carrier grade availability
 - For one component, higher availability levels for a complete system are very uncommon, as they are almost impossible to reach

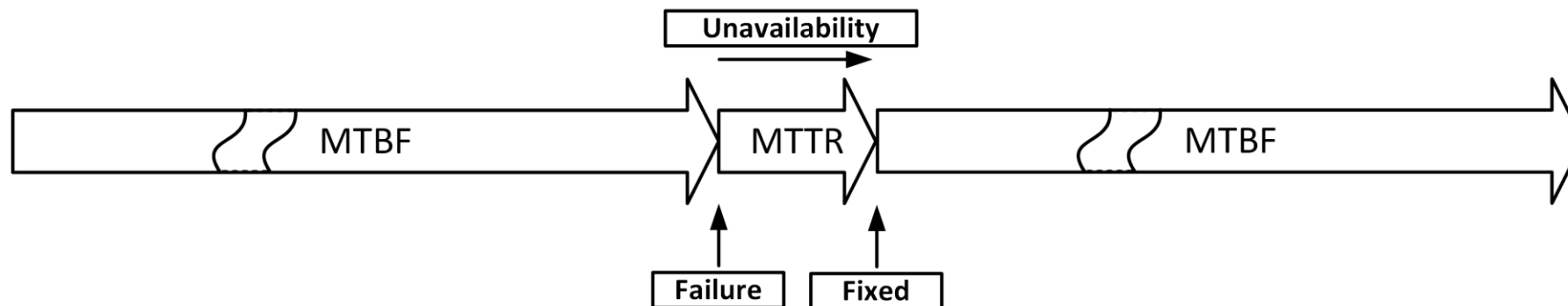
Calculation of Availability

- It is a good practice to agree on the maximum frequency of unavailability

Unavailability (minutes)	Number of events (per year)
0 - 5	≤ 35
5 - 10	≤ 10
10 - 20	≤ 5
20 - 30	≤ 2
> 30	≤ 1

MTBF and MTTR

- Mean Time Between Failures (MTBF)
 - The average time that passes between failures
- Mean Time To Repair (MTTR)
 - The time it takes to recover from a failure



MTBF and MTTR

- Some components have higher MTBF than others
- Some typical MTB's:

Component	MTBF (hours)
Hard disk	750,000
Power supply	100,000
Fan	100,000
Ethernet Network Switch	350,000
RAM	1,000,000

MTTR

- MTTR can be kept low by:
 - Having a service contract with the supplier
 - Having spare parts on-site
 - Automated redundancy and failover

MTTR

- Steps to complete repairs:
 - Notification of the fault (time before seeing an alarm message)
 - Processing the alarm
 - Finding the root cause of the error
 - Looking up repair information
 - Getting spare components from storage
 - Having technician come to the datacenter with the spare component
 - Physically repairing the fault
 - Restarting and testing the component

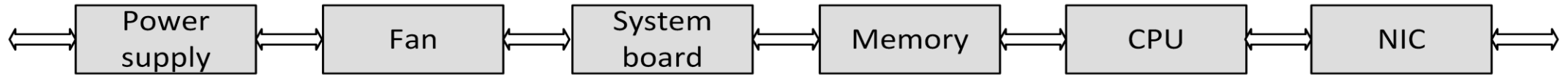
Calculation Examples

$$\text{Availability} = \frac{\text{MTBF}}{(\text{MTBF} + \text{MTTR})} \times 100\%$$

Component	MTBF (h)	MTTR (h)	Availability	in %
Power supply	100,000	8	0.9999200	99.99200
Fan	100,000	8	0.9999200	99.99200
System board	300,000	8	0.9999733	99.99733
Memory	1,000,000	8	0.9999920	99.99920
CPU	500,000	8	0.9999840	99.99840
Network Interface Controller (NIC)	250,000	8	0.9999680	99.99680

Calculation Examples

- Serial components: One defect leads to downtime



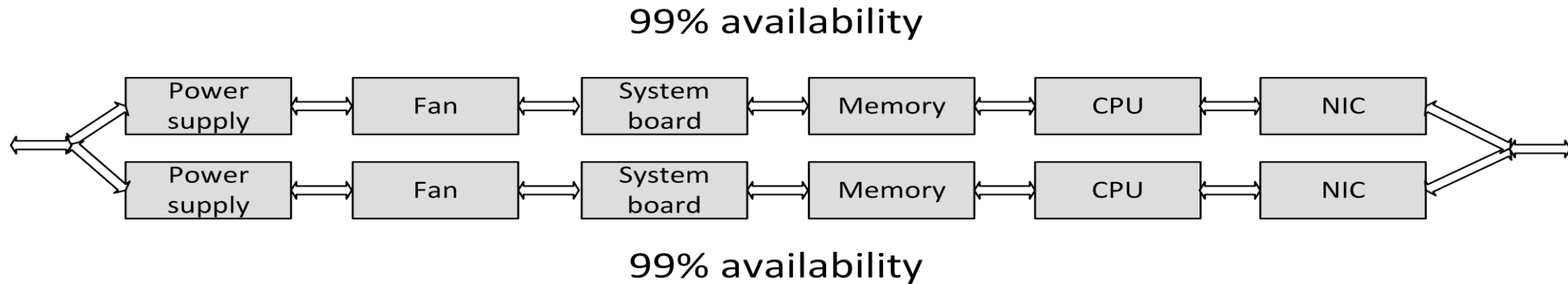
- Example: the above system's availability is:

$$0.9999200 \times 0.9999200 \times 0.9999733 \times 0.9999920 \\ \times 0.9999840 \times 0.9999680 = 0.99977 = \mathbf{99.977\%}$$

(each components' availability is at least 99.99%)

Calculation Examples

- Parallel components: One defect: no downtime!
- But beware of SPOFs!



- Calculate availability:

$$A = 1 - (1 - A_1)^n$$

- Total availability = $1 - (1 - 0.99)^2 = 99.99\%$

Sources of Unavailability - Human Errors

- 80% of outages impacting mission-critical services is caused by people and process issues
- Examples:
 - Performing a test in the production environment
 - Switching off the wrong component for repair
 - Swapping a good working disk in a RAID set instead of the defective one
 - Restoring the wrong backup tape to production
 - Accidentally removing files
 - Mail folders, configuration files
 - Accidentally removing database entries
 - Drop table x instead of drop table y

Sources of Unavailability - Software Bugs

- Because of the complexity of the software, it is nearly impossible (and very costly) to create bug-free software
- Application software bugs can stop an entire system
- Operating systems are software too
 - Operating systems containing bugs can lead to
 - corrupted file systems,
 - network failures, or
 - other sources of unavailability

Sources of Unavailability - Planned Maintenance

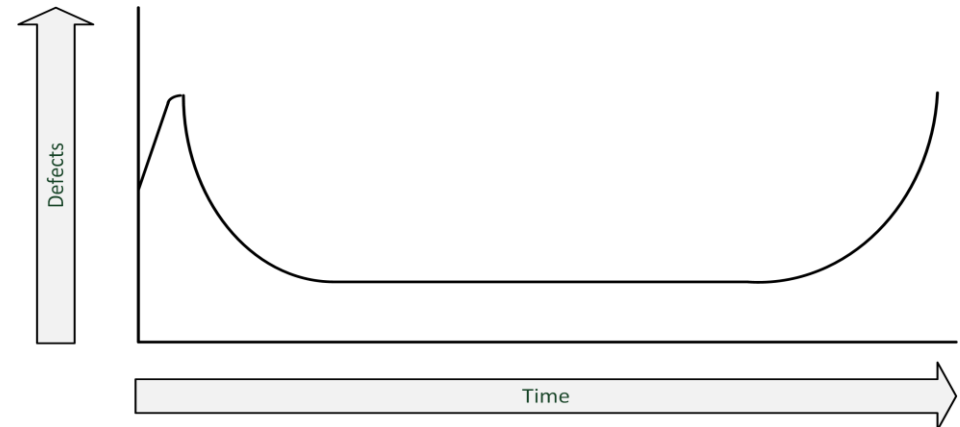
- Sometimes needed to perform systems management tasks:
 - Upgrading hardware or software
 - Implementing software changes
 - Migrating data
 - Creation of backups
- During planned maintenance the system is more vulnerable to downtime than under normal circumstances
 - A temporary SPOF could be introduced
 - Systems managers could make mistakes

Sources of Unavailability - Physical Defects

- Everything breaks down eventually
- Mechanical parts are most likely to break first
- Examples:
 - **Fans for cooling equipment** usually break because of dust in the bearings
 - **Disk drives** contain moving parts
 - **Tapes** are very vulnerable to defects as the tape is spun on and off the reels all the time
 - **Tape drives** contain very sensitive pieces of mechanics that can break easily

Sources of Unavailability - Bathtub Curve

- A component failure is most likely when the component is new
- Sometimes a component doesn't even work at all when unpacked for the first time. This is called a DOA component—Dead On Arrival.
- When a component still works after the first month, it is likely that it will continue working without failure until the end of its life



Sources of Unavailability - Environmental Issues

- Environmental issues can cause downtime. Issues with
 - Power
 - Cooling
 - External factors like:
 - Disasters
 - Fire
 - Earthquakes
 - Flooding

Sources of Unavailability - Complexity of the Infrastructure

- Adding more components to an overall system design can undermine high availability
 - Even if the extra components are implemented to achieve high availability
- Complex systems
 - Have more potential points of failure
 - Are more difficult to implement correctly
 - Are harder to manage
- Sometimes it is better to just have an extra spare system in the closet than to use complex redundant systems

Availability Patterns

- A single point of failure (SPOF) is a component in the infrastructure that, if it fails, causes downtime to the entire system.
- SPOFs should be avoided in IT infrastructures as they pose a large risk to the availability of a system.
- We just need to know what is shared and if the risk of sharing is acceptable.
- To eliminate SPOFs, a combination of redundancy, failover, and fallback can be used.

Redundancy

- Redundancy is the duplication of critical components in a single system, to avoid a single point of failure (SPOF)
- Examples:
 - A single component having two power supplies; if one fails, the other takes over
 - Dual networking interfaces
 - Redundant cabling

Failover

- Failover is the (semi)automatic switch-over to a standby system or component
- Examples:
 - Windows Server failover clustering
 - VMware High Availability
 - Oracle Real Application Cluster (RAC) database

Fallback

- Fallback is the manual switchover to an identical standby computer system in a different location
- Typically used for disaster recovery
- Three basic forms of fallback solutions:
 - Hot site
 - Cold site
 - Warm site

Business Continuity

- In case of a disaster, the infrastructure could become unavailable, in some cases for a longer period of time.
- Business continuity is about identifying threats an organization faces and providing an effective response.
- To handle the effect of disasters, following processes are
 - Business Continuity Management (BCM) and
 - Disaster Recovery Planning (DRP)

Business Continuity

- An IT disaster is defined as an irreparable problem in a datacenter, making the datacenter unusable

Natural disasters:

Floods

Hurricanes

Tornadoes

Earthquakes

Manmade disasters:

Hazardous material

spills

Infrastructure failure

Bio-terrorism

Information Technology Infrastructure

Information Technology Infrastructure

- **Credit Hours:** 3
- **Lecturer:** Shafaq Nisar
- **Lecturing Style:** Video Lectures of short duration (5-7 minutes)

Performance Concepts

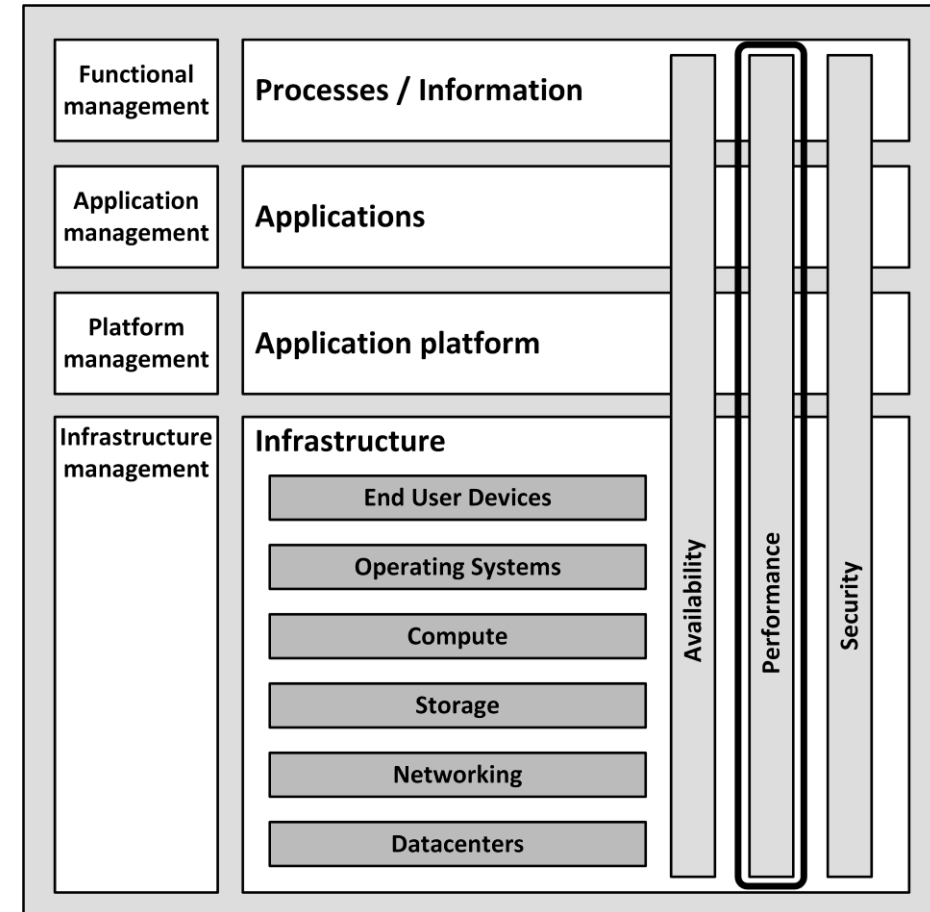
1. Introduction
2. Perceived Performance
3. Performance during Infrastructure Design
 - Benchmarking, User Profiling, Vendor experience
 - Prototyping
4. Performance of a Running System
 - Managing Bottlenecks and Performance Testing
5. Performance Patterns
 - Increasing performance on Upper layer
 - Caching
6. Grid Computing
7. Capacity management

Introduction

Performance Concepts

Introduction

- Performance is a typical hygiene factor
- Nobody notices a highly performing system
- But when a system is not performing well enough, users quickly start complaining

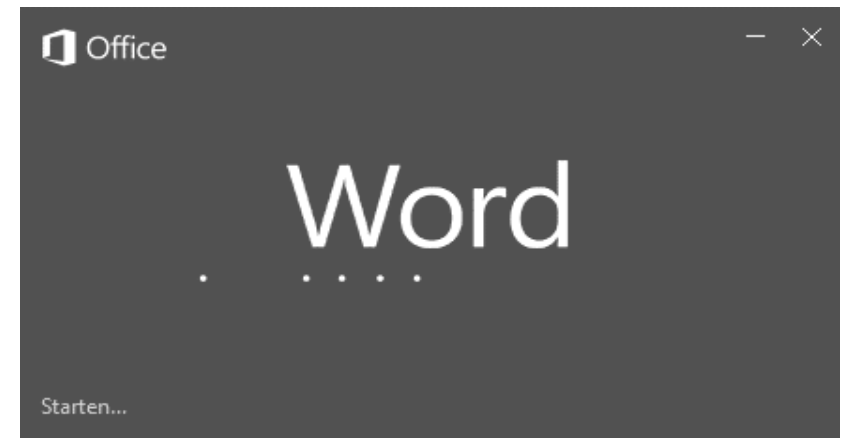
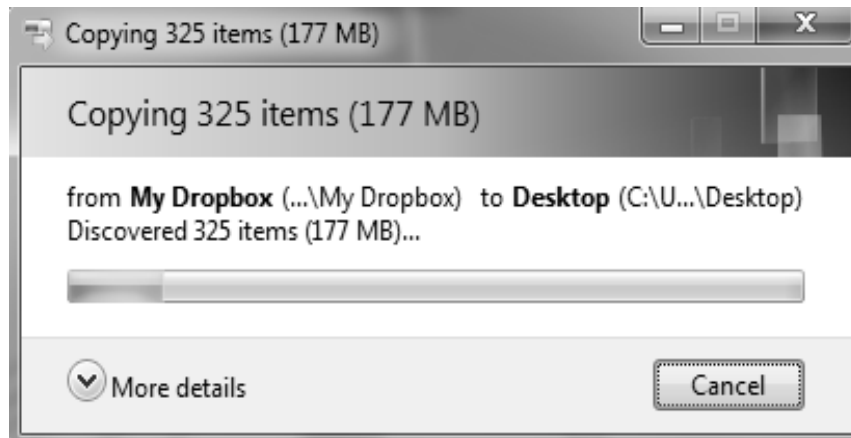


Perceived Performance

- Perceived performance refers to how quickly a system appears to perform its task
- In general, people tend to overestimate their own patience
- People tend to value predictability in performance
 - When the performance of a system is fluctuating, users remember a bad experience
 - Even if the fluctuation is relatively rare

Perceived Performance

- Inform the user about how long a task will take
 - Progress bars
 - Splash screens



Performance during Infrastructure Design

Performance during Infrastructure Design

- A solution must be designed, implemented, and supported to meet the performance requirements

Even under increasing load

- Calculating performance of a system in the design phase is:

Extremely difficult

Very unreliable

Performance during Infrastructure Design

- Performance must be considered:
 - When the system works as expected
 - When the system is in a special state, like:
 - Failing parts
 - Maintenance state
 - Performing backup
 - Running batch jobs

Performance during Infrastructure Design

- Some ways to do this are:
 - Benchmarking
 - Using vendor experience
 - Prototyping and User Profiling

Benchmarking

- A benchmark uses a specific test program to assess the relative performance of an infrastructure component
- Benchmarks compare:
 - Performance of various subsystems
 - Across different system architectures

Benchmarking

- Benchmarks comparing the raw speed of parts of an infrastructure
 - Like the speed difference between processors or between disk drives
 - Not taking into account the typical usage of such components
 - Examples:
 - Floating Point Operations Per Second – FLOPS
 - Million Instructions Per Second – MIPS of a CPU

Prototyping

- Also known as proof of concept (PoC)
- Prototypes measure the performance of a system at an early stage
- Building prototypes:
 - Hiring equipment from suppliers
 - Using data centre capacity at a vendor's premise
 - Using cloud computing resources
- Focus on those parts of the system that pose the highest risk, as early as possible in the design process

Vendor Experience

- The best way to determine the performance of a system in the design phase: use the experience of vendors
- They have a lot of experience running their products in various infrastructure configurations
- Vendors can provide:
 - Tools
 - Figures
 - Best practices

User Profiling

- Predict the load a new software system will pose on the infrastructure before the software is actually built
- Get a good indication of the expected usage of the system
- Steps:
 - Define a number of typical user groups (personas)
 - Create a list of tasks personas will perform on the new system
 - Decompose tasks to infrastructure actions
 - Estimate the load per infrastructure action
 - Calculate the total load

Performance of a Running System

Managing Bottlenecks

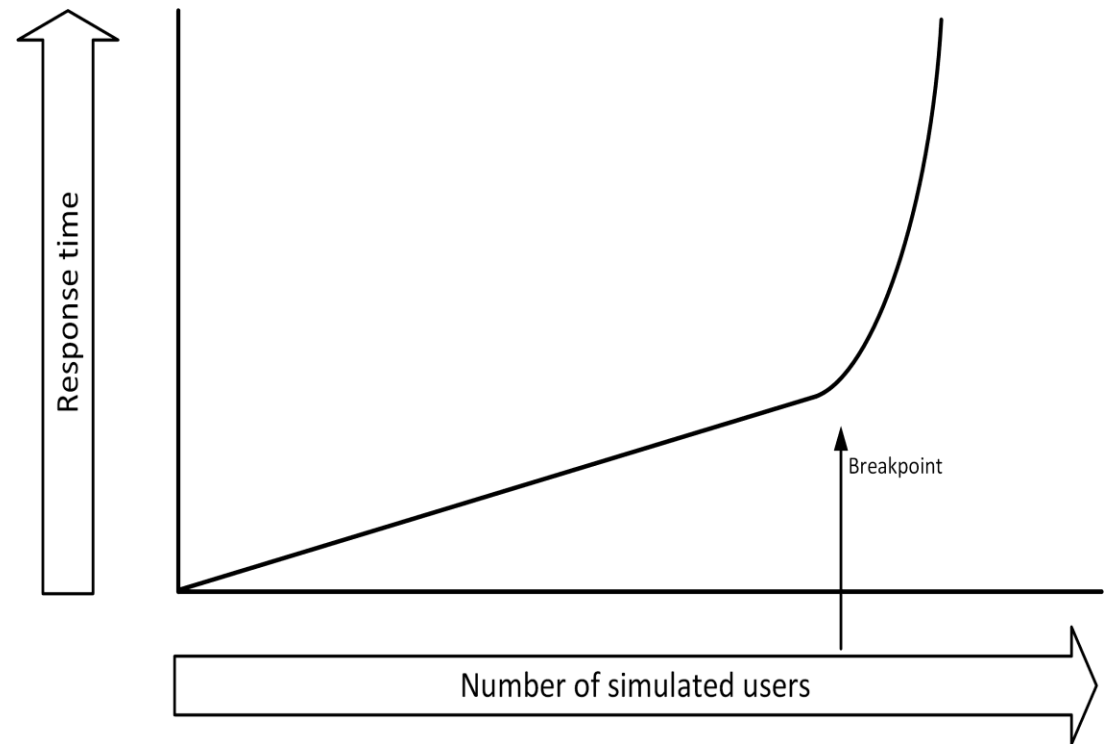
- The performance of a system is based on:
 - The performance of all its components
 - The interoperability of various components
- A component causing the system to reach some limit is referred to as the bottleneck of the system
- Every system has at least one bottleneck that limits its performance
- If the bottleneck does not negatively influence performance of the complete system under the highest expected load, it is OK

Performance Testing

- **Load testing** - shows how a system performs under the expected load
- **Stress testing** - shows how a system reacts when it is under extreme load
- **Endurance testing** - shows how a system behaves when it is used at the expected load for a long period of time

Performance Testing - Breakpoint

- Ramp up the load
 - Start with a small number of virtual users
 - Increase the number over a period of time
- The test result shows how the performance varies with the load, given as number of users versus response time.



Performance Testing

- Performance testing software typically uses:
 - One or more servers to act as injectors
 - Each emulating a number of users
 - Each running a sequence of interactions
 - A test conductor
 - Coordinating tasks
 - Gathering metrics from each of the injectors
 - Collecting performance data for reporting purposes

Performance Testing

- Performance testing should be done in a production-like environment
 - Performance tests in a development environment usually lead to results that are highly unreliable
 - Even when underpowered test systems perform well enough to get good test results, the faster production system could show performance issues that did not occur in the tests
- To reduce cost:
 - Use a temporary (hired) test environment

Performance Patterns

Increasing Performance on Upper Layers

- 80% of the performance issues are due to badly behaving applications
- Application performance can benefit from:
 - Database and application tuning
 - Prioritizing tasks
 - Working from memory as much as possible (as opposed to working with data on disk)
 - Making good use of queues and schedulers
- Typically more effective than adding compute power

Disk Caching

- Disks are mechanical devices that are slow by nature
- Caching can be implemented i:
 - Disks
 - Disk controllers
 - Operating system
- Cache memory:
 - Stores all data recently read from disk
 - Stores some of the disk blocks following the recently read disk blocks

Caching

Component	Time it takes to fetch <u>1 MB</u> of data (ms)
Network, 1 Gbit/s	675
Hard disk, 15k rpm, 4 KB disk blocks	105
Main memory DDR3 RAM	0.2
CPU L1 cache	0.016

Web Proxies

- When users browse the internet, data can be cached in a web proxy server
 - A web proxy server is a type of cache
 - Earlier accessed data can be fetched from cache, instead of from the internet
- Benefits:
 - Users get their data faster
 - All other users are provided more bandwidth to the internet, as the data does not have to be downloaded again

Grid Computing

- A computer grid is a high performance cluster that consists of systems that are spread geographically
- The limited bandwidth is the bottleneck
- Examples:
 - SETI@HOME
 - CERN LHC Computing Grid (140 computing centers in 35 countries)
- Broker firms exist for commercial exploitation of grids
- Security is a concern when computers in the grid are not under control

Capacity Management

- Capacity management guarantees high performance of a system in the long term
- To ensure performance stays within acceptable limits, performance must be monitored
- Trend analyses can be used to predict performance degradation
- Anticipate on business changes (like forthcoming marketing campaigns)

Information Technology Infrastructure

Information Technology Infrastructure

- **Credit Hours:** 3
- **Lecturer:** Shafaq Nisar
- **Lecturing Style:** Video Lectures of short duration (5-7 minutes)

Security Concepts

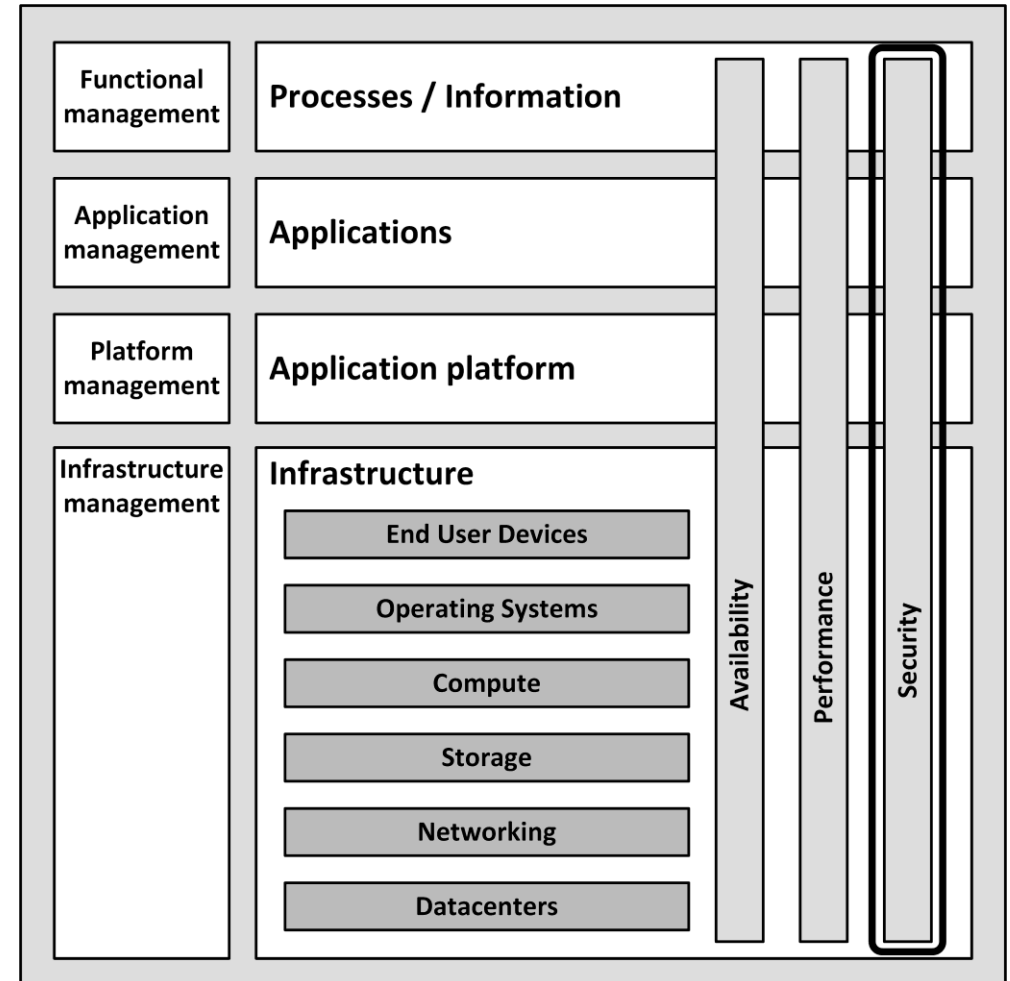
1. Introduction
2. Computer Crimes
3. Risk Management
 - Risk Response
 - Exploits
4. Security Controls
 - Attack vectors
5. Security Patterns
 - Identity and Access management and Layered security
 - Cryptography and Cryptographic Attacks

Introduction

Security Concepts

Introduction

- Security is the combination of:
 - Availability
 - Confidentiality
 - Integrity
- Focused on the recognition and resistance of attacks
- For IT infrastructures availability is a non-functional attribute in its own right



Computer Crimes

- Reasons for committing crime against IT infrastructures:
 - Personal exposure and prestige
 - Creating damage
 - Financial gain
 - Terrorism
 - Warfare

Personal Exposure and Prestige

- In the past, the hacker community was very keen on getting personal or group exposure by hacking into a secured IT infrastructure. When hackers proved that they could enter a secured system and made it public, they gained respect from other hackers.
- While nowadays most hacking activity is done for other reasons, there are still large communities of hackers that enjoy the game.

Creating Damage

- Creating damage to organizations to create bad publicity
- For instance, by defacing websites, bringing down systems or websites, or
- making internal documents public

Financial Gain

- For instance, by holding data hostage and asking for ransom money, stealing credit card data, changing account data in bank systems

OR

- Stealing passwords of customers and ordering goods on their behalf

Terrorism

- The main purpose of terrorism is creating fear in a society
- A well-planned attack targeted at certain computer systems, like the
- Computer system that manages the water supply
- or
- A nuclear power plant, could result in chaos and fear amongst citizens

Warfare

- Certain governments use hacking practices as acts of war
- Since economies and societies today largely depend on the IT infrastructures, bringing important IT systems down in a certain country could cause the economy to collapse.
- Bringing down the internet access of a country for example means: no access to social media, no e-mails, no web shops, no stock trading, no search engines, etc.

Risk management

Risk management

- ❖ Managing security is all about managing risks
- ❖ The effort we put in securing the infrastructure should be directly related to the risk at hand
- ❖ Risk management is the process of:
 - ❑ Determining an acceptable level of risk
 - ❑ Assessing the current level of risk
 - ❑ Taking steps to reduce risk to the acceptable level
 - ❑ Maintaining that level

Risk list

A risk list can be used to quantify risks

Risk is calculated based on:

Asset name - component that needs to be protected

Vulnerability - weakness, process or physical exposure that makes the asset susceptible to exploits

Exploit - a way to use one or more vulnerabilities to attack an asset

Probability - an estimation of the likelihood of the occurrence of an exploit

Impact - the severity of the damage when the vulnerability is exploited

Example of Part of a Risk List

Asset	Vulnerability	Exploit	P	I	R
Laptop	Laptop gets stolen	Sensitive data on hard disk is exposed	5	3	15
Printer	Printer hard disk contains sensitive data	Repair man could swap hard disk and the hard disk could get on the market with sensitive data	1	3	3
Work-stations	Virus attack unknown to virus scanner	Unavailability or disclosure of data	2	3	6
SAN storage system	Data protection via LUN masking contains error	Data could get exposed to wrong server	1	2	2

Risk Response

- Controls can be designed and implemented based on identified severity of the risk in the risk list.
- There four risk responses:
 - Acceptance of the risk
 - Avoidance of the risk - do not perform actions that impose risk
 - Transfer of the risk - for instance transfer the risk to an insurance company
 - Mitigation of the risk and accepting the residual risk

Exploits

- Information can be stolen in many ways
- Examples:
 - Key loggers can send sensitive information like passwords to third parties
 - Network sniffers can show network packages that contain sensitive information or replay a logon sequence
 - Data on backup tapes outside of the building can get into wrong hands
 - Disposed PCs or disks can get into the wrong hands
 - Corrupt or dissatisfied staff can copy information
 - End users are led to a malicious website that steals information (phishing)

Security Controls

CIA

- Three core goals of security (CIA):

- ☐ Confidentiality

- ☐ Integrity

- ☐ Availability

CIA

- Confidentiality - prevents the intentional or unintentional unauthorized disclosure of data
- Integrity - ensures that:
 - No modifications to data are made by unauthorized staff or processes
 - Unauthorized modifications to data are not made by authorized staff or processes
 - Data is consistent
- Availability - ensures the reliable and timely access to data or IT resources

- Example of confidentiality levels

Confidentiality Level	Description
1	Public information
2	Information for internal use only
3	Information for internal use by restricted group
4	Secret: reputational damage if information is made public
5	Top secret: damage to organization or society if information is made public

- Example of integrity levels

Integrity Level	Description
1	Integrity of information is of no importance
2	Errors in information are allowed
3	Only incidental errors in information are allowed
4	No errors are allowed, leads to reputational damage
5	No errors are allowed, leads to damage to organization or society

- Example of availability levels

Availability Level	Description
1	No requirements on availability
2	Some unavailability is allowed during office hours
3	Some unavailability is allowed only outside of office hours
4	No unavailability is allowed, 24/7/365 availability, risk for reputational damage
5	No unavailability is allowed risk for damage to organization or society

Security Controls

- Controls mitigate risks
- Security controls must address at least one of the CIA
- Information can be classified based on CIA levels
- Controls can be designed and implemented based on the identified risk level for CIA

Security controls -Example

[illegible]

Attack Vectors

- Malicious code
 - Applications that, when activated, can cause network and server overload, steal data and passwords, or erase data
- Worms
 - Self-replicating programs that spread from one computer to another, leaving infections as they travel

Attack Vectors

- Virus
 - Self-replicating program fragment that attaches itself to a program or file enabling it to spread from one computer to another, leaving infections as it travels
- Trojan Horse
 - Appears to be useful software but will actually do damage once installed or run on your computer

Attack Vectors

- **Denial of service attack**

- An attempt to overload an infrastructure to cause disruption of a service
- Can lead to downtime of a system, disabling an organization to do its business
- In a Distributed Denial of Service (DDoS) attack the attacker uses many computers to overload the server
- Groups of computers that are infected by malicious code, called botnets, perform an attack

Attack Vectors

- Preventive DDoS measures:
 - Split business and public resources
 - Move all public facing resources to an external cloud provider
 - Setup automatic scalability (auto scaling, auto deployment) using virtualization and cloud technology
 - Limit bandwidth for certain traffic
 - Lower the Time to Live (TTL) of the DNS records to be able to reroute traffic to other servers when an attack occurs
 - Setup monitoring for early detection

Attack Vectors

- Phishing
 - A technique of obtaining sensitive information
 - The phisher sends an e-mail that appears to come from a legitimate source, like a bank or credit card company, requesting "verification" of information
 - The e-mail usually contains a link to a fraudulent web page

Security Patterns

Identity and Access Management (IAM)

- The process of managing the identity of people and systems, and their permissions
- The IAM process follows three steps:
 - Users or systems claim who they are: **identification**
 - The claimed identity is checked: **authentication**
 - Permissions are granted related to the identity and the groups it belongs to: **authorization**

Layered Security

- Layered security (also known as a Defense-In-Depth strategy) implements various security measures in various parts of the IT infrastructure
 - Instead of having one big firewall and have all your security depend on it, it is better to implement several layers of security
- Preferably security layers make use of different technologies
 - This makes it harder for hackers to break through all barriers, as they will need specific knowledge for each step
- Disadvantage: increases the complexity of the system

Cryptography

- The practice of **hiding information** using encryption and decryption techniques
- **Encryption** is the conversion of information from a readable state to apparent random data
- Only the receiver has the ability to **decrypt** this data, transforming it back to the original information
- A **cipher** is a pair of algorithms that implements the encryption and decryption process. The operation of a cipher is controlled by a **key**

Cryptography

- Block ciphers
 - Input:
 - A block of plaintext
 - A key
 - Output:
 - A block of cipher text
- Used across a wide range of applications, from ATM machine data encryption to e-mail privacy and secure remote access
- Standards:
 - Data Encryption Standard (DES)
 - Advanced Encryption Standard (AES)

Cryptography

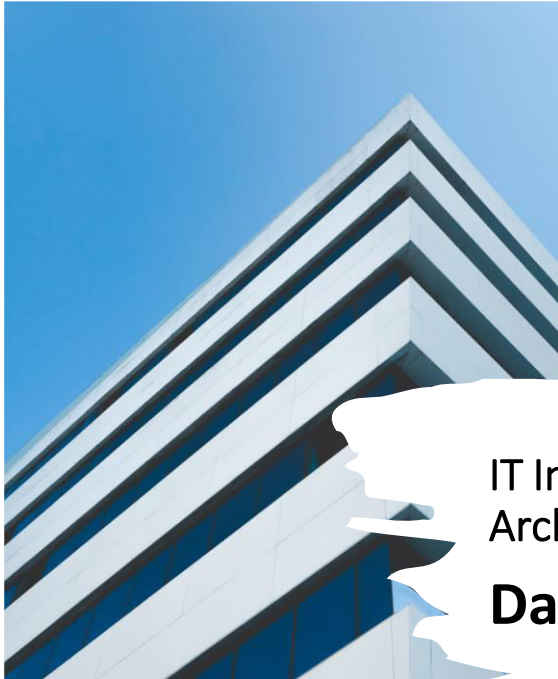
- Stream ciphers
 - Create an arbitrarily long stream of key material
 - Combines key stream with the plaintext bit-by-bit or character-by-character
 - Used when data is in transit over the network
 - RC4 is a widely-used stream cipher

Cryptographic Attacks

- Every encryption method can be broken using a brute force attack
 - Except a one-time pad cipher with the key of equal or greater length than the message
- A brute force attack consists of systematically checking all possible keys until the correct key is found
- The amount of effort needed is exponentially dependent on the size of the key

Cryptographic Attacks

- Effective security could be achieved if it is proven that no efficient method (as opposed to the time consuming brute force method) can be found to break the cipher
- Most successful attacks are based on flaws in the implementation of an encryption cipher
- To ensure a cipher is flawless, the source code is usually open source and thus open to inspection to everyone



IT Infrastructure
Architecture

Data Centers



Topic outline :

What is Data Center ?

Today's Data Center

Data Center Categories

Data Center :

Centralized facility used by an organizations to store, manage, and disseminate large amounts of data.

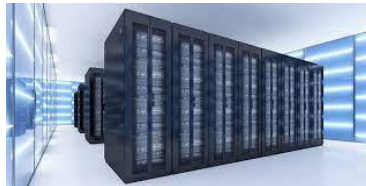
Houses servers, networking equipment, storage devices, and other infrastructure required to support data processing and storage.

Provides power supply, cooling, fire prevention and detection, equipment racks, and other facilities needed to host the installed infrastructure components.

Today's Data Center :



Large datacenters contain shipping containers packed with thousands of servers each .



When repairs or upgrades are needed, entire containers are replaced.

Data Center Categories :

Typical datacentre categories:

- **Sub Equipment Room (SER)** – a SER is also known as a patch closet
- **Main Equipment Room (MER)** – a MER is a small datacenter in the organization's subsidiaries or buildings
- **Organization owned datacenter** – a datacenter that contains all central IT equipment for the organization
- **Multi-tenant datacenter** – used by service providers that provide services for multiple other organizations. These datacenters are typically the largest



Datacenter location

Environment

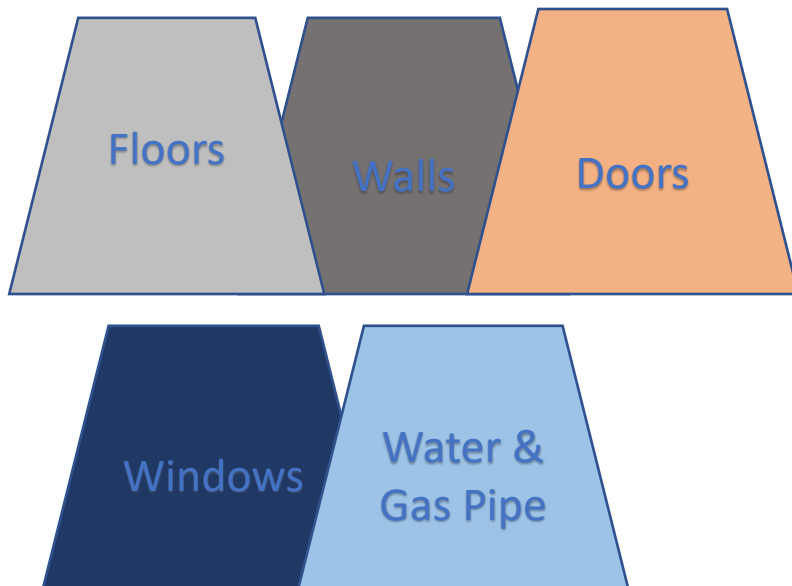
Visibility

Utilities

Located in foreign countries

Physical structure

The physical structure of a datacentre includes components that need special attention:



Floors

- One fully filled 19" computer rack weighs up to 700 kg
- In a typical datacenter, the floor must be able to carry 1500 to 2000 kg/m²



Floors

- Raised floors consist of a metal framework carrying removable floor tiles
- Tiles are usually 60×60 cm
- Tiles can be lifted individually to reach cables installed under the raised floor
- Vents provide cool air flow to the racks placed on the floor
- Under the raised floor, data and power cables are installed
 - As alternative, overhead cable trays can be used



Floors

Disadvantages

- They are expensive
- The maximum floor load is limited
- Doors and equipment loading slopes are hard to install due to the difference in floor height
- Under the raised floor, a fire could easily spread through the entire datacenter



Walls, windows, and doors

- Walls should reach from the floor to the building's ceiling
 - Because of fire safety and physical intrusion prevention
- Windows are not desirable in a datacenter
- Windows must be:
 - Translucent
 - Shatterproof
 - Impossible to open
- Doors should be large enough to have equipment brought in
- Doors must resist forced entry



Water and gas pipes

- Water or gas pipes may have been installed:
 - Under the floor
 - In the walls
 - Above the ceiling of the datacenter
- Datacenter operators should know where the shutoff valves are



Power supply

- Energy usage is a key issue for datacenters
- Power drawn by datacenters:
 - A few kilowatts (kW) for one rack of servers
 - Dozens of megawatts (MW) for large facilities.

Power supply in a datacenter includes:

- Utility power supply
- Backup power systems(UPS, Generators)
- Power Distribution units



Utility Power Supply (UPS)

Power issues can occur in the utility power supply.

- Types of power issues:
 - Blackout
 - Surge
 - Spike
 - Waveform issues
- Possibly leading to:
 - Downtime
 - Damage to equipment



Backup Power Systems

Uninterruptable Power Supply (UPS)

- Independent of the Utility power supply
- Provide power for short period of time

Power Generators

- **Power the data centre for an indefinite period of time as long as fuel is available**

Power distribution

A power distribution unit (PDU) is a device with multiple power outlets

- Distributes power to equipment located in the datacenter

Two types of PDUs:

- Large floor mounted PDUs take main feeds (usually 3 phase power) and distribute it into multiple smaller feeds to computer racks
- Power Strips that feed equipment in racks

Most Infrastructure components can be equipped with two power supplies for redundancy.
For availability reasons at least two power strips are needed to power equipment in a rack

Cooling

90% of all power used by IT infrastructure components is converted into heat, all heat has to be dissipated by a cooling system

Two types of cooling systems:

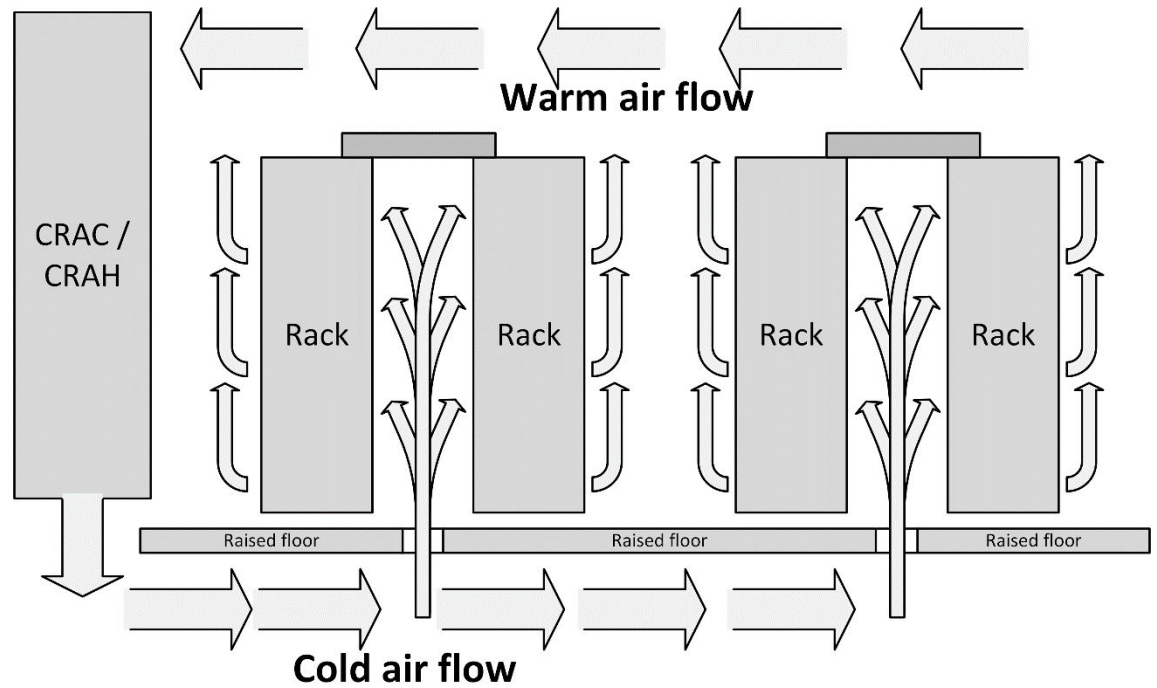
- **Computer Room Air Conditioners (CRAC)**

Refrigerant-based units connected to outside condensing units

- **Computer Room Air Handlers (CRAH)**

A chiller produces chilled water via a refrigeration process

Airflow



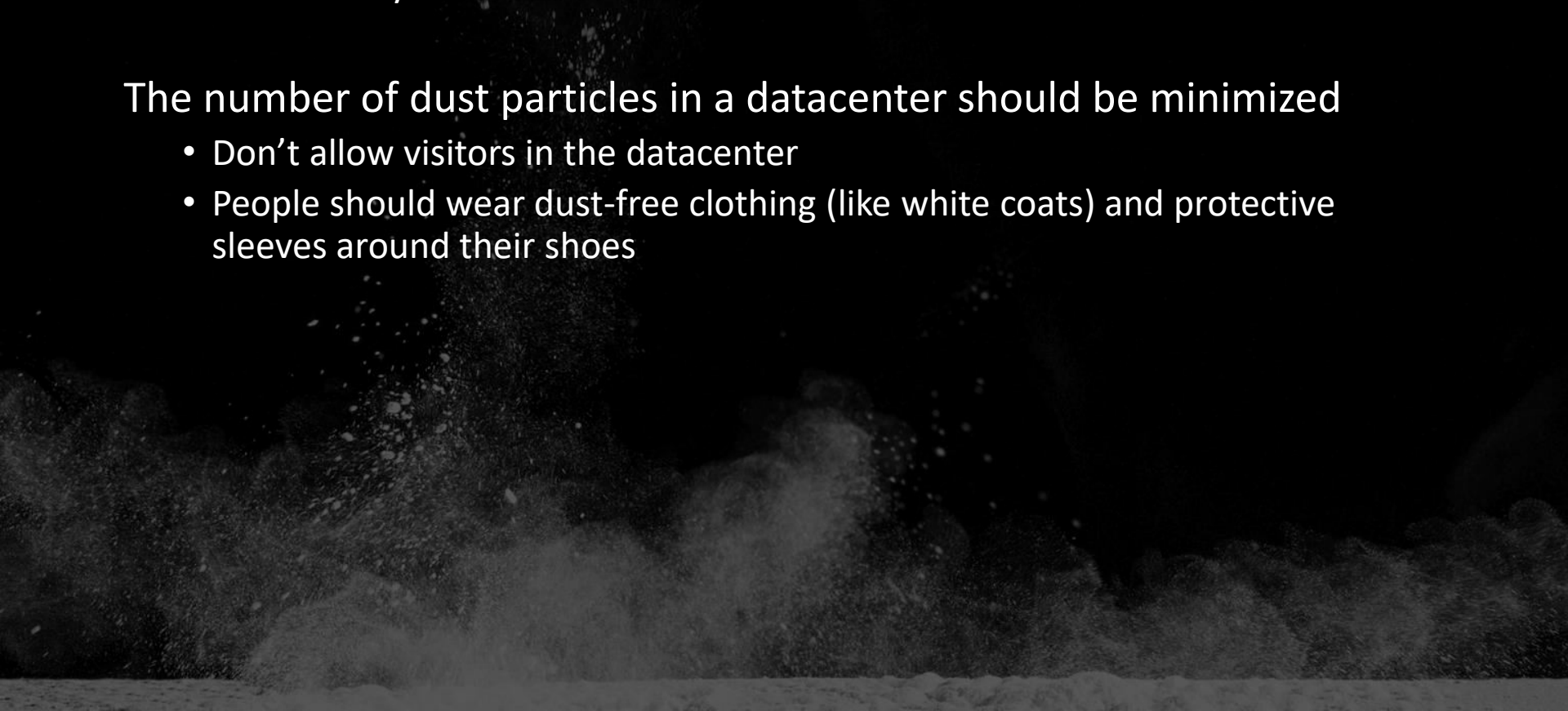
Humidity and dust

The humidity of the air in a datacenter is critical for the IT infrastructure components

- Air humidity should be between 40% and 60%

The number of dust particles in a datacenter should be minimized

- Don't allow visitors in the datacenter
- People should wear dust-free clothing (like white coats) and protective sleeves around their shoes



Operating temperatures

Infrastructure components have maximum operating temperatures:

- Servers shut themselves down at an air inlet temperature of 40 degrees Celsius
- The air temperature in the datacenter usually ranges from 18 degrees to 27 degrees Celsius.

Fire prevention, detection, and suppression

Fire is one of the main enemies of a datacenter

- A short circuit in a cable
- Defect equipment

Fires can spread around very quickly

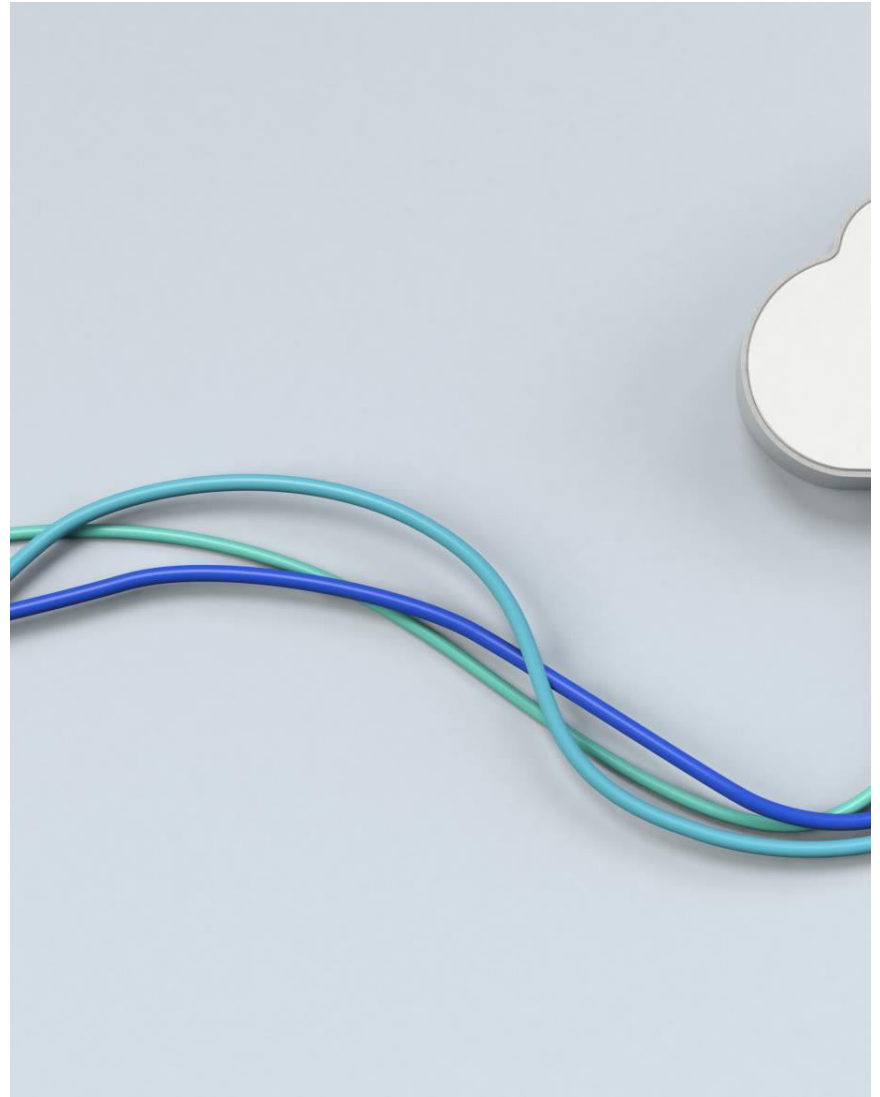
- Because of the air flow in the datacenter and the frequent use of raised floors

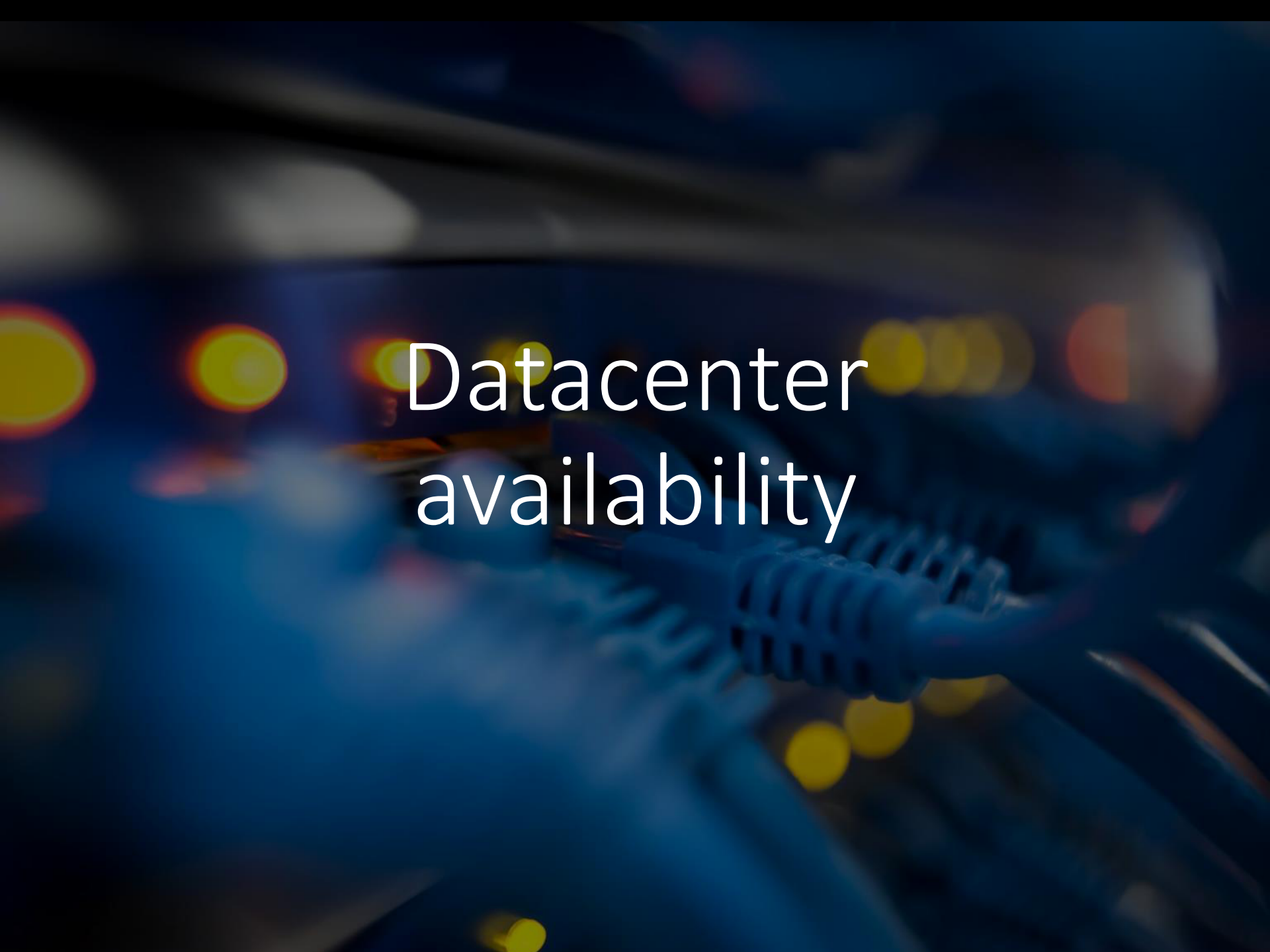
Smoke could damage equipment in the datacenter

- Even if a fire starts outside of the datacenter's computer room

Datacenter energy efficiency

- The Power Usage Effectiveness (PUE) metric measures the power used by the datacenter
- The PUE is calculated by dividing the amount of power used by the datacenter, by the power used to run the IT equipment in it
- Typical PUE value of a datacenter is between 1.1 and 2.0



The background is a blurred image of a datacenter. It features several glowing yellow lights, likely from server racks, and blue cables or conduits running across the scene. The overall color palette is dark blue and black, with the yellow lights providing a strong contrast.

Datacenter availability

Availability tiers

Tier	Measures	Expected downtime
Tier 1 <u>Availability</u> 99.671% <u>Type</u> Basic	Single path for power and cooling distribution No redundant components	Downtime very likely for planned and unplanned maintenance
Tier 2 <u>Availability</u> 99.741% <u>Type</u> Redundant components	Fulfills all Tier 1 requirements Single path for power and cooling distribution Redundant components	Downtime likely for planned and unplanned maintenance

Availability tiers

Tier 3 <u>Availability</u> 99.982% <u>Type</u> Concurrently maintainable	Fulfills all Tier 1 and Tier 2 requirements Multiple active power and cooling distribution paths Only one path active Redundant components All IT equipment must be dual-powered	No downtime due to planned maintenance Downtime unlikely for unplanned maintenance
Tier 4 <u>Availability</u> 99.995% <u>Type</u> Fault tolerant	Fulfills all Tier 1, Tier 2, and Tier 3 requirements Multiple active power and cooling distribution paths Redundant components All cooling equipment is independently dual-powered, including chillers and Heating, Ventilating and Air Conditioning (HVAC) systems	No downtime due to planned or unplanned maintenance



Datacentre Performance

The datacenter itself does not provide performance to IT Infrastructures, except for the bandwidth of the internet connectivity and the scalability of the location.

Performance can be measured by :

- Uptime
- Response time
- Throughput
- Scalability

Datacenter security



Physical security



Ensure that equipment is physically safe behind the datacenter doors



Physical access to the datacenter must be restricted to selected and qualified staff



An entry registration system should be used



A log should be maintained containing all staff entering and leaving the datacenter



Doors must be secured using conventional locks (for instance for dock loading doors) or electronic locks

Electronic locks

should open only after proper

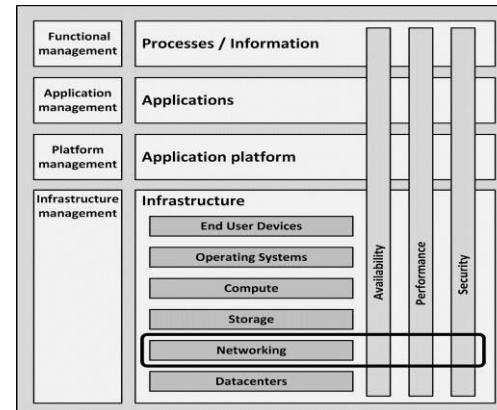
IT Infrastructure Architecture

Infrastructure Building Blocks
and Concepts

Networking – Part 1
(chapter 8)

Introduction

- IT Infrastructure-Networking
- Stand-alone machines Mainframe Computers in the 1960s-Computing and Punching machines
- ARPANET-Late 1960s, a number of computers were connected by The interface message processor (IMPS) was the first packet-router.
- The predecessor of the Internet
- With PCs in the 1980s, local Area Networks (LANs) were introduced



Mainframe Computers



Local Area Network

Networking Building Blocks

OSI Reference Model

- The OSI Reference Model (OSI-RM) was developed in 1984 by the International Organization for Standardization (ISO)
- Seven layers define the different stages that data must go through to travel from one host to another over a network

7 Layers of the OSI Model

Application

- End User layer
- HTTP, FTP, IRC, SSH, DNS

Presentation

- Syntax layer
- SSL, SSH, IMAP, FTP, MPEG, JPEG

Session

- Synch & send to port
- API's, Sockets, WinSock

Transport

- End-to-end connections
- TCP, UDP

Network

- Packets
- IP, ICMP, IPsec, IGMP

Data Link

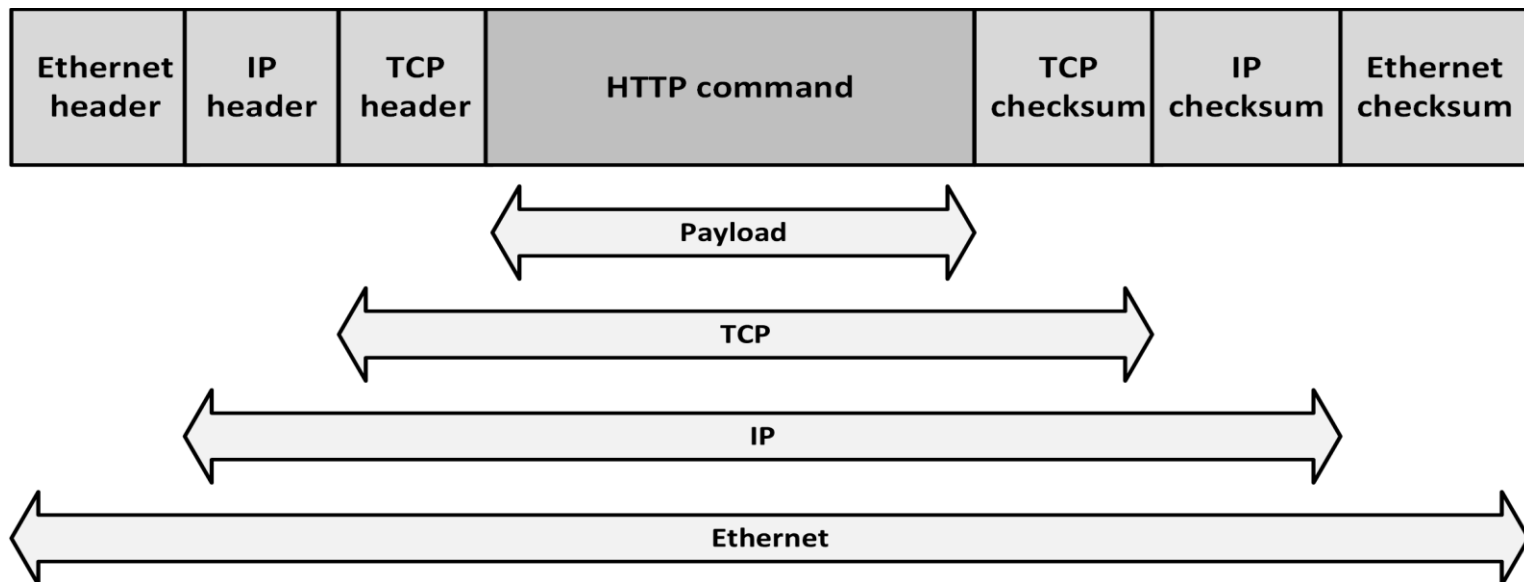
- Frames
- Ethernet, PPP, Switch, Bridge

Physical

- Physical structure
- Coax, Fiber, Wireless, Hubs, Repeaters

OSI Reference Model

- The OSI stack allows:
 - Implementing network components independently of each other
 - Ensuring all components work together
- Provides freedom to implement the network stack in an optimal way for a certain usage
- Each layer's payload contains the protocol for the next layer



Physical Layer

Cables

At the most elementary level, networking is about cables

- **Copper based cables:**
 - Coax
 - Twisted pair
- UTP comes in several quality ratings called categories
- **fiber optic cable**
- multiple strands of fiber glass or plastic
 - Each provide an optical path for light pulses
- Light source:
 - Light-emitting diode (LED)
 - Laser
- Two types of fiber optic cable are most common:
 - Multi-Mode Fiber (MMF)
 - Single Mode Fiber (SMF)



Category	Maximum bandwidth
5 or 5e	1 Gbit/s
6	10 Gbit/s
7	10 Gbit/s
8	40 Gbit/s



MULTIMODE

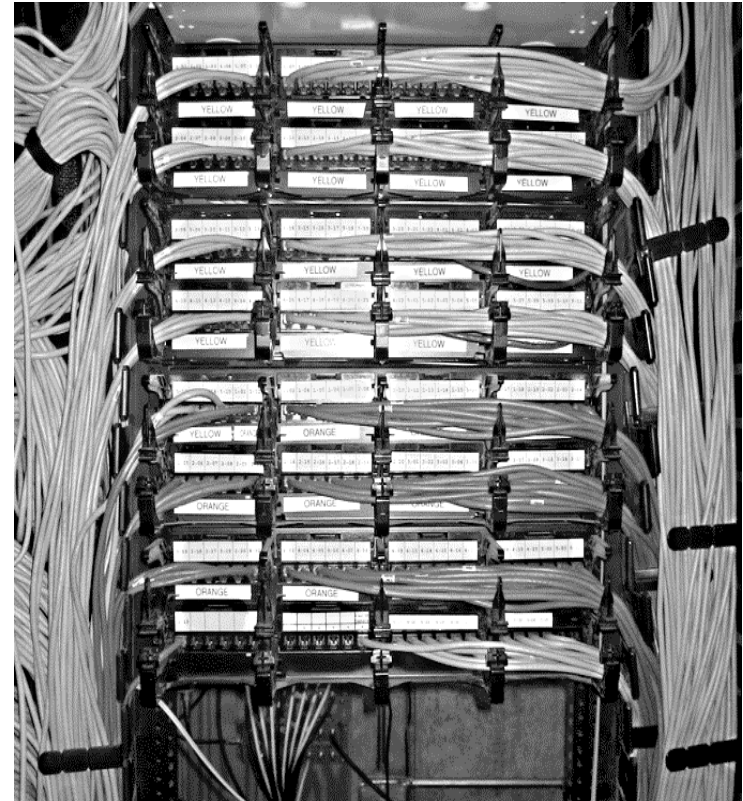


SINGLE MODE



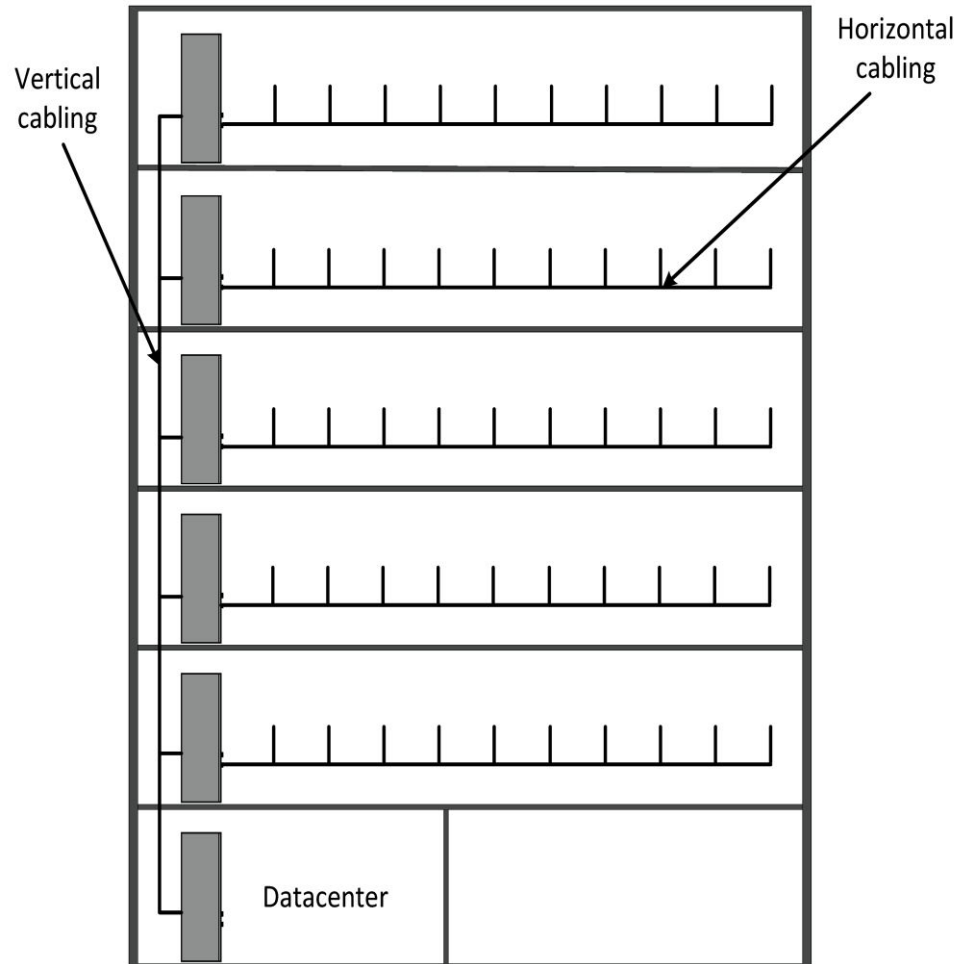
Patch Panels

- Cables in buildings are most visible in patch panels
 - In racks in the datacenter
 - In patch closets in various locations in (office) buildings
- They connect systems in a flexible way, without having to change the installed cabling in the building
- Patch panels are passive connecting devices
- Connecting systems is done using patch cables



Vertical and Horizontal Cabling

- The main distribution cabling in buildings connects the patch panels on the floors to the datacentre (vertical cabling)
- Endpoints in the walls are connected to the patch panels (horizontal cabling)



Leased Lines

Leased lines are dedicated data connections between two locations, provided by a telecom provider



Leased lines are based on: T or E carrier lines, SONET, SDH and Dark fiber

Internet Access

- Three ways to connect to the internet:

Leased line

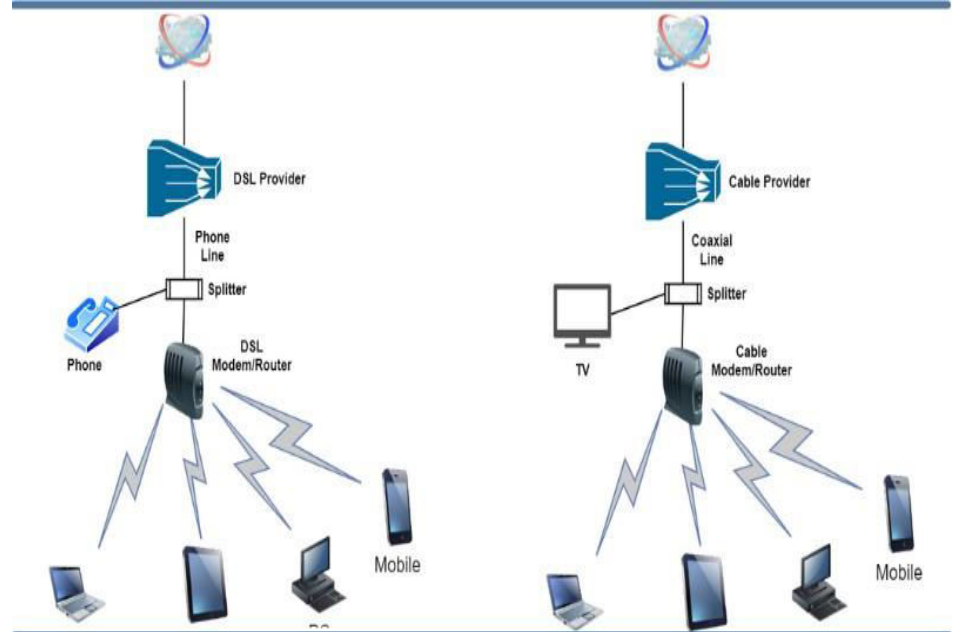
Cable internet access

- Uses cable television infrastructure

Digital Subscriber Line (DSL)

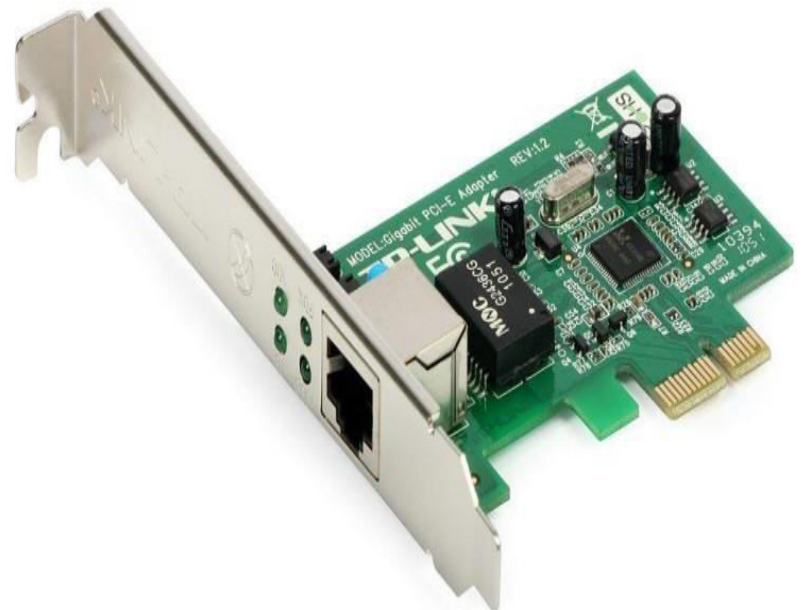
- Asymmetric DSL (ADSL)
- Symmetric DSL (SDSL)
- Very High DSL (VDSL)

Leased Line Internet Connectivity



Network Interface Controllers (NICs)

- Hardware component that connects a server or end user device to a physical network cable
- The NIC is actually both a physical layer and data link layer device
 - Provides physical access to a networking cable and an implementation of a datalink protocol like Ethernet
- A NIC has a fixed MAC address that is uniquely assigned to its network interface

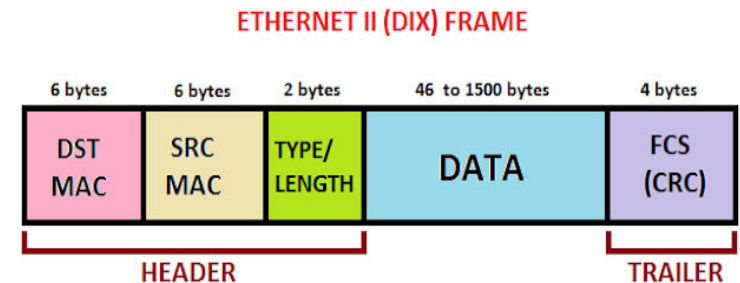
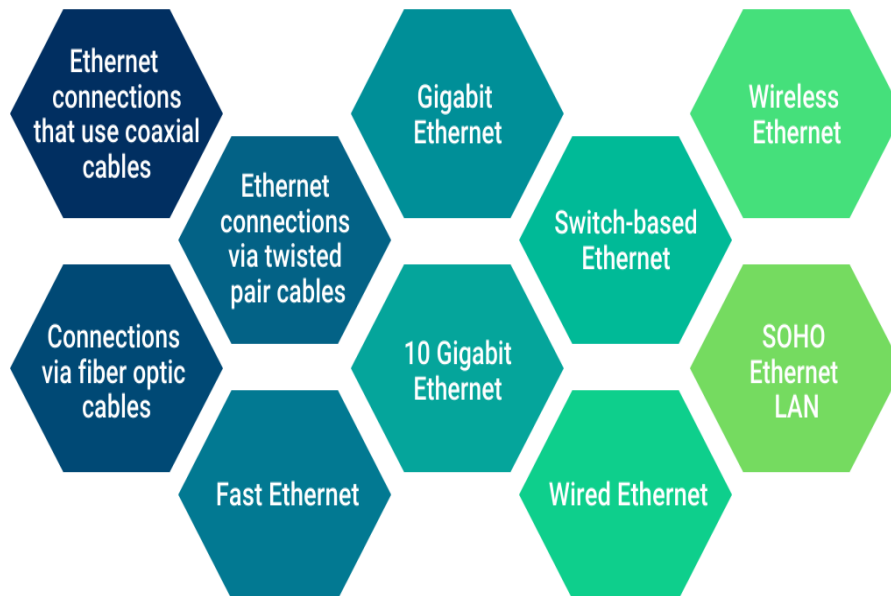


Datalink Layer

Ethernet

- Developed at Xerox PARC between 1973 and 1975

Types of Ethernet



Ethernet CSMA/CD

- Carrier Sense Multiple Access with Collision Detection
- Any machine can start transmitting packets when the shared carrier is not in use
 - Coax cable, twisted-pair hub or Wi-Fi radio signal spectrum
- Carrier sensing circuitry checks the activity on the carrier
- When two machines start to transmit a packet at the same time, a packet collision occurs
 - This is detected by all sending machines
 - They will stop the transmission immediately
 - After a short waiting time, they will retransmit their packet when the carrier is not in use anymore

Ethernet: CSMA/CD (continued)

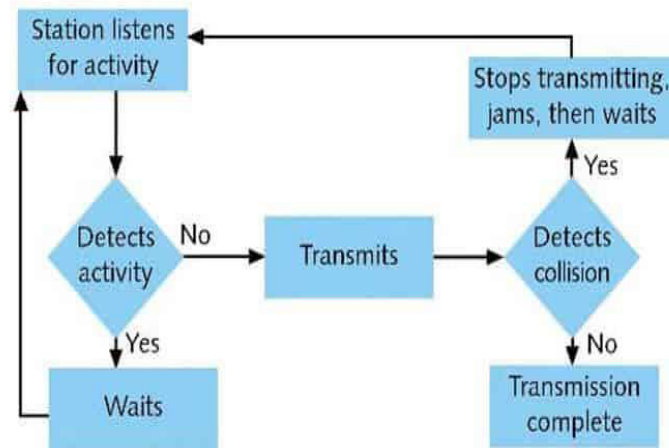
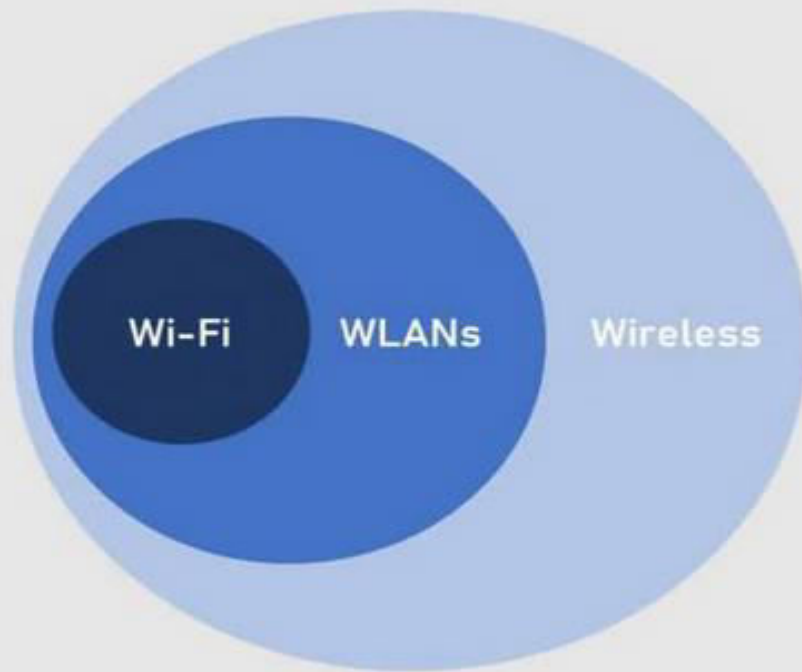


Figure 6-11: CSMA/CD process

WLAN (Wi-Fi)



Wi-Fi vs. WLANs vs. Wireless

Wi-Fi, is a certain type of WLAN that uses specifications of the 802.11 wireless standard.

The terms WLAN and Wi-Fi are often linked and used interchangeably. A WLAN can be built on various wireless technologies.

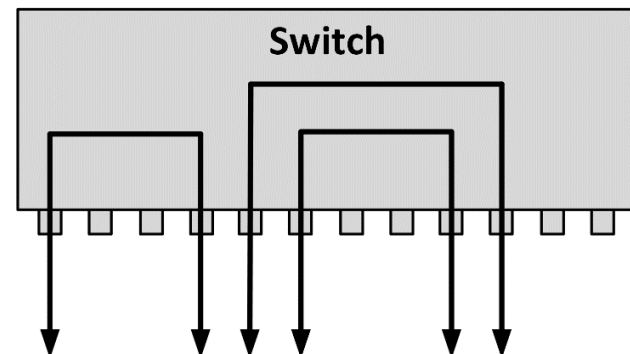
Wi-Fi networks are WLANs. But Wi-Fi is not the only type of WLAN.

WLAN (Wi-Fi)

- Wi-Fi range is about 30 m
- Access points are base stations for a wireless network
- Data encryption: Wi-Fi Protected Access (WPA)
 - WPA dynamically generates a new key for each packet
 - WPA includes a Message Integrity Check
 - Prevents an attacker from capturing, altering and/or resending data packets

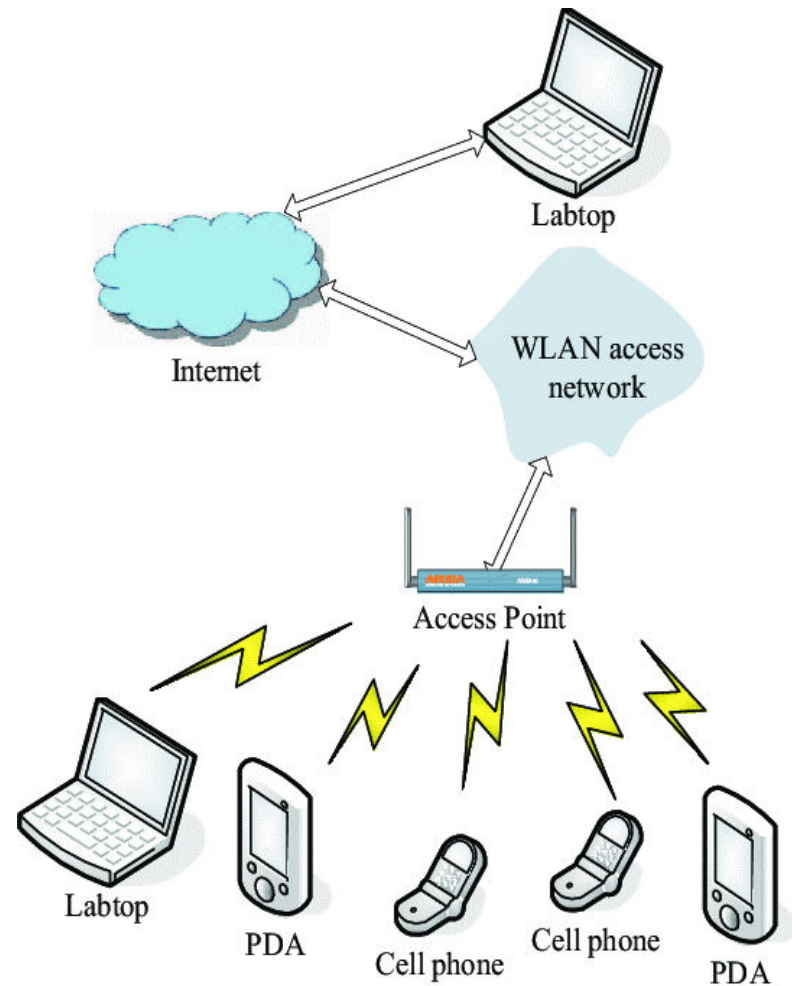
Switching

- **Switches split a single network segment into multiple segments**
 - Each segment has one device
- **Switches learn which MAC address is connected to which port**
 - Data sent to a certain MAC address will only be forwarded to the switch port that has that MAC address connected
- **On a switched network, many simultaneous data transfers can take place, in full-duplex**



WAN

- Wide Area Networks (WANs) started in the 1980s
- Packet Switching technologies in WLAN
- Reliable Network
- Most WAN Connections Converted to VPN using
 - MPLS network of a network provider
 - The internet using IPsec or SSL
 - Dark fiber



Public Wireless Networks

- Public wireless (mobile) networks are getting more popular every day
- Public wireless networks are much less reliable than private wireless networks and have lower bandwidth
- Technologies:
 - 1G and 2G: GSM, CDMA, GPRS and EDGE
 - 3G: UMTS and HSDPA
 - 4G: LTE

Network Layer

The IP Protocol

- IP is the defining set of protocols that enable the modern internet.
- IP, in combination with TCP, was invented by Robert Kahn and Vinton Cerf in 1973
- The IP protocol-mostly used layer 3 protocol in the world
- IPv4 is the dominant protocol on the internet today
- The IP protocol assumes that the network is inherently unreliable and that it is dynamic in terms of availability of links and nodes
- IP uses data packets that contain:
 - Source address
 - Destination address
 - Payload data (typically an Ethernet packet)

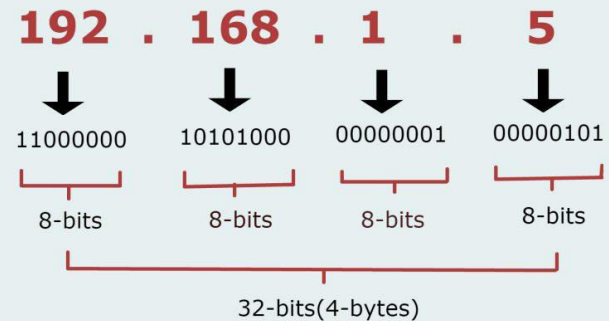
The IP Protocol

- IP routing protocols dynamically define the path of IP packets from source to destination
- Routing issues:
 - Due to network disruption, IP packets can get lost or corrupted
 - When an error is detected, the IP packet is dropped by the node that found the error
 - Since each IP packet is routed individually, IP packets can arrive at the destination out of order
- The effects of dropped IP packets and IP packets arriving out of order is handled by upper layer protocols like TCP

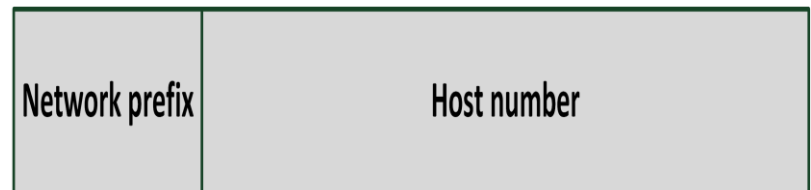
IPv4 Addresses

- IPv4 addresses are composed of 4 bytes (32 bits)
- An IP address has a network prefix and a host number
- All hosts with the same network prefix can communicate directly to each other
- Hosts in other networks can only be reached using a router

IPV4 address represented in dotted-decimal notation



10 . 121 . 12 . 16



IPv4 Classes

- First three bits of the first byte of an IP address define the class of the address
- Three classes of networks are defined

Class A



Class B



Class C



Class	First byte	Max number of hosts	Number of available networks
A	0-127	16,777,214	128
B	128-191	65,534	16,384
C	192-223	254	2,097,152

IPv4 Subnetting

- Subnetting is used to split up the host part of an IP network in smaller subnets, each forming a new IP network

CIDR prefix	Subnet mask	Available subnets	Hosts per subnet
/24	255.255.255.0	1	254
/25	255.255.255.128	2	126
/26	255.255.255.192	4	62
/27	255.255.255.224	8	30
/28	255.255.255.240	16	14
/29	255.255.255.248	32	6
/30	255.255.255.252	64	2
/31	255.255.255.254	128	2 (only point-to-point)

Address 196 . 121 . 12 . 241

Network prefix	Sub net	Host
----------------	---------	------

Subnet mask 255 . 255 . 255 . 240

Binary 11111111 . 11111111 . 11111111 . 11110000

IPv4 - Private IP Ranges

- Private IP addresses should be used for **LANs**
 - The number of unique IP addresses on the internet is limited
 - Hosts with public internet IP addresses can reach the internet directly
- Private IP address ranges:
 - 10.0.0.0 to 10.255.255.255 (class A address range)
 - 172.16.0.0 to 172.31.255.255 (class B address range)
 - 192.168.0.0 to 192.168.255.255 (class C address range)
- Private IP addresses:
 - Are not used on the internet
 - Are not routed by internet routers

IPv6

- IPv6 was introduced in 1998 as a successor of IPv4 to solve the problem of limited IP address space
- IPv6 uses 128-bit addresses represented in eight groups of four hexadecimal digits separated by colons
- Example:

2001:0bb8:86a2:0000:0000:8b1e:1350:7c34

IPv6

- IPv6 has the following benefits over IPv4:
 - Expanded address space
 - Better support for mobile IP
 - Fixed header length
 - Auto configuration
 - Quality of Service
 - Security
 - MTU discovery
- IPv6 is not backwards compatible with IPv4
- Deployment models for IPv6:
 - Use IPv6 on the LAN and on dedicated WAN links
 - Protocol translation
 - Dual stack
 - IPv6 over IPv4 tunnels
- Dual stack is the simplest way to begin deploying IPv6

ICMP

- The Internet Control Message Protocol (ICMP) is an integral part of the IP protocol
- The best-known use of ICMP:
 - 'ping'
 - 'traceroute'

Routing

- A router copies IP packages between (sub)networks
- Routers compile routing tables to make IP packet forwarding decisions
- Routing and switching functionality may be combined in one device
 - A switch capable of handling routing protocols is also known as a layer 3 switch

Routing Protocols

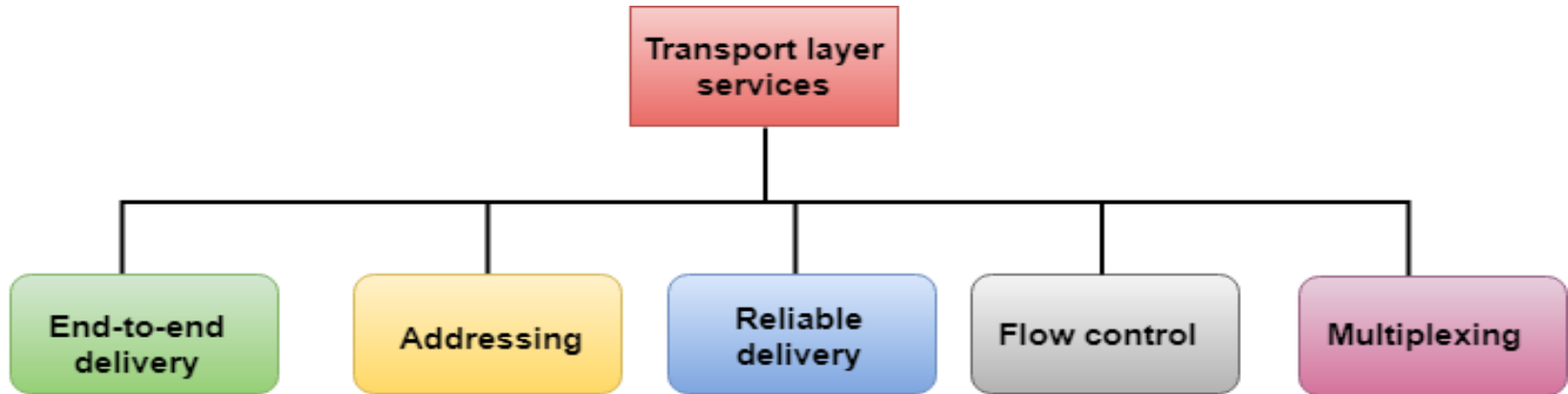
- Dynamic routing protocols automatically create routing tables
 - Based on information exchange with neighboring routers
- When a network connection experiences problems, the routing protocol automatically reconfigures the routing tables to use alternative routes
- LAN and WAN routing protocols can be divided in three classes:
 - Distance vector protocols (like RIP and IGRP)
 - Link state protocols (like OSPF and IS-IS)
 - Path vector routing (like BGP)

Multiprotocol Label Switching

- Multiprotocol Label Switching (MPLS) routes data from one network node to the next with the help of labels
- MPLS allows setting up end-to-end circuit
 - Across any type of physical transport medium
 - Using any protocol
- In practice, MPLS is mainly used to forward IP and Ethernet traffic

Transport Layer

Transport Layer



- The transport layer can maintain flow control, and can provide error checking and recovery of data between network devices
- The most used transport layer protocols are TCP and UDP

TCP

- Transmission Control Protocol (TCP) uses the IP protocol to create reliable transmission of so-called TCP/IP packets
 - TCP provides reliable, ordered delivery of a stream of data between applications
 - TCP introduces much overhead

UDP

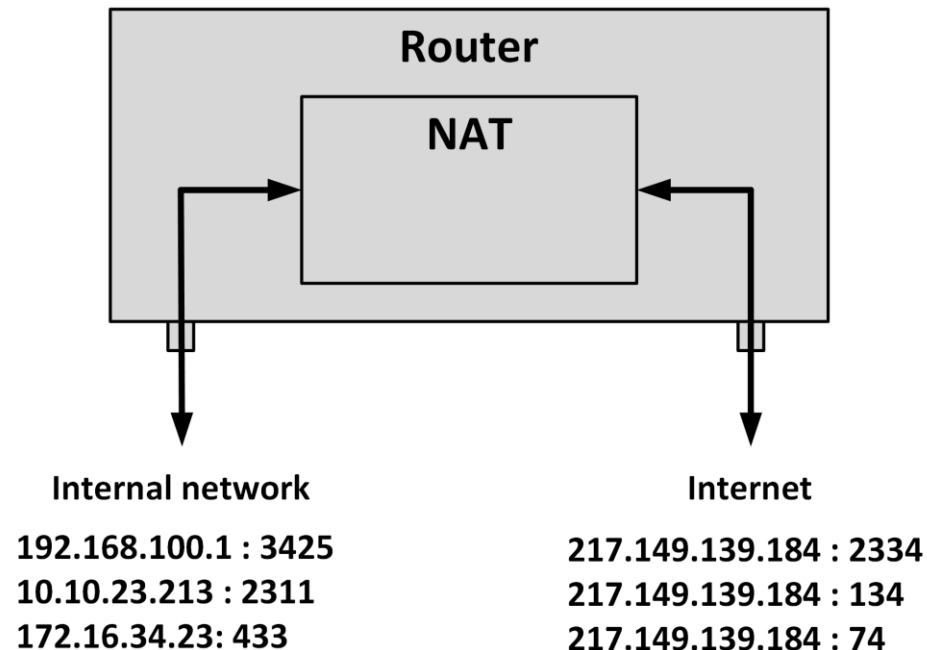
- User Datagram Protocol (UDP) emphasizes reduced latency over reliability
 - It sends data without checking if the data arrived
 - Reduces much overhead
 - UDP is typically used when some packet loss is acceptable
 - Real-time voice and video streams
 - When only small amounts of data are transmitted, that fit in one IP packet

TCP and UDP Ports

- TCP and UDP use logical port numbers
- Each side of a TCP or UDP connection uses an associated port number between 0 and 65,535
- Received TCP or UDP packets are identified as belonging to a specific connection by its combination of the IP address, and the TCP or UDP port number
 - For instance: 192.168.1.2:80, the number after the colon represents the port number (80 in this case)
- Servers running a specific service listen to well-known ports:
 - FTP (port 21)
 - SSH (port 22)
 - SMTP (port 25)
 - DNS (port 53)
 - HTTP (port 80)

Network Address Translation (NAT)

- NAT allows the use of a private addressing space within an organization, while using globally unique addresses for routing data to the internet
- As a packet passes a NAT enabled router from its internal network interface to its internet interface, NAT replaces the packet's private IP address with its public IP address

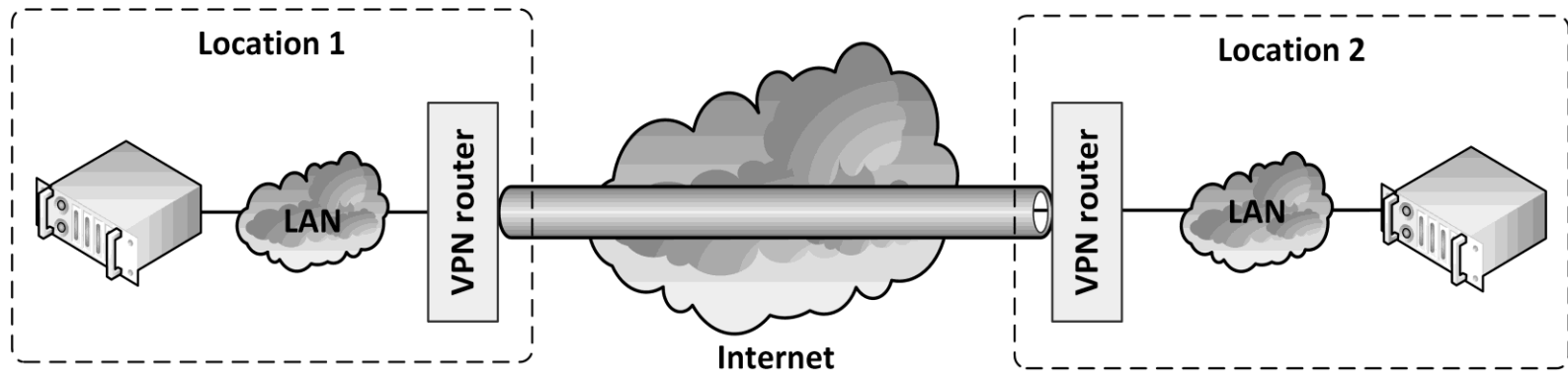


Session Layer

Session Layer

The session layer provides mechanisms for opening, closing and managing a session between end-user application processes

Virtual Private Network (VPN)



- A Virtual Private Network (VPN) uses a public network to interconnect private sites in a secure way
 - Also known as a VPN tunnel
- VPN uses "virtual" connections based on IPsec/SSL
- Most network providers also offer private VPNs based on MPLS

Virtual Private Network (VPN)

- VPNs use strong encryption and strong user authentication
 - Using the internet for transmitting sensitive data is considered safe
- VPN tunnels are often used for remote access to the LAN by users outside of the organization's premises
- Most common VPN communications protocol standards:
 - Point-to-Point Tunneling Protocol (PPTP) for individual client to server connections
 - Layer 2 Tunneling Protocol (L2TP) for individual client to server connections
 - IPsec for network-to-network connectivity
- IPsec is built into IPv6 standard and is implemented as an add-on to IPv4

Presentation Layer

Presentation Layer

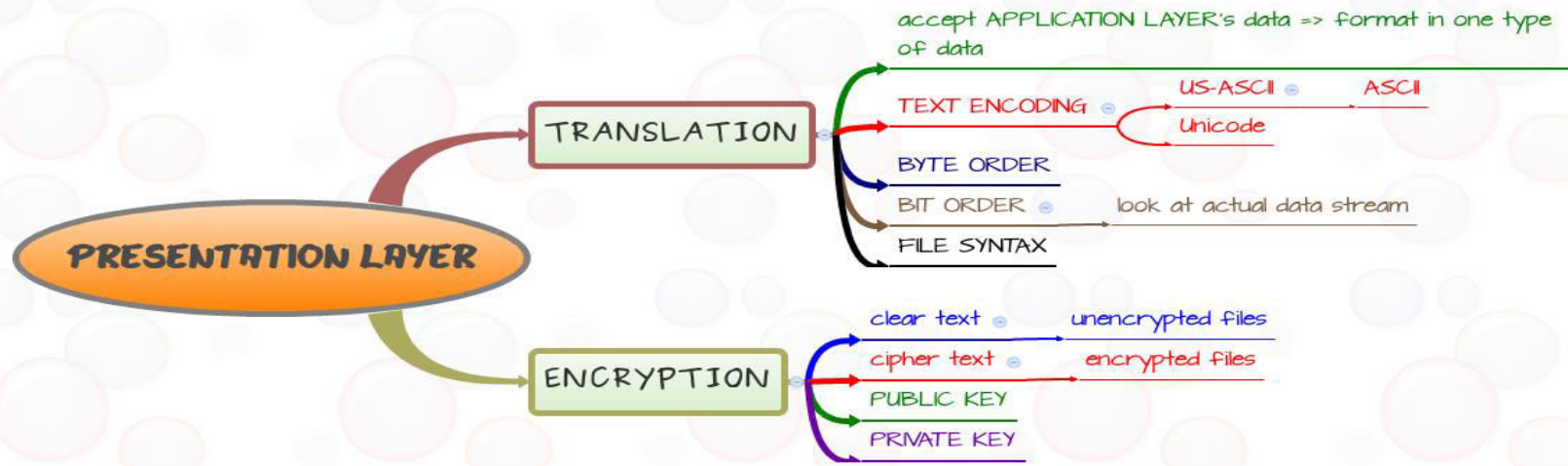


Figure from <http://www.xmind.net/m/WV6b>

- This layer takes the data provided by the application layer and converts it into a standard format that the other layers can understand
- Many protocols are implemented in the presentation layer
 - SSL and TLS are the most important ones

SSL and TLS

- Allow applications to communicate securely over the internet using data encryption
- Secure Sockets Layer (SSL)
 - SSL is considered insecure and should not be used
- Transport Layer Security (TLS)
 - TLS is securing WWW traffic carried by HTTP to form HTTPS
 - Version 1.2 is considered secure
 - Version 1.3 is in a draft state
 - TLS relies on an application capable of handling the protocol (like a Web browser)

Application Layer

Application Layer

- This layer interacts with the operating system or application
- Examples:
 - HTTP
 - FTP
 - SMTP and POP3 (e-mail)
 - CIFS Windows file sharing

Application Layer

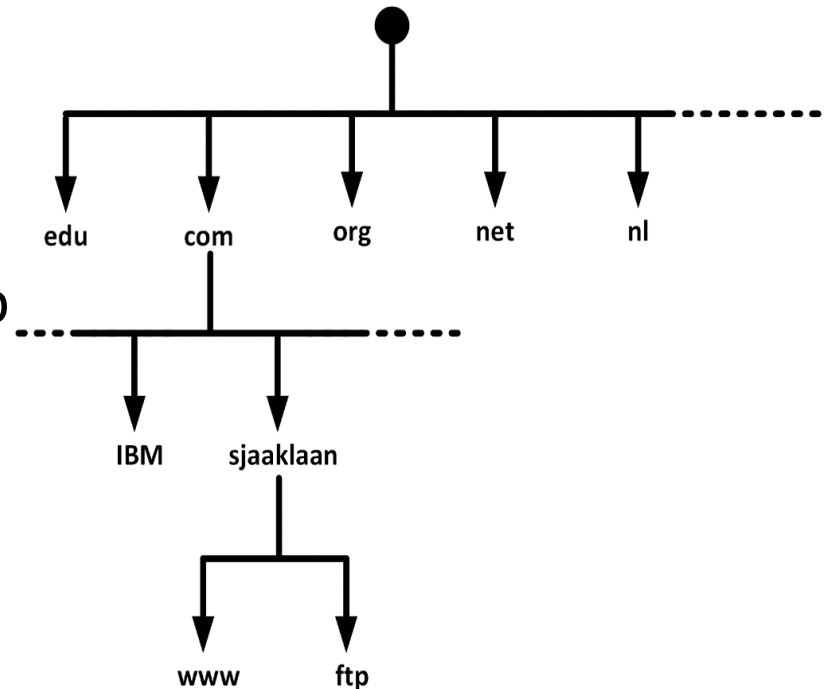
- This layer also contains the relatively simple infrastructure services
- Examples:
 - BOOTP
 - DHCP
 - DNS
 - NTP
- These infrastructure services are used by the infrastructure itself
 - Not necessarily used by upper layer applications
- If infrastructure services fail, usually the entire infrastructure fails!

BOOTP and DHCP

- BOOTP automatically assigns IP addresses to hosts
 - Uses a centralized BOOTP server
 - BOOTP requires manual configuration for each host in the network
- DHCP is an extension to BOOTP
 - It superseded BOOTP because it has more options
- DHCP dynamically assigns network related parameters to hosts:
 - IP addresses
 - Subnet masks
 - Default gateway to be used for routing
 - DNS server to be used
- A DHCP assigned IP address has a limited life span
 - Typically a few hours
 - This is called a lease

DNS

- DNS is a distributed database that links IP addresses with domain names
- Translates domain names, meaningful to humans, into IP addresses
- For example, *www.sjaaklaan.com* is translated to 217.149.139.184
- This IP address is used by the browser to connect to the web server
- DNS distributes the responsibility of mapping domain names to IP addresses by designating authoritative name servers for each domain



DNSSEC

- DNS has a number of security issues
 - DNS was not designed with security in mind
 - Updates to DNS records are done in non-encrypted clear text
 - Authorization is based on IP addresses only
- DNSSEC is a set of extensions to DNS
 - Provides origin authentication of DNS data
 - Provides data integrity
- DNSSEC is not in wide spread use today
 - All DNS servers must implement DNSSEC in order to make full use of all benefits

IPAM Systems

- IP address management (IPAM) systems are appliances that can be used to plan, track, and manage IP addresses in a network
- IPAM systems integrate DNS, DHCP, and IP address administration in one high available redundant set of appliances

Network Time Protocol (NTP)

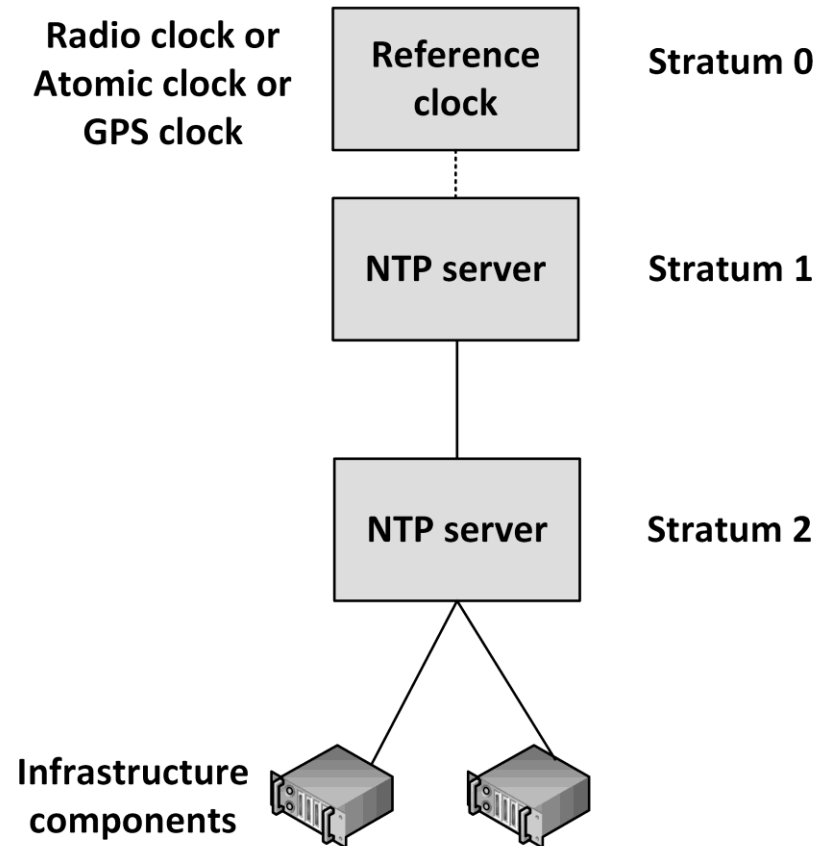
- NTP ensures all infrastructure components use the same time in their real-time clocks
- Particularly important for:
 - Log file analysis
 - Clustering software
 - Kerberos authentication
- NTP can maintain time:
 - To within 10 milliseconds over the internet
 - Accurate to 0.2 milliseconds or better in LANs
- When the time in an operating system is incorrect, the NTP client in the operating system changes the operating system clock

Network Time Protocol (NTP)

- NTP servers can be implemented as:
 - Software on operating systems, routers, and switches
 - Dedicated hardware appliances – often using some external signal like long wave radio clocks or GPS clocks
 - NTP time synchronization services on the internet
- NTP provides time in Coordinated Universal Time (UTC, previously known as GMT)
- The translation to the local time zone, including the switch to and from daylight saving time, is done at the operating system level, not in NTP clocks

Network Time Protocol (NTP)

- NTP operates within a hierarchy
- Each level in the hierarchy is assigned a number called the stratum
- The stratum defines its distance from the reference clock



IT Infrastructure Architecture

Infrastructure Building Blocks
and Concepts

Networking – Part 2
(chapter 8)

Virtualization

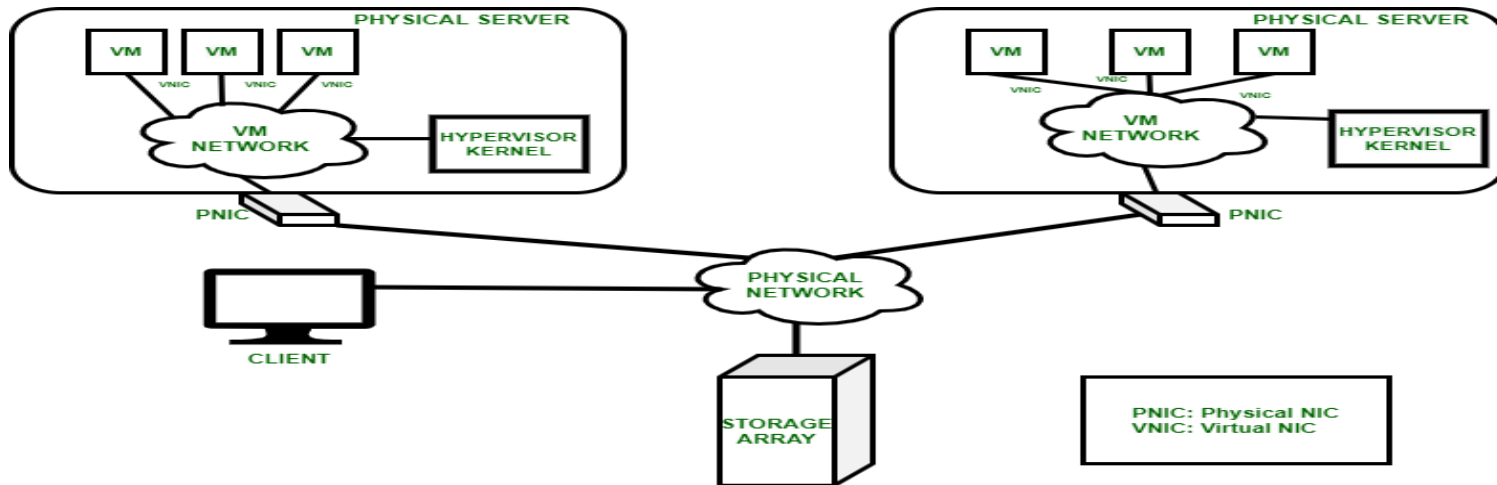
Objectives

- Virtualization
- What is Network Virtualization?
- What are VLANs?
- Virtual Switches
- Software Defined Networking
- Network Function Virtualization

Virtualization

- Virtualization is technology that you can use to create virtual representations of servers, storage, networks, and other physical machines.
- Virtual software mimics the functions of physical hardware to run multiple virtual machines simultaneously on a single physical machine.
- Businesses use virtualization to use their hardware resources efficiently and get greater returns from their investment.
- It also powers cloud computing services that help organizations manage infrastructure more efficiently.

Network Virtualization



<https://www.geeksforgeeks.org/network-virtualization-in-cloud-computing/>

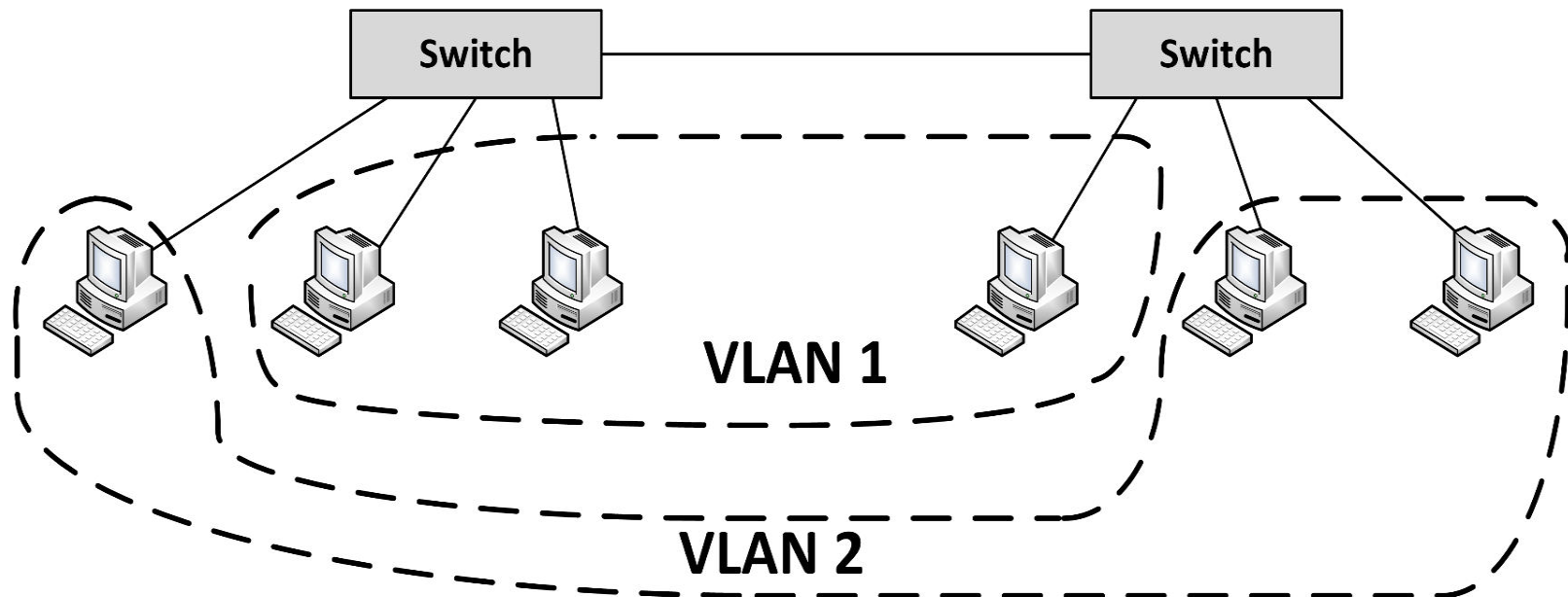
- Network Virtualization (NV) refers to abstracting network resources that were traditionally delivered in hardware to software. NV can combine multiple physical networks to one virtual, software-based network, or it can divide one physical network into separate, independent virtual networks.

Network Virtualization

- Network virtualization decouples network services from the underlying hardware and allows virtual provisioning of an entire network. It makes it possible to programmatically create, provision, and manage networks all in software, while continuing to leverage the underlying physical network as the packet-forwarding backplane. Physical network resources, such as switching, routing, firewalling, load balancing, virtual private networks (VPNs), and more, are pooled, delivered in software, and require only Internet Protocol (IP) packet forwarding from the underlying physical network.

Virtual LAN (VLAN)

- VLANs enable logical grouping of network nodes on the same LAN
 - Configured on network switches
 - Operate at the Ethernet level



Virtual LAN (VLAN)

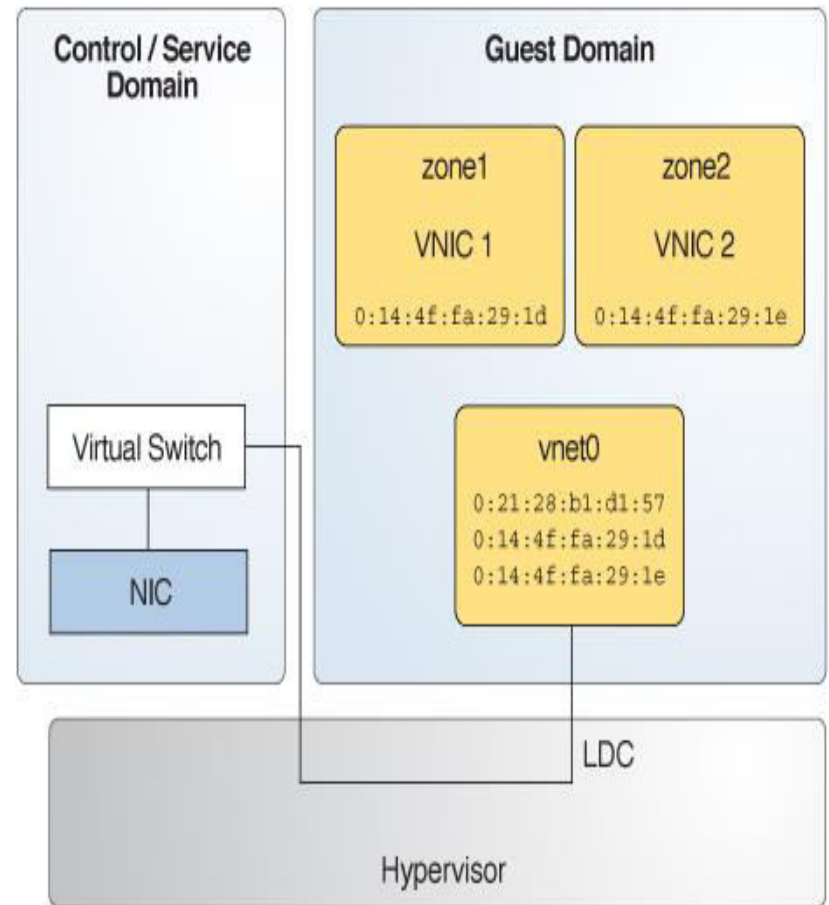
- VLANs:
 - Allow segmenting a network at the data link layer
 - Allow end stations to be grouped together even if they are not physically connected to the same switch
 - Can adapt to changes in network requirements and allow simplified administration
 - Enhance security by preventing traffic in one VLAN from being seen by hosts in a different VLAN
- For VLANs to communicate with each other a **router** is needed

VXLAN

- Virtual Extensible LAN (VXLAN) is an encapsulation protocol
- Can be used to create a logical switched layer 2 network across routed layer 3 networks
- Only servers within the same logical network can communicate with each other
- VXLANs are heavily used in multi-tenant cloud environments

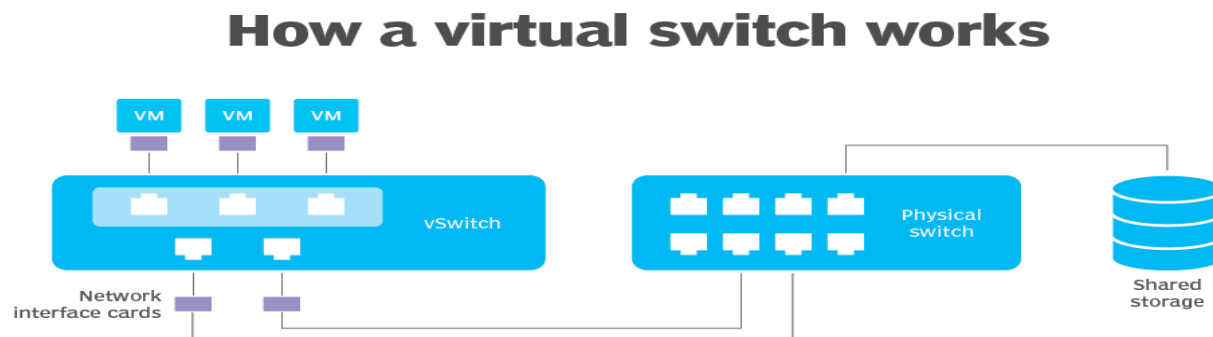
Virtual NICs

- Virtual machines are only aware of virtual Network Interface Controllers (NICs) provided to them
- Virtual machines running on physical machines share physical NICs
- Communications between virtual machines on the same physical machine are routed directly in memory space by the hypervisor, without using the physical NIC
- The hypervisor routes Ethernet packages from the virtual NIC on the virtual machine to the physical NIC on the physical machine



Virtual Switch

- Virtual NICs are connected to virtual switches
- A virtual switch is an application running in the hypervisor, with most of the capabilities of a physical network switch
- A virtual switch is dynamically configured
 - Ports in the virtual switch are configured at runtime
 - The number of ports on the switch is in theory unlimited



Virtual Switch

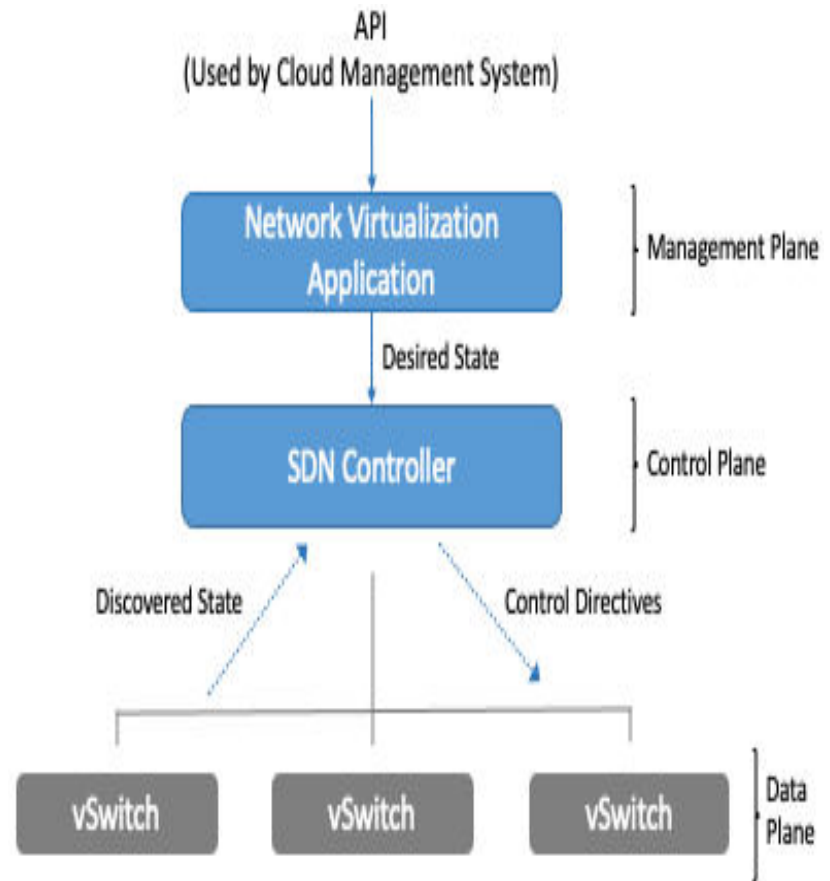
- No cable disconnects
- No need for auto-detecting network speed
- Availability:
 - No network hubs, routers, adapters, or cables that could physically fail
- Security:
 - No easy way to intercept network communications between virtual machines from outside of the physical machine

Software Defined Networking

- Software Defined Networking (SDN) allows networks to be defined and controlled using software external to the physical networking devices
- A set of physical network switches can be programmed as a virtual network:
 - Hierarchical
 - Complex
 - Secured
- A virtual network can easily be changed without touching the physical network components

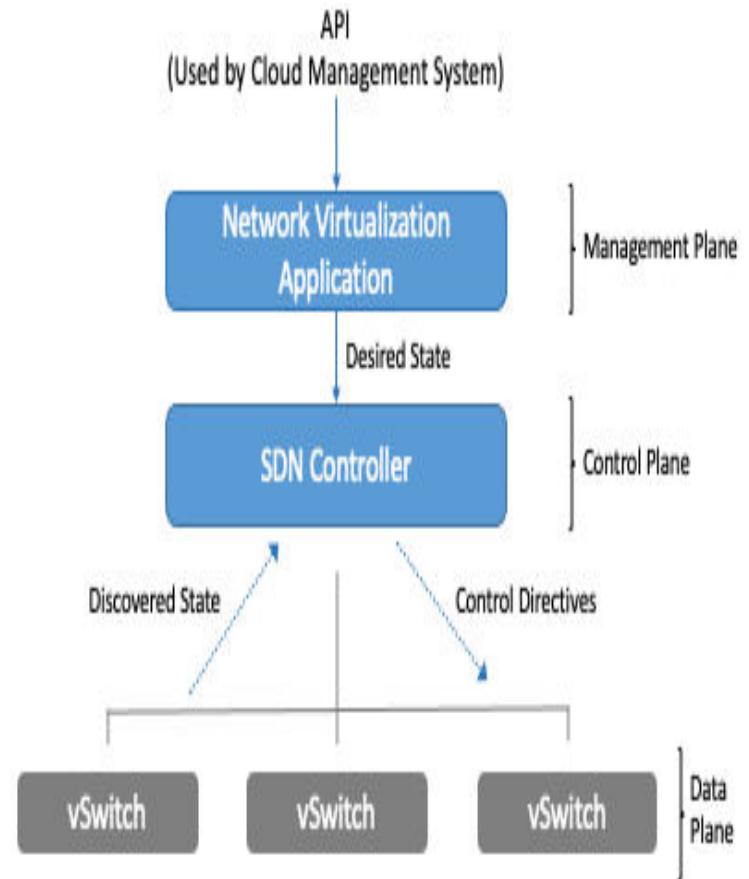
How does SDN work?

- In SDN (like anything virtualized), the software is decoupled from the hardware.
- SDN moves the **control plane** that determines where to send traffic to software, and leaves the data plane that actually forwards the traffic in the hardware.
- This allows network administrators who use software-defined networking to program and control the entire network via a single pane of glass instead of on a device by device basis.



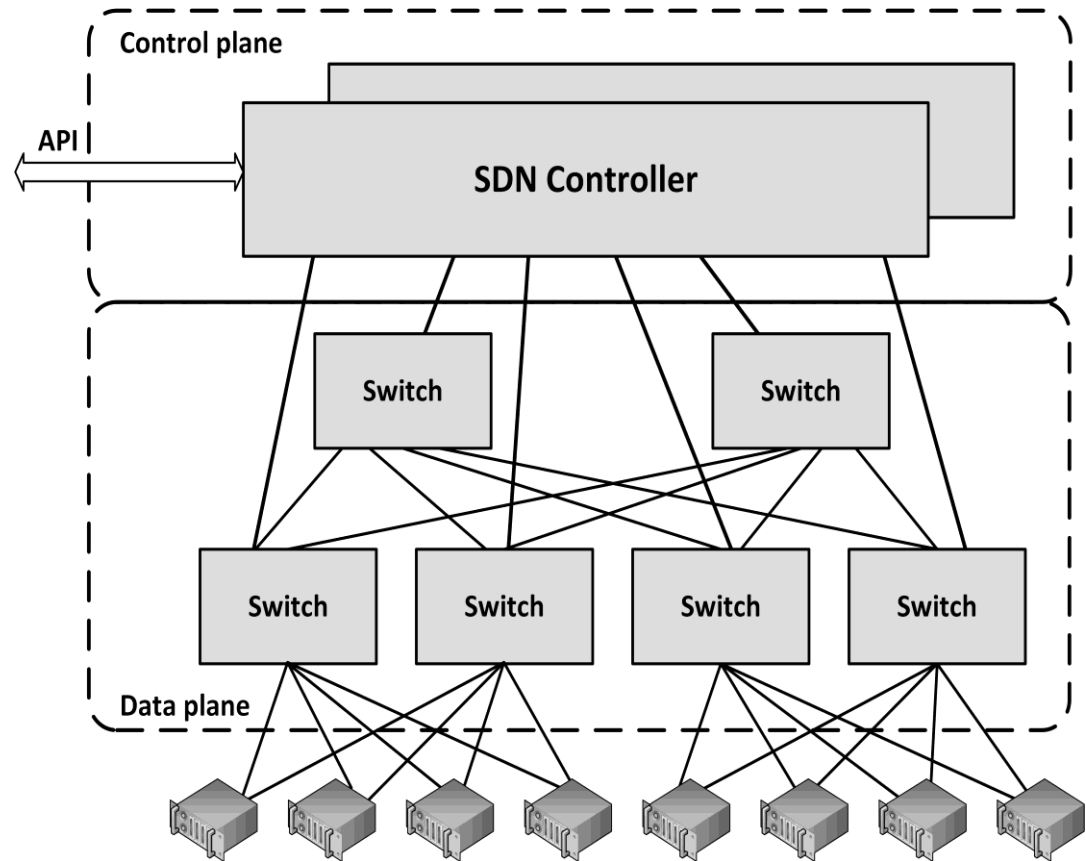
Software Defined Networking

- SDN can be controlled from a single management console
- Provides open APIs that can be used to manage the network using third party software
- In an SDN, the distributed data plane devices are only forwarding network packets based on ARP or routing rules that are preloaded into the devices by the SDN controller in the control plane
 - This allows the physical devices to be much simpler and more cost effective



Software Defined Networking

- Control plane resides centrally
- Data plane (the physical switches) remain distributed



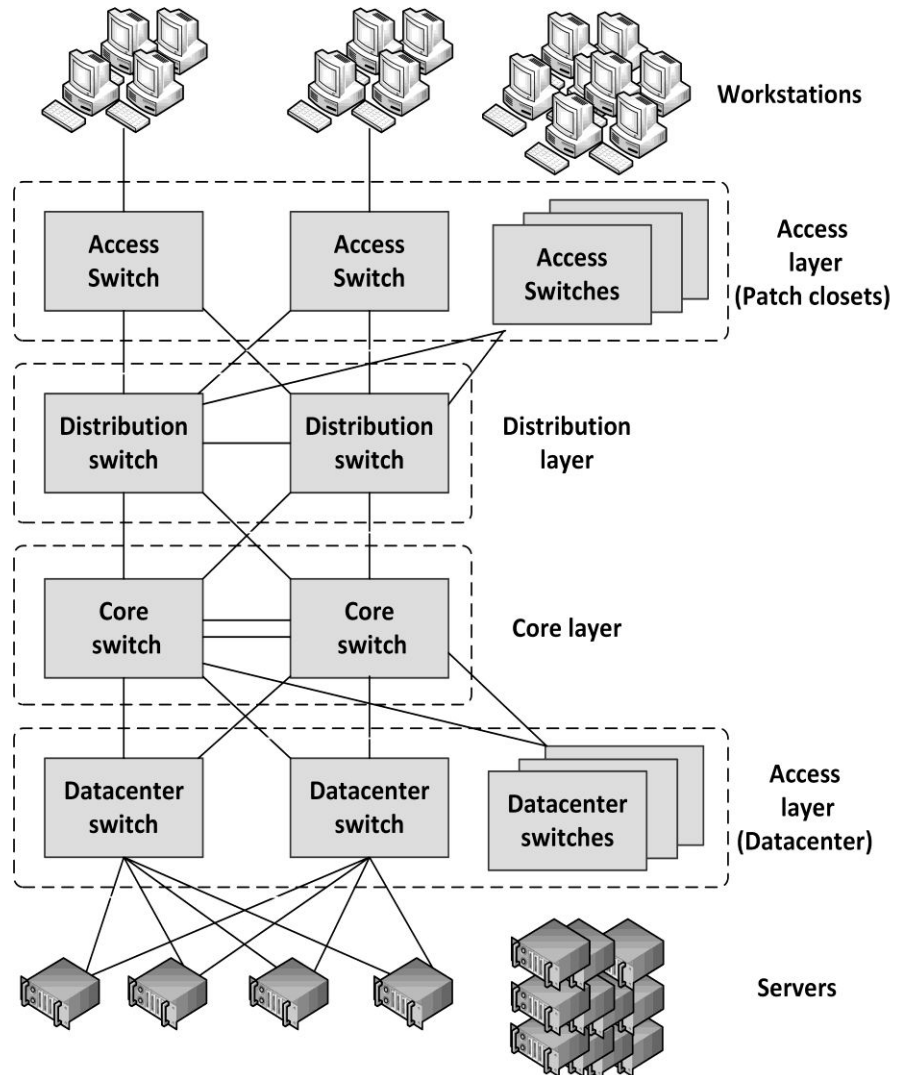
Network Function Virtualization

- Network Function Virtualization (NFV) is a way to virtualize networking devices
 - Firewalls
 - VPN gateways
 - Load balancers
- NFV appliances are implemented as virtual machines running applications that perform the network functions
- NFV virtual appliances can be created and configured dynamically and on-demand using APIs
- Example:
 - Deploy a new firewall as part of a script that creates a number of connected virtual machines in a cloud environment

Network Availability

Layered Network Topology

- A network infrastructure should be built up in layers
 - Improve availability and performance
 - Provides scalability
 - Provides deterministic routing
 - Avoids unmanaged ad-hoc data streams
- Provides high availability
 - Because the layering provides multiple paths to any piece of equipment

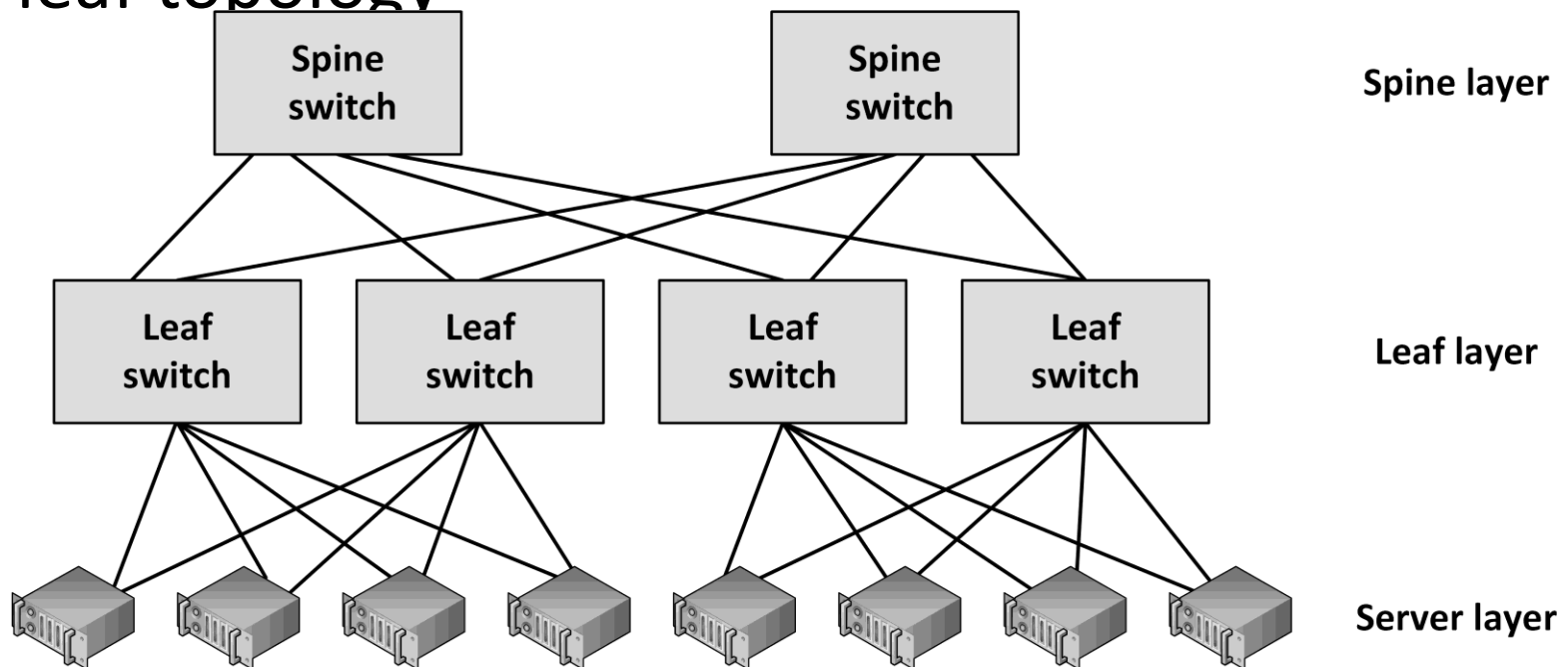


Layered Network Topology

- Core layer
 - This is the center of the network
- Distribution layer
 - An intermediate layer between the core layer in the datacenter and the access switches in the patch closets
 - Combines the access layer data and sends its combined data to one or two ports on the core switches
- Access layer
 - Connect workstations and servers to the distribution layer
 - For servers, located at the top of the individual server racks or in blade enclosures
 - For workstations, placed in patch closets in various parts of the building

Spine and Leaf Topology

- In a SDN, a simple physical network is used that can be programmed to act as a complex virtual network
- Such a network can be organized in a spine and leaf topology



Spine and Leaf Topology

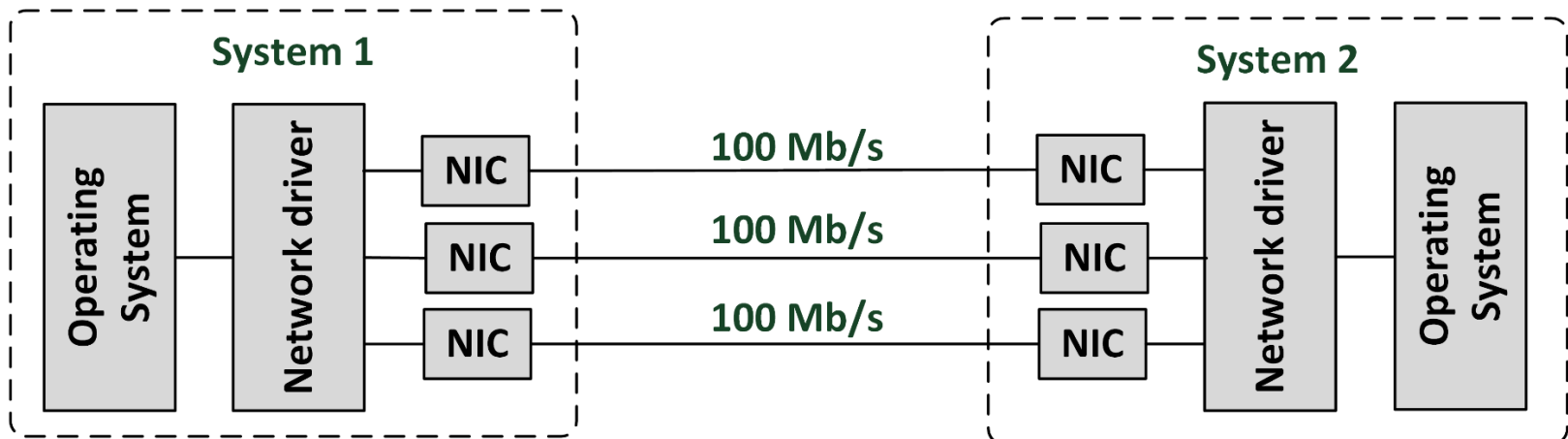
- Characteristics:
 - The spine switches are not interconnected
 - Each leaf switch is connected to all spine switches
 - Each server is connected to two leaf switches
 - The connections between spine and leaf switches typically have ten times the bandwidth of the connectivity between the leaf switches and the servers

Spine and Leaf Topology

- Benefits:
 - Highly scalable
 - There are no interconnects between the spine switches
 - Simple to scale
 - Just add spine or leaf servers
 - With today's high density switches, many physical servers can be connected using relatively few switches
 - Each server is always exactly four hops away from every other server
 - Leads to a very predictable latency

Network Teaming

- Network teaming is also known as:
 - Link aggregation
 - Port trunking
 - Network bonding
- Provides a virtual network connection using multiple physical cables for high availability and increased bandwidth



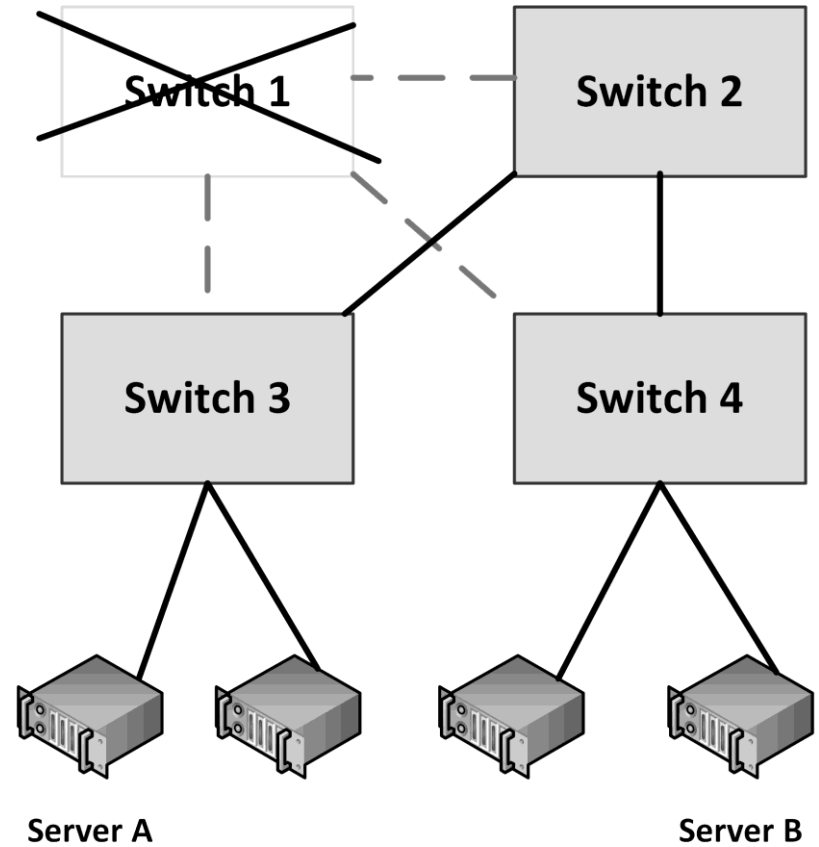
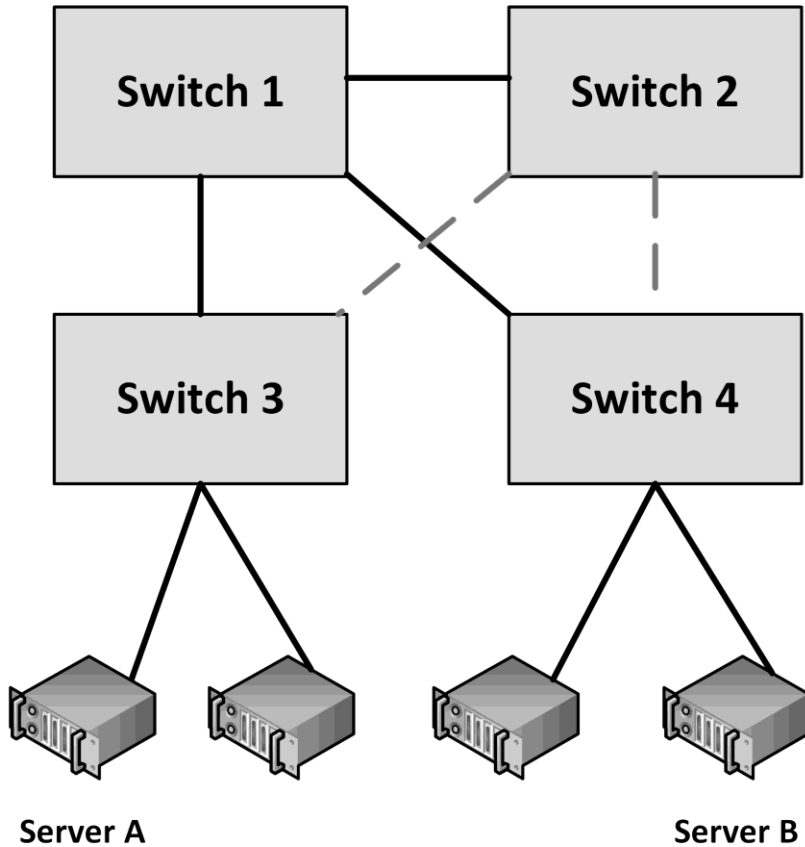
Network Teaming

- Network teaming bonds physical NICs together to form a logical network team
 - Sends traffic to the team's destination to all NICs in the team
 - Allows a single NIC, cable, or switch to be unavailable without interrupting traffic

Spanning Tree Protocol (STP)

- STP is an Ethernet level protocol that runs on switches
- Guarantees that only one path is active between two network endpoints at any given time
- Redundant paths are automatically activated when the active path experiences problems
- Ensures no loops are created when redundant paths are available in the network
- A disadvantage of using the spanning tree protocol is that it is not using half of the network links in a network, since it blocks redundant paths
- Rapid Spanning Tree Protocol (RSTP) provides for fast spanning tree convergence after a topology change (6 s instead of 30-60 s)

Spanning Tree Protocol



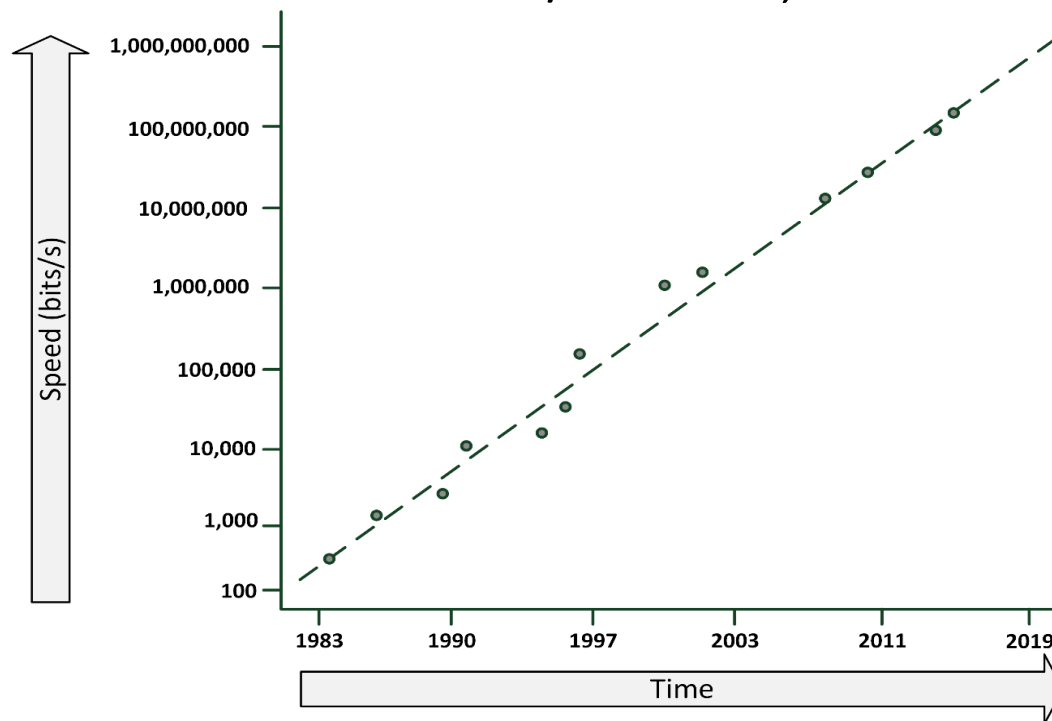
Multihoming

- Connecting a network to two different Internet Service Providers (ISPs) is called multihoming
- Four options for multihoming:
 - Single router with **dual links** to a single ISP
 - Single router with dual links to **two separate** ISPs
 - **Dual routers**, each with its own link to a single ISP
 - Dual routers, each with its own link to a **separate ISP**
- It is not always guaranteed that multiple network paths actually run on a different set of cables
 - WAN cables are typically installed alongside highways and railway tracks
 - Cables are used by multiple carrier providers

Network Performance

Nielsen's Law

- Network connection speeds for high-end home users increase 50% per year, they double every 21 months
- Bandwidths should be 15 Gbit/s in 2025, for about \$50 per month



Please note that the vertical scale is logarithmic instead of linear

Throughput and Bandwidth

- Throughput is the amount of data that is transferred through the network during a specific time interval
- Throughput is limited by the available bandwidth
- When an application requires more throughput than a network connection can deliver:
 - Queues in the network components temporarily buffer data
 - Buffered data is sent as soon as the network connection is free again
 - When more data arrives than the queues can store in the buffer, packet loss occurs

Latency

- Latency is defined as the time from the start of packet transmission to the start of packet reception
- Latency is dependent on:
 - The physical distance a packet has to travel
 - The number of switches and routers the packet has to pass
- Rules of thumb:
 - 6 ms latency per 100 km
 - WANs: Each switch in the path adds 10 ms to the one-way delay
 - LANs: add 1 ms for each switch

Latency

- One-way latency: the time from the source sending a packet to the destination receiving it
- Round-trip latency: the one-way latency from source to destination plus the one-way latency from the destination back to the source
- “ping” can be used to measure round-trip latency

Quality of Service (QoS)

- Quality of service (QoS) is the ability to provide different data flow priority to different applications, users, or types of data
- QoS allows better service to certain important data flows compared to less important data flows
- QoS is mainly used for real-time applications like video and audio streams and VoIP telephony

Quality of Service (QoS)

- Four basic ways to implement QoS:
 - Congestion management
 - Defines what must be done if the amount of data to be sent exceeds the bandwidth of the network link
 - Packets can either be dropped or queued
 - Queue management
 - When queues are full, packets will be dropped
 - Queue management defines criteria for dropping packets that are of lower priority before dropping higher priority packets

Quality of Service (QoS)

– Link efficiency

- Ensures the link is used in an optimized way
- For instance by fragmenting large packets with a low QoS, allowing packets with a high QoS to be sent between the fragments of low QoS packets

– Traffic shaping

- Limiting the full bandwidth of streams with a low QoS to benefit streams with a high QoS
- High QoS streams have a reserved amount of bandwidth

WAN Link Compression

- Data compression reduces the size of data before it is transmitted over a WAN connection
- WAN acceleration appliances:
 - Provide compression
 - Perform some caching of regularly used data at remote sites

Network Security

Network Security

Network security is a set of technologies that protects the usability and integrity of a company's infrastructure by preventing the entry or proliferation within a network of a wide variety of potential threats.

A network security architecture is composed of tools that protect the network itself and the applications that run over it. Effective network security strategies employ multiple lines of defense that are scalable and automated. Each defensive layer enforces a set of security policies determined by the administrator.

Why Network Security?

In today's hyper-connected world, network security presents a greater challenge as more business applications move to private and public clouds.

Moreover, the applications themselves now tend to be virtualized and distributed across many locations, some of which are outside the physical control of IT security teams. With the number of attacks on companies climbing ever higher, protecting network traffic and infrastructure is critical.

Firewalls

- Firewalls separate two or more LAN or WAN segments for security reasons
- Firewalls block all unpermitted network traffic between network segments
- Permitted traffic must be explicitly enabled by configuring the firewall to allow it
- Firewalls can be implemented:
 - In hardware appliances
 - As an application on physical servers
 - In virtual machines
- Host based firewalls
 - Protect a server or end user computer against network based attacks
 - Part of the operating system

Firewalls

- Firewalls use one or more of the following methods to control traffic:
 - Packet filtering
 - Data packets are analyzed using preconfigured filters
 - This functionality is almost always available on routers and most operating systems
 - Proxy (also known as application layer firewalls)
 - A proxy terminates the session on the application level on behalf of the server (proxy) or the client (reverse proxy) and creates a new session to the client or server
 - Stateful inspection
 - Inspects the placement of each individual packet within a packet stream
 - Maintains records of all connections passing through the firewall and determines whether a packet is the start of a new connection, part of an existing connection, or is an invalid packet

IDS/IPS

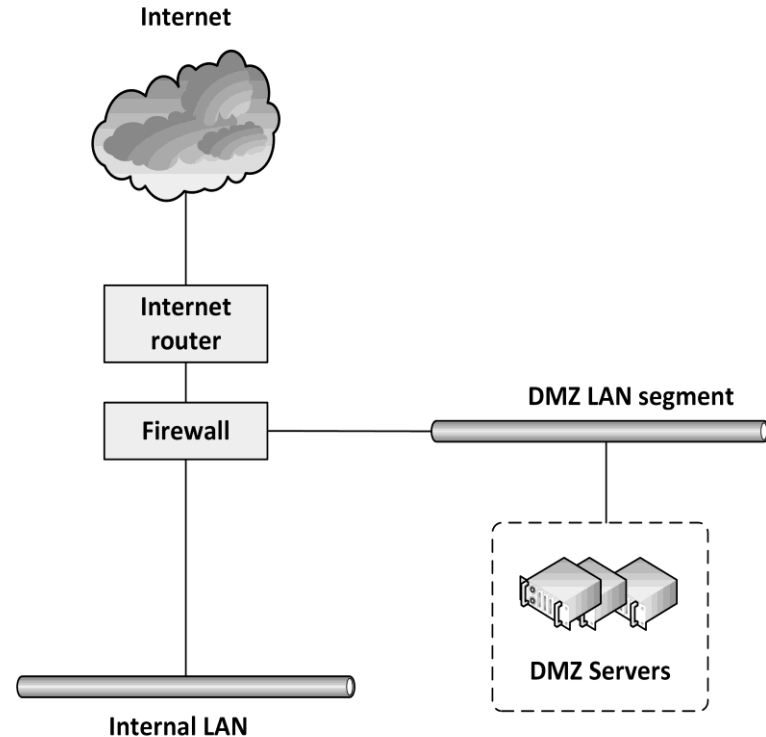
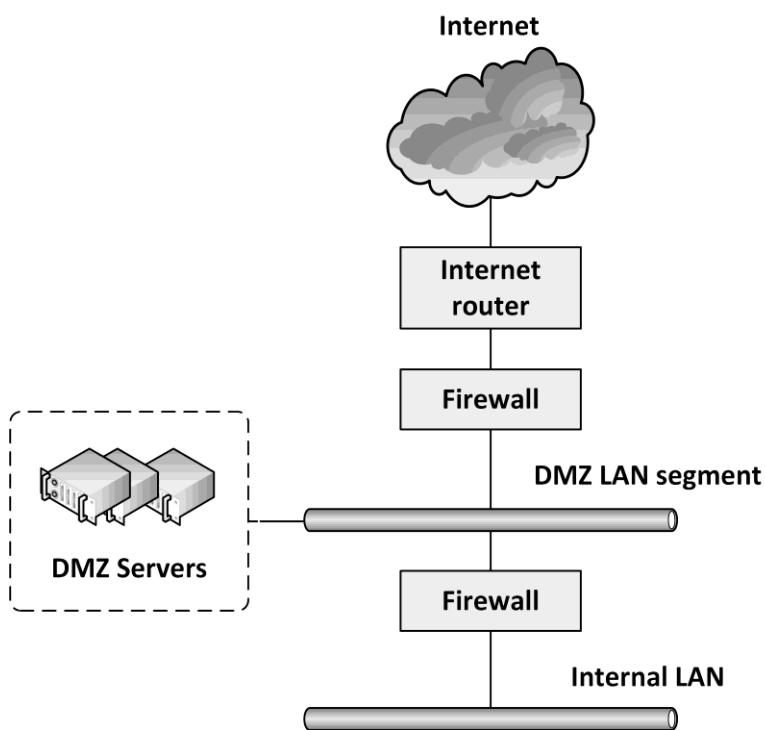
- An Intrusion Detection System (IDS) or Intrusion Prevention System (IPS) detects and – if possible – prevents activities that compromise system security, or are a hacking attempt
- An IDS/IPS monitors for suspicious activity and alerts the systems manager when these activities are detected
- An IPS can stop attacks by changing firewall rules on the fly

IDS/IPS

- Two types of IDS/IPS systems:
 - A Network-based IDS (NIDS) is placed at a strategic point in the network
 - Monitors traffic to and from all devices on that network
 - The NIDS is not part of the network flow, but just “looks at it”, to avoid detection of the NIDS by hackers
 - A Host-based IDS (HIDS) runs on individual servers or network devices
 - It monitors the network traffic of that device
 - It also monitors user behavior and the alteration of critical (system) files

DMZ

- DMZ is short for De-Militarized Zone, also known as screened subnet, or the Perimeter Network
- A DMZ is a network that serves as a buffer between a secure protected internal network and the insecure internet



RADIUS

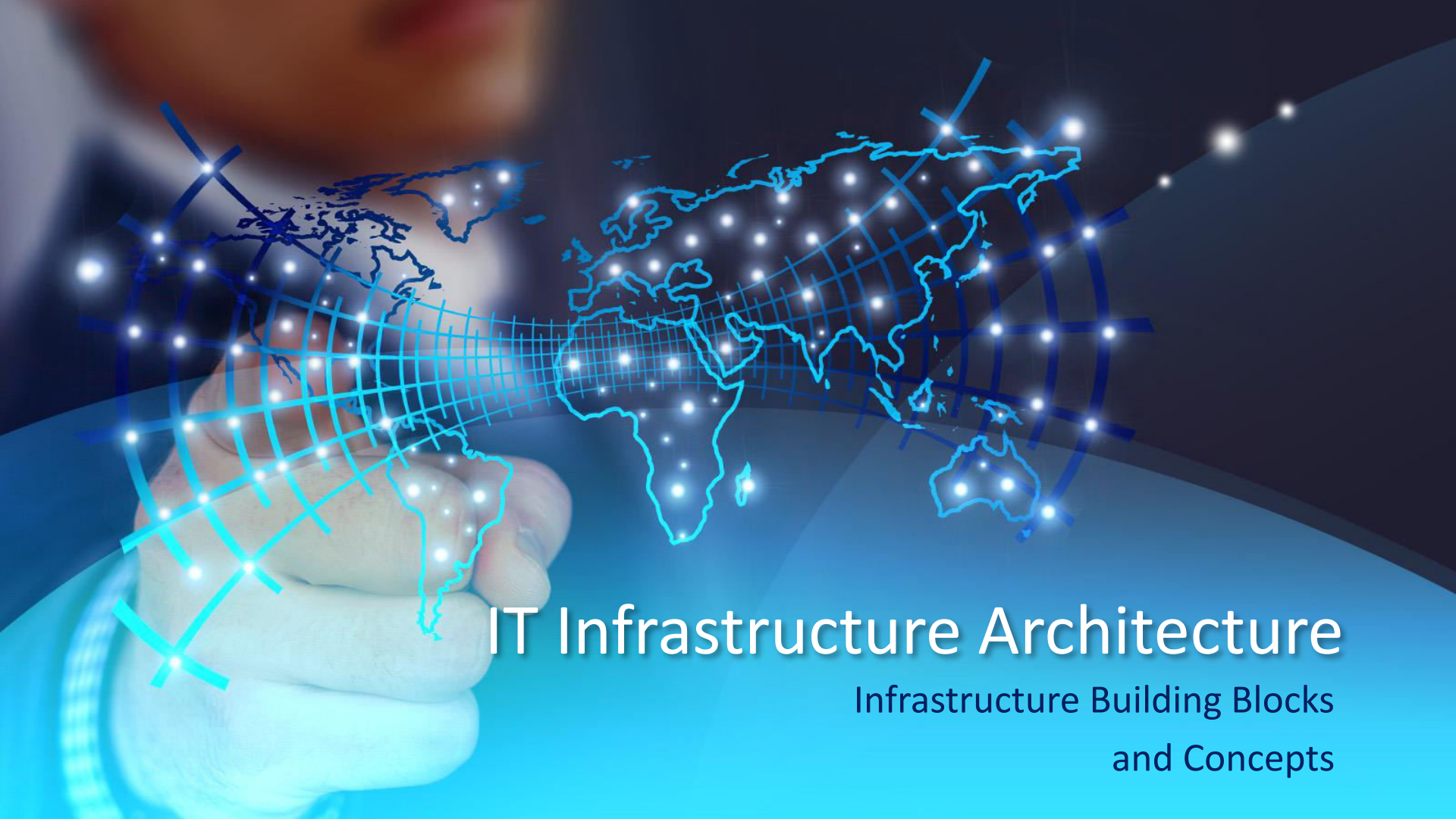
- Remote Authentication Dial In User Service (RADIUS) is a networking protocol that provides centralized user and authorization management for network devices
 - Routers
 - Modem servers
 - Switches
 - VPN routers
 - Wireless network access points
- RADIUS
 - Authenticates users or devices before granting them access to a network
 - Authorizes users or devices for certain network services

Network Access Control (NAC)

- Network Access Control (NAC) is used at the network end points, where end user devices (like laptops) can be connected to the network
- It allows predefined levels of network access based on:
 - A client's identity (is the laptop known to the organization?)
 - The groups to which a client belongs
 - The degree to which a client's device complies with the organization's governance policies (does it run the most recent virus scanner?)

Network Access Control (NAC)

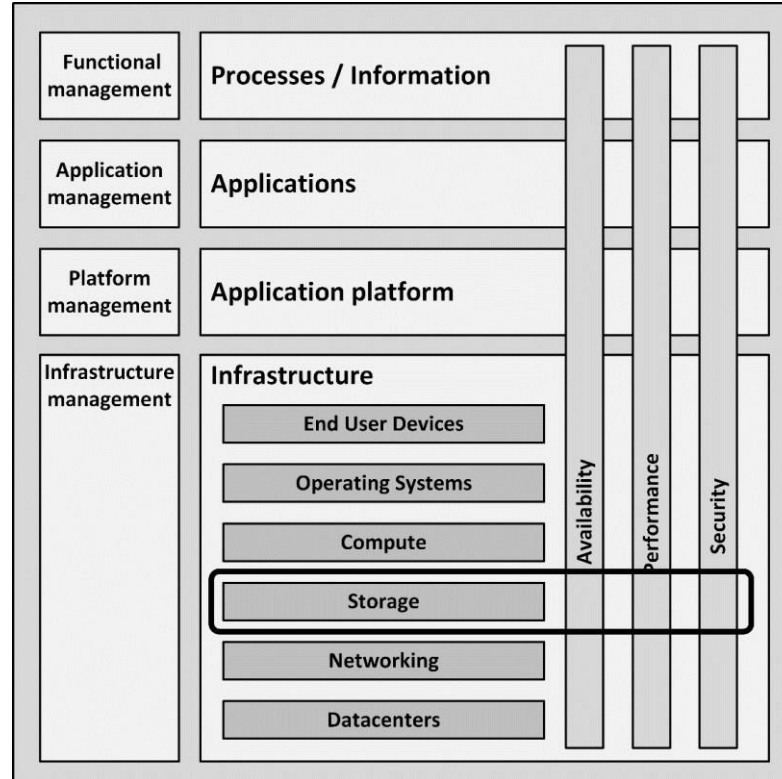
- If a client device is not compliant, NAC provides a mechanism to automatically bring it into compliance
- For instance:
 - Installing the latest virus scanner updates while connected on an isolated LAN segment
 - After the update finishes, access is granted to the rest of the network

A hand holding a glowing globe with a grid of lines and stars, symbolizing global connectivity and technology.

IT Infrastructure Architecture

Infrastructure Building Blocks
and Concepts

Storage -Part-1



Introduction

- Each day 15 petabytes of new information is generated worldwide
- The total amount of digital data doubles approximately every two years



History

- Early computers used punched cards or paper tape as storage



Paper Tape

History - Drum Memory

- Drum memory was one of the first magnetic read/write storage systems
- Widely used in the 1950s and 1960s
- Large rotating metal cylinder coated with magnetic recording material
- Multiple rows of fixed read-write heads
- Each head reading or writing to one track
- Store 62 kB of data



History – Hard Disks

- The first commercial digital disk storage device was part of the IBM RAMAC 350 system, shipped in 1956
 - Approximately 5 MB of data
 - Fifty 61 cm diameter disks
 - Weighed over a ton
- Over the years:
 - Physical size of hard disks shrunk
 - Magnetic density increased
 - Rotation speed increased from 3,600 rpm to 15,000 rpm
 - Seek times lowered as a result of using servo controlled read/write heads instead of stepper motors



History – Tapes

- The IBM 726, introduced in 1952, was one of the first magnetic tape systems
 - 2 MB per 20-centimeter-diameter reel of tape
- Reel tapes were used until the late 1980s, mostly in mainframes
- In 1984, DEC introduced the Digital Linear Tape (DLT)
 - Super DLT (SDLT) tape cartridges can store up to 300 GB of data
- Linear Tape Open (LTO) was originally developed in the late 1990s
 - LTO version 7 was released in 2015 and can hold up to 6 TB of data





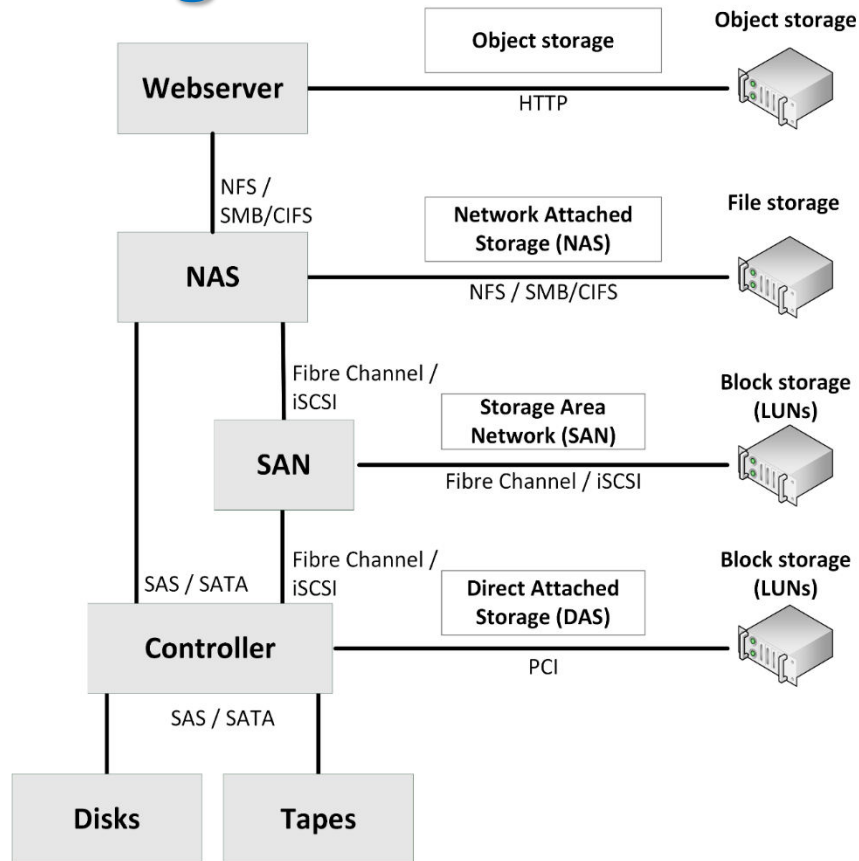
STORAGE BUILDING BLOCKS



Storage Model

- Most servers use external storage, sometimes combined with internal storage

Storage Model





Disks – Command Sets

- Disks use ATA or SCSI command sets to connect to controllers.

Advanced Technology Attachment (ATA),

- Also known as IDE,
- Uses relatively simple hardware and communication protocol

Small Computer System Interface (SCSI)

- Set of standards for physically connecting and transferring data between computers
- The SCSI command set is complex - there are about 60 different SCSI commands in total

Mechanical Hard Disks

- Mechanical disks consist of:
 - ✓ A vacuum sealed case
 - ✓ One or more spinning magnetic disks on one spindle
 - ✓ A number of read/write heads that can move to reach each part of the spinning disks



Mechanical Hard Disks

- **Serial ATA (SATA) disks**
 - Low-end high-capacity disks
 - Ideal for bulk storage applications (like archiving or backup)
 - Have a low cost per gigabyte
 - Often used in PCs and laptops
 - Use the SMART command set to control the disk



Mechanical Hard Disks

- **Serial Attached SCSI (SAS) disks**
 - Relatively expensive
 - High end disks
 - Spinning disk platters with a rotational speed of 10,000 or 15,000 rpm
 - Typically have 25% of the capacity of SATA or NL-SAS disks
 - Uses the SCSI command set that includes error-recovery and error-reporting and more functionality than the SMART commands used by SATA disks



Mechanical Hard Disks

- **Near-Line SAS (NL-SAS) disks**
 - Have a SAS interface, but the mechanics of SATA disks
 - Can be combined with faster SAS disks in one storage array





Solid State Drives (SSDs)

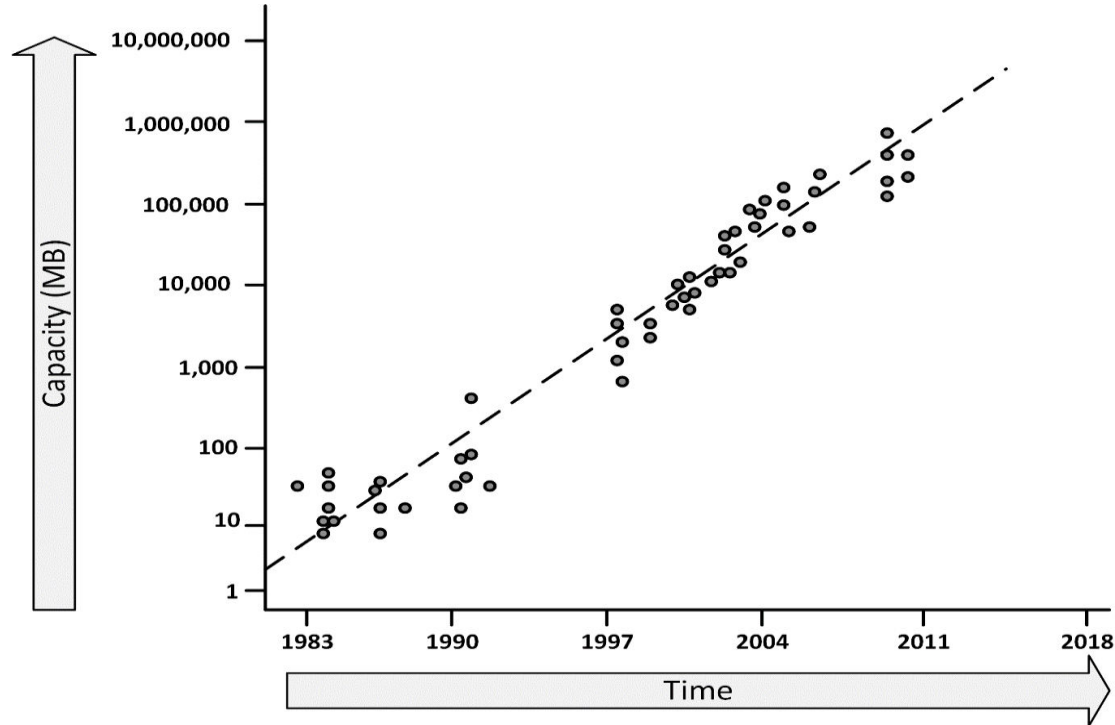
- Based on flash technology
- Connected using a standard SAS disk interface
- Consume less power, and therefore generate less heat
- No moving parts and no vibration
- Data can be accessed much faster than using mechanical disks



Disk Capacity - Kryder's Law

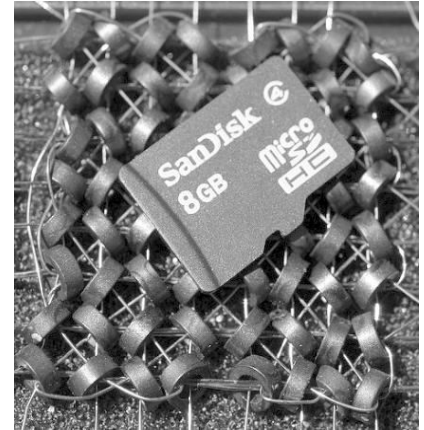
- The density of information on hard drives doubles every 13 months
- An average single disk drive in 2025 will hold more than 20,000 TB (20 PB) of data

Disk Capacity - Kryder's Law



Disk Capacity - Kryder's Law

- The picture on the right shows 8 bytes core memory and 8 GB SD flash memory
 - An increase of 1,000,000,000 times in 50 years
- To have full benefits of Kryder's law, the storage infrastructure should be designed to handle just in time expansion





Tapes

- When storing large amounts of data, tape is the most inexpensive option
- Tapes are suitable for archiving
 - Tape manufacturers guarantee a long life expectancy
 - DLT, SDLT, and LTO cartridges are guaranteed to be readable after 30 years on the shelf



Tapes

Disadvantages:

- Tapes are fragile
 - Manual handling can lead to mechanical defects:
 - Tapes dropping on the floor
 - Bumping
 - Bad insertions of tapes in tape drives
- Tape cartridges contain mechanical parts
 - Manually changed tapes get damaged easily
- Frequent rewinding causes stress to the tape substrate
 - Leads to lower reliability of data reads



Tapes

- Tapes are extremely slow
 - They only write and read data sequentially
 - When a particular piece of data is required, it must be searched by reading all data on tape until the required data is found
 - Together with rewinding of the tape (needed for ejecting the tapes) handling tapes is expressed in minutes instead of in milliseconds or microseconds

Tape Library

- Tape libraries can be used to automate tape handling
- A tape library is a storage device that contains:
 - One or more tape drives
 - A number of slots to hold tape cartridges
 - A barcode or RFID tag reader to identify tape cartridges
 - An automated method for loading tapes





Virtual Tape Library

- A Virtual Tape Library (VTL) uses disks for storing backups
- A VTL consists of:
 - An appliance or server
 - Software that emulates traditional tape devices and formats
- VTLs combine high performance disk based backup and restore with well-known backup applications, standards, processes, and policies
- Most of the current VTL solutions use NL-SAS or SATA disk arrays because of their relatively low cost
- They provide multiple virtual tape drives for handling multiple tapes in parallel

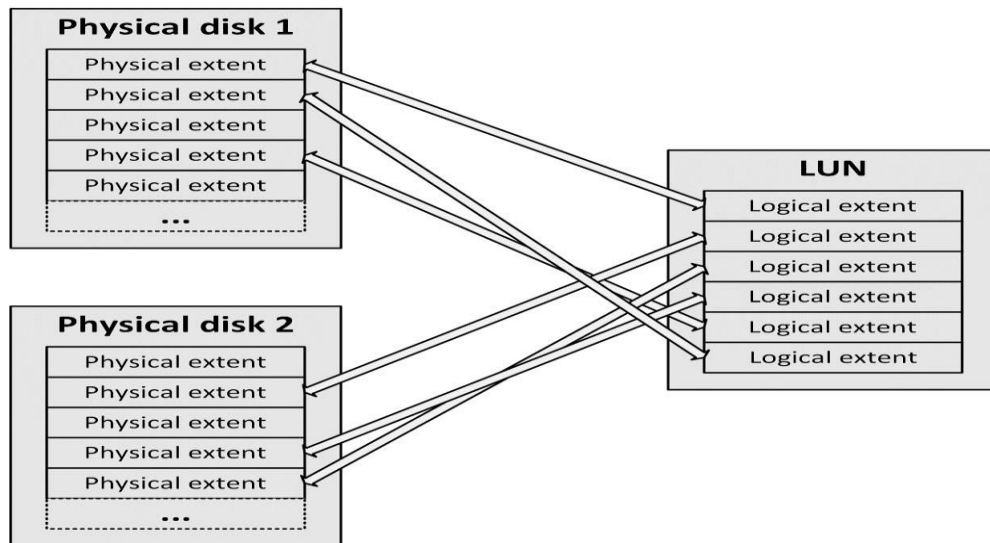


Controllers

- Controllers connect disks and/or tapes to a server, in one of two ways:
 - Implemented as a PCI expansion boards in the server
 - Part of a NAS or SAN deployment, where they connect all available disks and tapes to redundant Fibre Channel, iSCSI, or FCoE connections

Controllers

- Splits all disks in small pieces called physical extents
- From these physical extents, new virtual disks (Logical Unit Numbers - LUNs) are composed and presented to the operating system





RAID

- Redundant Array of Independent Disks (RAID) solutions provide:
 - High availability of data
 - Improvements of performance
- RAID uses multiple redundant disks
- RAID can be implemented:
 - In the disk controller's hardware
 - As software running in a server's operating system

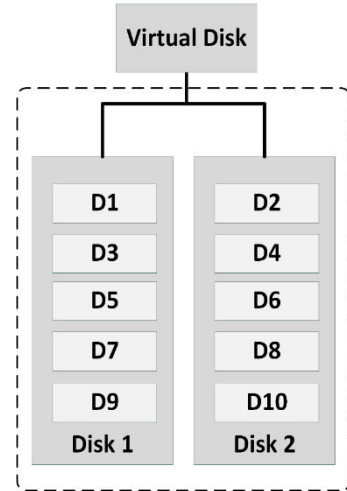


RAID

- RAID can be implemented in several configurations, called RAID levels
- In practice, five RAID levels are implemented most often:
 - **RAID 0** - Striping
 - **RAID 1** - Mirroring
 - **RAID 10** - Striping and Mirroring
 - **RAID 5** - Striping with distributed parity
 - **RAID 6** - Striping with distributed double parity

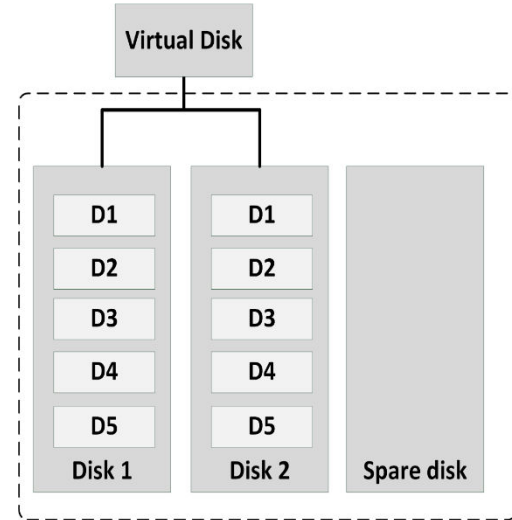
RAID 0 - Striping

- RAID 0 is also known as striping
- Provides an easy and cheap way to increase performance
- Uses multiple disks, each with a part of the data on it
- RAID 0 actually lowers availability
 - If one of the disks in a RAID 0 set fails, all data is lost
- Only acceptable if losing all data on the RAID set is no problem (for instance for temporary data)



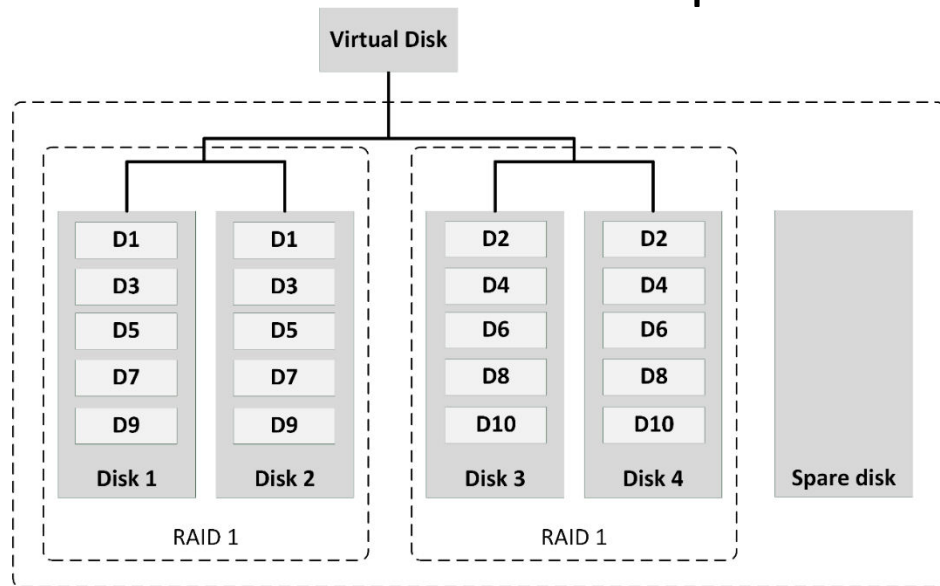
RAID 1 - Mirroring

- RAID 1 is also known as mirroring
- A high availability solution that uses two disks that contain the same data
- If one disk fails, data is not lost as it is still available on the mirror disk
- The most reliable RAID level
- High price
 - 50% of the disks are used for redundancy only
- A spare physical disk can be configured to automatically take over the task of a failed disk



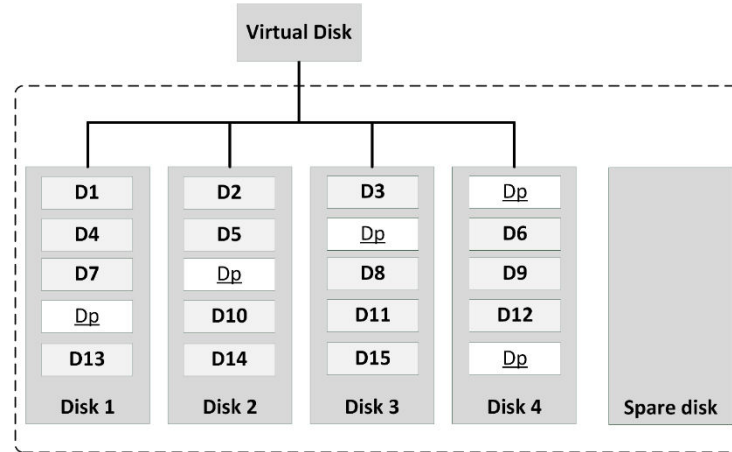
RAID 10 - Striping and Mirroring

- RAID 10 uses a combination of striping and mirroring
- Provides high performance and availability
- Only 50% of the available disk space is used



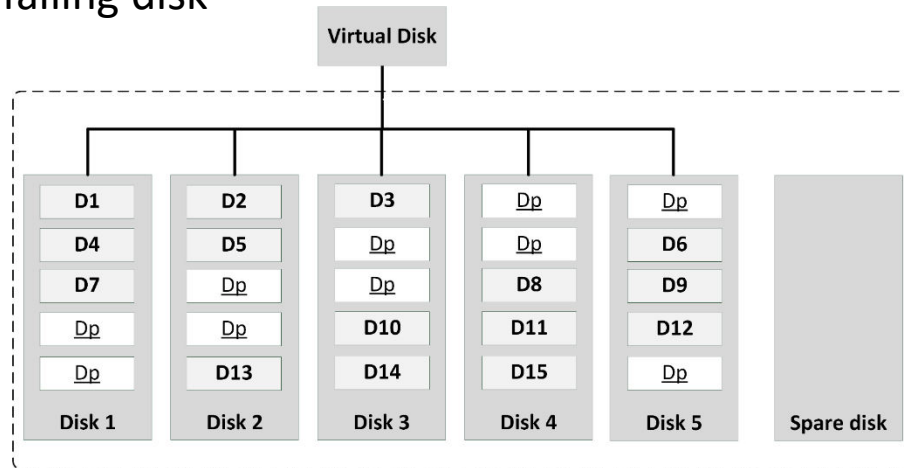
RAID 5 - Striping with Distributed Parity

- Data is written in disk blocks on all disks
- A parity block of the written disk blocks is stored as well
- This parity block is used to automatically reconstruct data in a RAID 5 set (using a spare disk) in case of a disk failure



RAID 6 - Striping with Distributed Double Parity

- RAID 6 protects against double disk failures by using two distributed parity blocks instead of one
- Important in case a second disk fails during reconstruction of the first failing disk



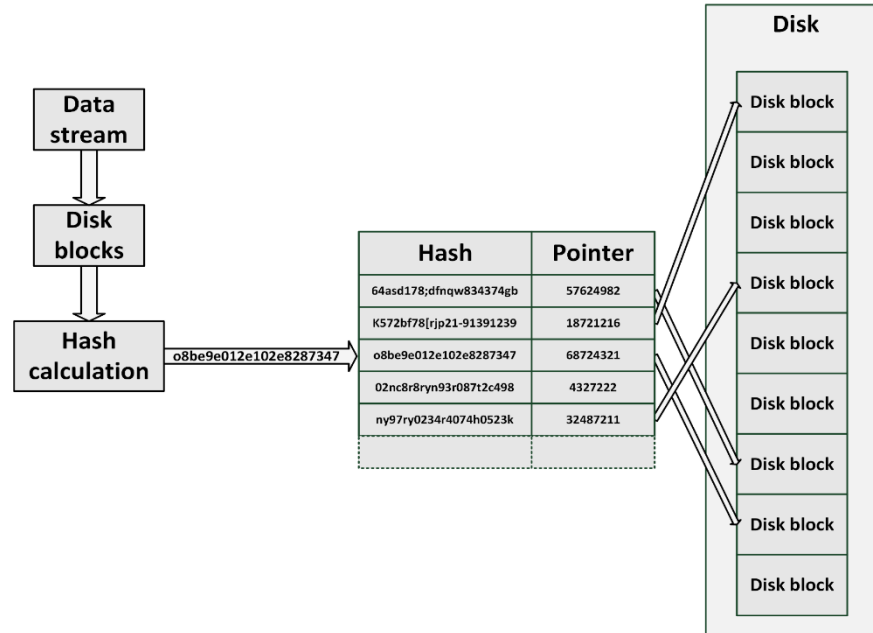


Data deduplication

- Data deduplication searches the storage system for duplicate data segments (disk blocks or files) and removes these duplicates
- Data deduplication is used in archived as well as in production data

Data Deduplication

- The deduplication system keeps a table of hash tags to quickly identify duplicate disk blocks





Data Deduplication Methods

- Deduplication can be done inline or periodically
 - **Inline** deduplication checks for duplicate data segments before data is written to disk
 - Avoids duplicate data on disks at any time
 - Large performance penalty
 - **Periodically:** writing data to disk first, and periodically check if duplicate data exists
 - Duplicate data is deduplicated by changing the duplicate data to a pointer to existing data on disk, and freeing disk space of the original block
 - Performed when performance needs are low
 - Duplicate data will be stored on the disks for some time



Cloning and Snapshots

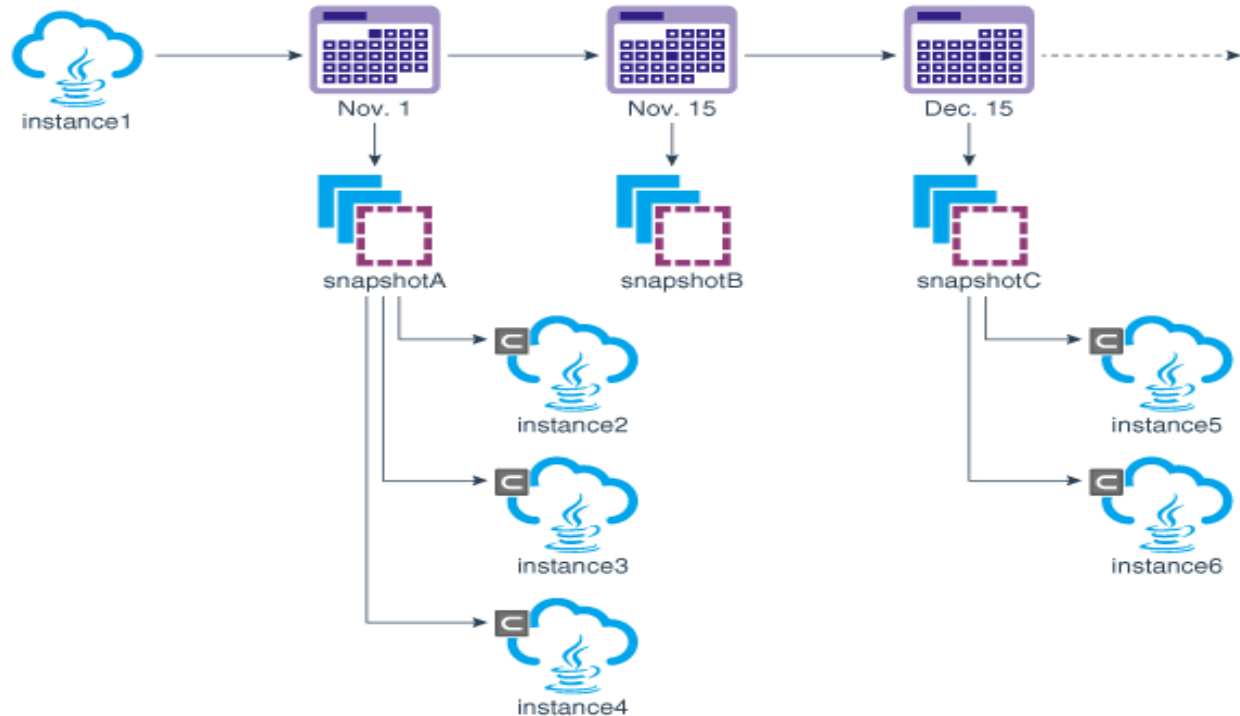
- With cloning and snapshotting, a copy of data is made at a specific point in time that can be used independently from the source data
- **Usage:**
 - Create a backup at a specific point in time, when the data is in a stable, consistent state
 - Creating test sets of data and an easy way to revert to older data without restoring data from a backup
- **Cloning:** the storage system creates a full copy of a disk, much like a RAID 1 mirror disk



Cloning and Snapshots

- **Snapshot:** represents a point in time of the data on the disks
 - No writing to those disks is permitted anymore, as long as the snapshot is active
 - All writing is done on a separate disk volume in the storage system
 - The original disks still provide read-access

Cloning and Snapshots

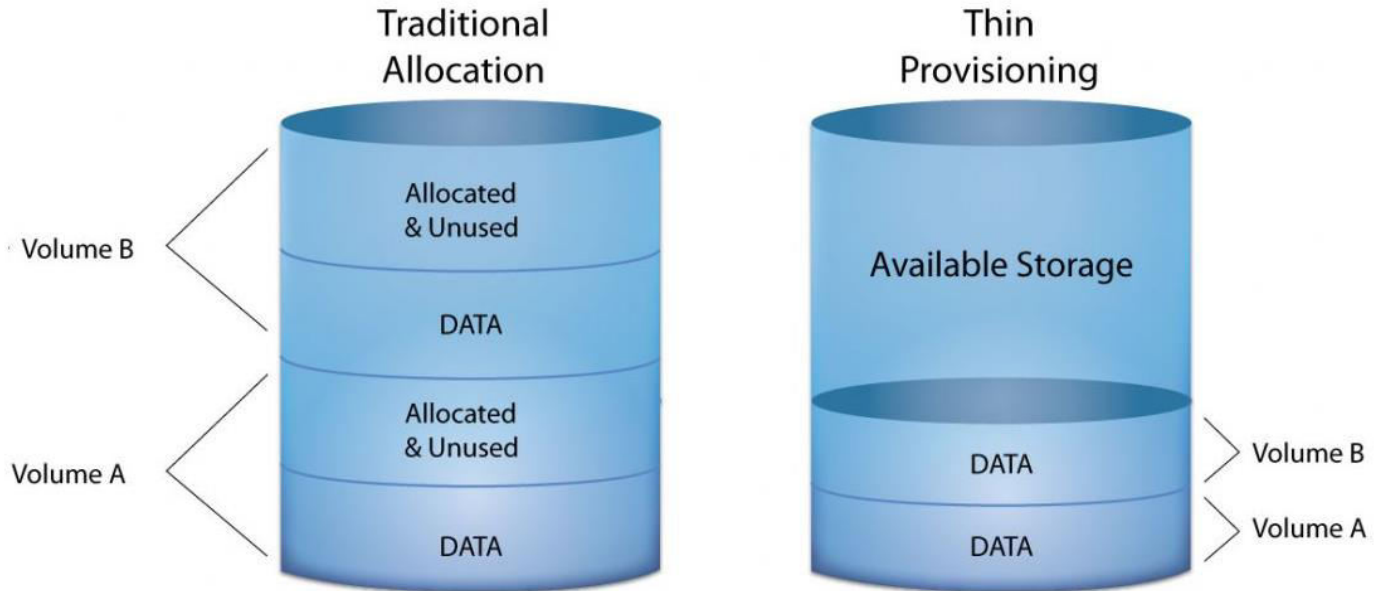




Thin Provisioning

- Thin provisioning enables the allocation of more storage capacity to users than is physically installed
 - About 50% of allocated storage is never used
- Thin provisioning still provides the applications with the required storage
 - Storage is not really available on physical disks
 - Uses automated capacity management
 - The application's real storage need is monitored closely
 - Physical disk space is added when needed
- Typical use: Providing users with large sized home directories or email storage

Thin Provisioning



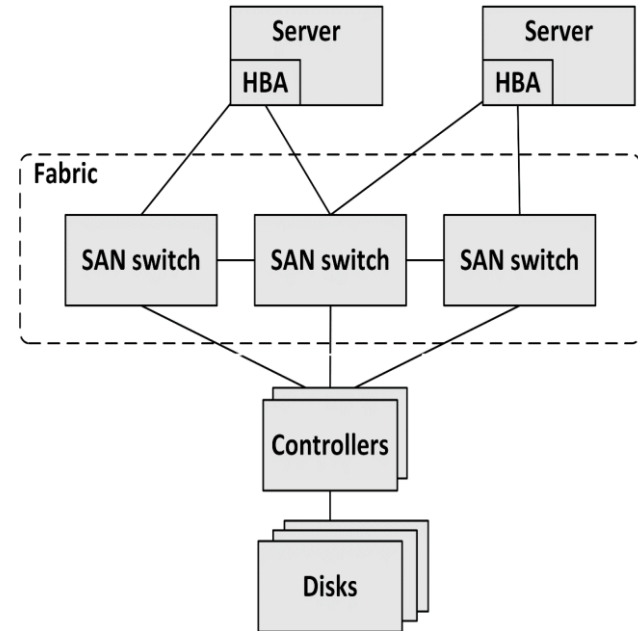


Storage Area Network (SAN)

- Specialized storage network that consists of SAN switches, controllers and storage devices
- Connects a large pool of central storage to multiple servers
- Physically connects servers to disk controllers using specialized networking technologies like Fibre Channel or iSCSI
- Via the SAN, disk controllers offer virtual disks to servers, also known as LUNs (Logical Unit Numbers)
- LUNs are only available to the server that has that specific LUN mounted

Storage Area Network (SAN)

- The core of the SAN is a set of SAN switches, called the Fabric
- Host bus adapters (HBAs) are interface cards implemented in servers



Storage Area Network (SAN)

- In SANs, a large number of disks are installed in one or more disk arrays
- The number of disks varies between dozens of disks and hundreds of disks
- A disk array can easily contain many
- hundreds of terabytes (TB) of data or more





SAN Connectivity Protocols

- The most used SAN connectivity protocols:
 - Fibre Channel
 - FCoE
 - iSCSI



Fibre Channel

- Dedicated level 2 network protocol
- Designed for transportation of storage data
- Speeds: 2 Gbit/s, 4 Gbit/s, 8 Gbit/s, or 16 Gbit/s
- Runs on:
 - Twisted pair copper wire (i.e. UTP and STP)
 - Fiber optic cables
- Very reliable, with guaranteed zero data loss



Fibre Channel

- Three network topologies:
 - Point-to-Point
 - Two devices are connected directly to each other
 - Arbitrated loop
 - Also known as FC-AL
 - All devices are in a loop
 - Switched fabric
 - All devices are connected to Fibre Channel switches
 - A similar concept as in Ethernet implementations
- Most implementations today use a switched fabric



FCoE

- Fibre Channel over Ethernet (FCoE) encapsulates Fibre Channel data in Ethernet packets
- Allows Fibre Channel traffic to be transported over 10 Gbit or higher Ethernet networks
- Eliminates the need for separate Ethernet and Fibre Channel cabling and switching technology
- PCoE needs at least 10 Gbit Ethernet with special extensions, known as Data Center Bridging (DCB) or Converged Enhanced Ethernet (CEE)

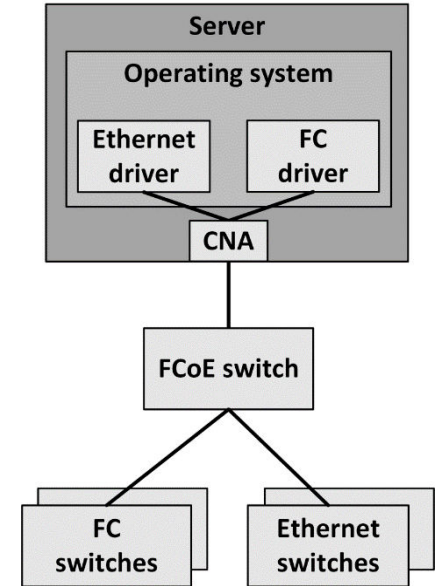


FCoE

- Ethernet extensions:
 - Lossless Ethernet connections
 - A FCoE implementation must guarantee that no Ethernet packets are lost
 - Quality of Service (QoS)
 - Allows FCoE packets to have priority over other Ethernet packets to avoid storage performance issues
 - Large Maximum Transfer Unit (MTU) support
 - Allows Ethernet packets of 2500 bytes in size, instead of the standard 1500 bytes
 - Also known as Jumbo frames

FCoE

- FCoE needs specialized Converged Network Adapters (CNAs)
- CNAs support the Ethernet extensions
- They present themselves to the operating system as two adapters:
 - Ethernet Network Interface Controller (NIC)
 - Fibre Channel Host Bus Adapter (HBA)





iSCSI

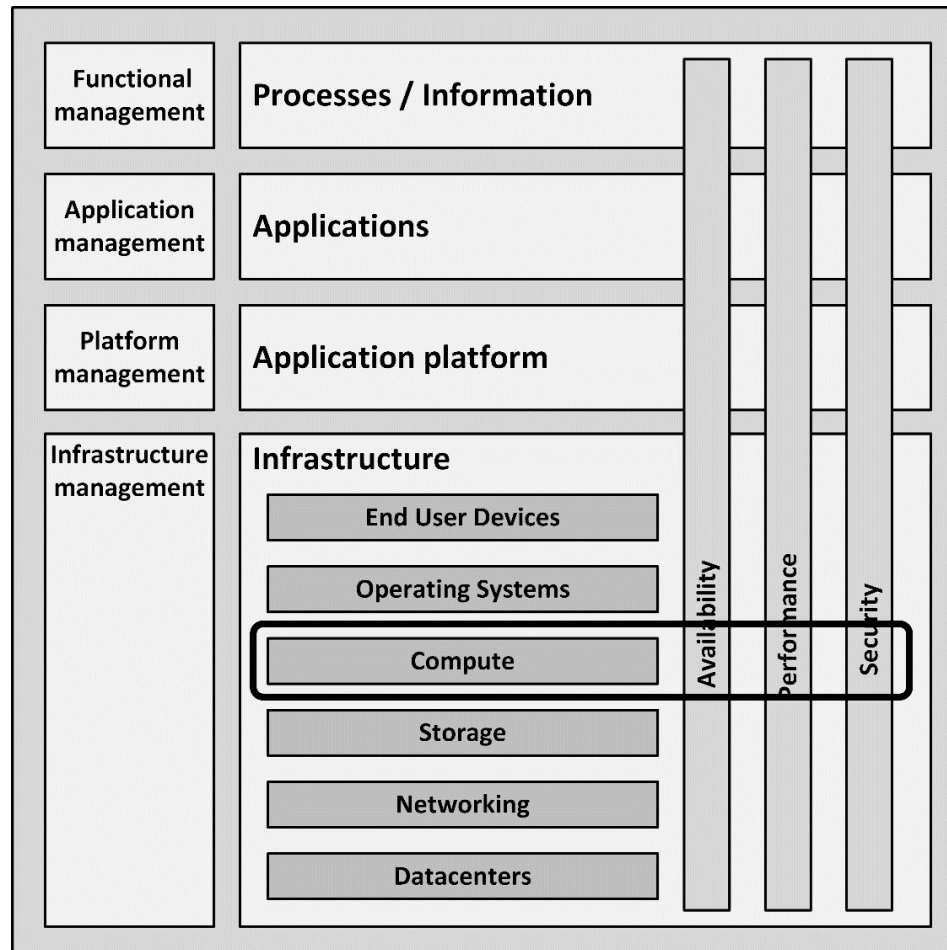
- Allows the SCSI protocol to run over Ethernet LANs using TCP/IP
- Uses the familiar TCP/IP protocols and well known SCSI commands
- Performance is typically lower than that of Fibre Channel, due to the TCP/IP overhead
- With 10 or 40 Gbit/s Ethernet and jumbo frames, iSCSI is now rapidly conquering a big part of the SAN market

IT Infrastructure Architecture

Infrastructure Building Blocks and Concepts

Compute – Part 1
(Chapter 10)

Compute



Compute

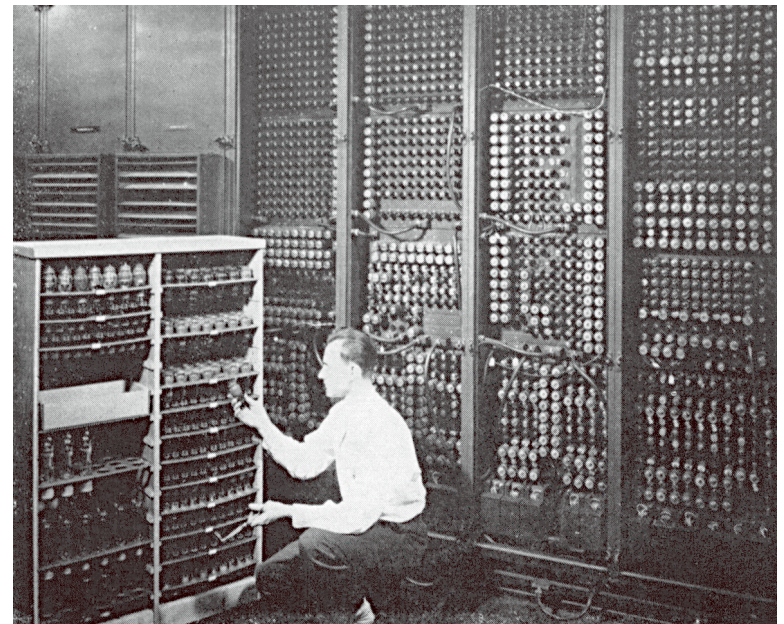
- Compute is an umbrella term for computers located in the datacenter.
- The term "compute" generally refers to the process of performing calculations or executing instructions using a computer or other electronic device.
- It encompasses a wide range of activities related to data processing, including mathematical operations, logical operations, data manipulation, and algorithm execution.
 - Physical machines or virtual machines



Three groups:

- Mainframes
- Midrange systems
- x86 servers
- Physical computers contain:
 - Power supplies
 - Central Processing Units
 - A Basic Input/Output System
 - Memory
 - Expansion ports
 - Network connectivity
 - A keyboard, mouse, and monitor

- The British Colossus computer, created during World War II, was the world's first programmable computer
 - Information about it was classified under British secrecy laws
- The first publicly recognized general purpose computer was the ENIAC (Electronic Numerical Integrator And Computer)
 - The ENIAC was designed in 1943 and was financed by the United States Army in the midst of World War II



- The ENIAC:
 - Could perform 5,000 operations per second
 - Used more than 17,000 vacuum tubes
 - Got its input using an IBM punched card reader
 - Punched cards were used for output
- In the 1960s computers started to be built using transistors instead of vacuum tubes
 - Smaller
 - Faster
 - Cheaper to produce
 - Required less power
 - Much more reliable

- The transistor based computers were followed in the 1970s by computers based on integrated circuit (IC) technology
 - ICs are small chips that contain a set of transistors providing standardized building blocks like AND gates, OR gates, counters, adders, and flip-flops
 - By combining building blocks, CPUs and memory circuits could be created
- Microprocessors decreased size and cost of computers even further
 - Increased their speed and reliability
 - In the 1980s microprocessors were cheap enough to be used in personal computers

Compute Building Blocks

- Computer Housing
- Processors
- Memory
- Compute Virtualization
- Container Technology
- Mainframe Computers
- Midrange Systems
- X86 Servers
- Super Computers

Computer Housing

- Originally, computers were stand-alone complete systems, called pedestal or tower computers
 - Placed on the datacenter floor
- Most x86 servers and midrange systems are now:
 - Rack mounted
 - Blade servers
- Blade servers are less expensive than rack mounted servers
 - They use the enclosure's shared components like power supplies and fans



Computer Housing

- A blade enclosure typically hosts from 8 to 16 blade servers
- Blade enclosure provides:
 - Shared redundant power supplies for all blades
 - Shared backplane to connect all blades
 - Redundant network switches to connect the blades' Ethernet interfaces providing redundant Ethernet connections to other systems
 - Redundant SAN switches to connect the HBA interfaces on the blade servers providing dual redundant Fibre Channel connections to other systems
 - A management module to manage the enclosure and the blades in it

Computer Housing

- The amount of wiring in a blade server setup is substantially reduced when compared to traditional server racks
 - Less possible points of failure
 - Lower initial deployment costs
- Enclosures are often not only used for blade servers, but also for storage components like disks, controllers, and SAN switches

Processors

- In a computer, the Central Processing Unit (CPU) – or processor – executes a set of instructions
- A CPU is the electronic circuitry that carries out the instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions
- Today's processors contain billions of transistors and are extremely powerful



Processor Instructions

- CPU can perform a fixed number of instructions such as ADD, SHIFT BITS, MOVE DATA, and JUMP TO CODE LOCATION, called the instruction set
- A program created using CPU instructions is referred to as machine code
- Each instruction is associated with an English like mnemonic
 - Easier for people to remember
 - Set of mnemonics is called the assembly language
- For example:
 - Binary code for the ADD WITH CARRY
 - Machine code instruction: 10011101
 - Mnemonic : ADC

Processors - Programming

- The assembler programming language is the lowest level programming language for computers
- Higher level programming languages are much more human friendly
 - C#
 - Java
 - Python
- Programs written in these languages are translated to assembly code before they can run on a specific CPU
- This compiling is done by a high-level language compiler

Processors - Speed

- A CPU needs a high frequency clock to operate, generating so-called clock ticks or clock cycles
 - Each machine code instruction takes one or more clock ticks to execute
 - An ADD instruction typically costs 1 tick to compute
- The speed at which the CPU operates is defined in GHz (billions of clock ticks per second)
 - A single core of a 2.4 GHz CPU can perform 2.4 billion additions in 1 second

Processors – Word Size

- Each CPU is designed to handle data in chunks, called words, with a specific size
 - The first CPUs had a word size of 4 bits
 - Today, most CPUs have a word size of 64 bits
- The word size is reflected in many aspects of a CPU's structure and operation:
 - The majority of the internal memory registers are the size of one word
 - The largest piece of data that can be transferred to and from the working memory in a single operation is a word
 - A 64-bit CPU can address 17,179,869,184 TB of memory (64-bit word)

Intel x86 Processors

- Intel CPUs became the de-facto standard for many computer architectures
 - The original PC used a 4.77 MHz 16-bit 8088 CPU
 - A few years later, Intel produced the 32-bit 80386 and the 80486 processors
- Since these names all ended with the number 86, the generic architecture was referred to as x86
- In 2017, the latest Intel x86 model is the 22-core E5-2699A Xeon Processor, running on 2.4 GHz

AMD x86 Processors

- Advanced Micro Devices, Inc. (AMD) is the second-largest global supplier of microprocessors based on the x86 architecture
 - In 1982, AMD signed a contract with Intel, becoming a licensed second-source manufacturer of 8086 and 8088 processors for IBM
 - Intel cancelled the licensing contract in 1986
 - AMD still produces x86 compatible CPUs, forcing Intel to keep innovating and to keep CPU prices relatively low
- In 2017, the latest model is the 16-core AMD Opteron 6386 SE CPU, running on 2.8 GHz

Itanium and x86-64 Processors

- The Itanium processor line was a family of 64-bit high-end CPUs meant for high-end servers and workstations
 - Not based on the x86 architecture
 - HP was the only company to actively produce Itanium based systems, running HP-UX and OpenVMS
- In 2005, AMD released the K8 core processor architecture as an answer to Intel's Itanium architecture
 - The K8 included a 64-bit extension to the x86 instruction set
 - Later, Intel adopted AMD's processor's instruction set as an extension to its x86 processor line, called x86-64
- Today, the x86-64 architecture is used in all Intel and AMD processors

ARM Processors

- The ARM (Advanced RISC Machine) is the most used CPU in the world
- In 2013, 10 billion ARM processors were shipped, running on:
 - 95% of smartphones
 - 90% of hard disk drives
 - 40% of digital televisions and set-top boxes
 - 15% of microcontrollers
 - 20% of mobile computers
- The CPU is produced by a large number of manufacturers under license of ARM
- Since 2016, ARM is owned by Japanese telecommunications company SoftBank Group

Oracle SPARC Processors

- In 1986, Sun Microsystems started to produce the SPARC processor series for their Solaris UNIX based systems
- The SPARC architecture is fully open and non-proprietary
 - A true open source hardware design
 - Any manufacturer can get a license to produce a SPARC CPU
- Oracle bought Sun Microsystems in 2010
- SPARC processors are still used by Oracle in their Exadata and Exalogic products
- In 2017, the latest model is the 32-core SPARC M7 CPU, running on 4.1 GHz

IBM POWER Processors

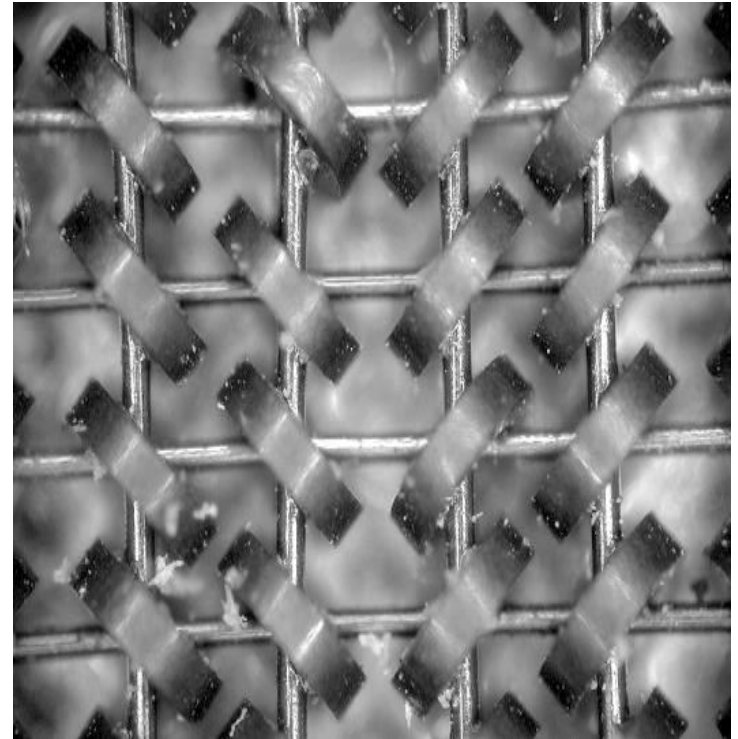
- POWER (also known as PowerPC) is a series of CPUs
 - Created by IBM
 - Introduced in 1990
- IBM uses POWER CPUs in many of their high-end server products
 - Watson, the supercomputer that won Jeopardy in 2011, was equipped with 3,000 POWER7 CPU cores
- In 2017, the latest model is the 24-core POWER9 CPU, running on 4 GHz

Memory – Early Systems

- The first computers used vacuum tubes to store data
 - Extremely expensive, uses much power, fragile, generates much heat
- An alternative to vacuum tubes were relays
 - Mechanical parts that use magnetism to move a physical switch
 - Two relays can be combined to create a single bit of memory storage
 - Slow, uses much power, noisy, heavy, and expensive
- Based on cathode ray tubes, the Williams tube was the first random access memory, capable of storing several thousands of bits, but only for some seconds

Memory – Early Systems

- The first truly useable type of main memory was magnetic core memory, introduced in 1951
- The dominant type of memory until the late 1960s
 - Uses very small magnetic rings, called cores, with wires running through them
 - The wires can polarize the magnetic field one direction or the other in each individual core
 - One direction means 1, the other means 0
- Core memory was replaced by RAM chips in the 1970s



- RAM: Random Access Memory
 - Any piece of data stored in RAM can be read in the same amount of time, regardless of its physical location
- Based on transistor technology, typically implemented in large amounts in Integrated Circuits (ICs)
- Data is volatile – it remains available as long as the RAM is powered

- Static RAM (SRAM)
 - Uses flip-flop circuitry to store bits
 - Six transistors per bit
- Dynamic RAM (DRAM)
 - Uses a charge in a capacitor
 - One transistor per bit
 - DRAM loses its data after a short time due to the leakage of the capacitors
 - To keep data available in DRAM it must be refreshed regularly (typically 16 times per second)

- The Basic Input/Output System (BIOS) is a set of instructions stored on a memory chip located on the computer's motherboard
- The BIOS controls a computer from the moment it is powered on, to the point where the operating system is started
- Mostly implemented in a Flash memory chip
- It is good practice to update the BIOS software regularly
 - Upgrading computers to the latest version of the BIOS is called BIOS flashing

- Connecting computers to external peripherals is done using interfaces
- External interfaces use connectors located at the outside of the computer case
 - One of the first standardized external interfaces was the serial bus based on RS-232
 - RS-232 is still used today in some systems to connect:
 - Older type of peripherals
 - Industrial equipment
 - Console ports
 - Special purpose equipment

- The Universal Serial Bus (USB) was introduced in 1996 as a replacement for most of the external interfaces on servers and PCs
- Can provide operating power to attached devices
- Up to seven devices can be daisy-chained
 - Hubs can be used to connect multiple devices to one USB computer port
- In 2013, USB 3.1 was introduced
 - Provides a throughput of 10 Gbit/s
- In 2014, USB Type-C was introduced
 - Smaller connector
 - Ability to provide more power to connected devices

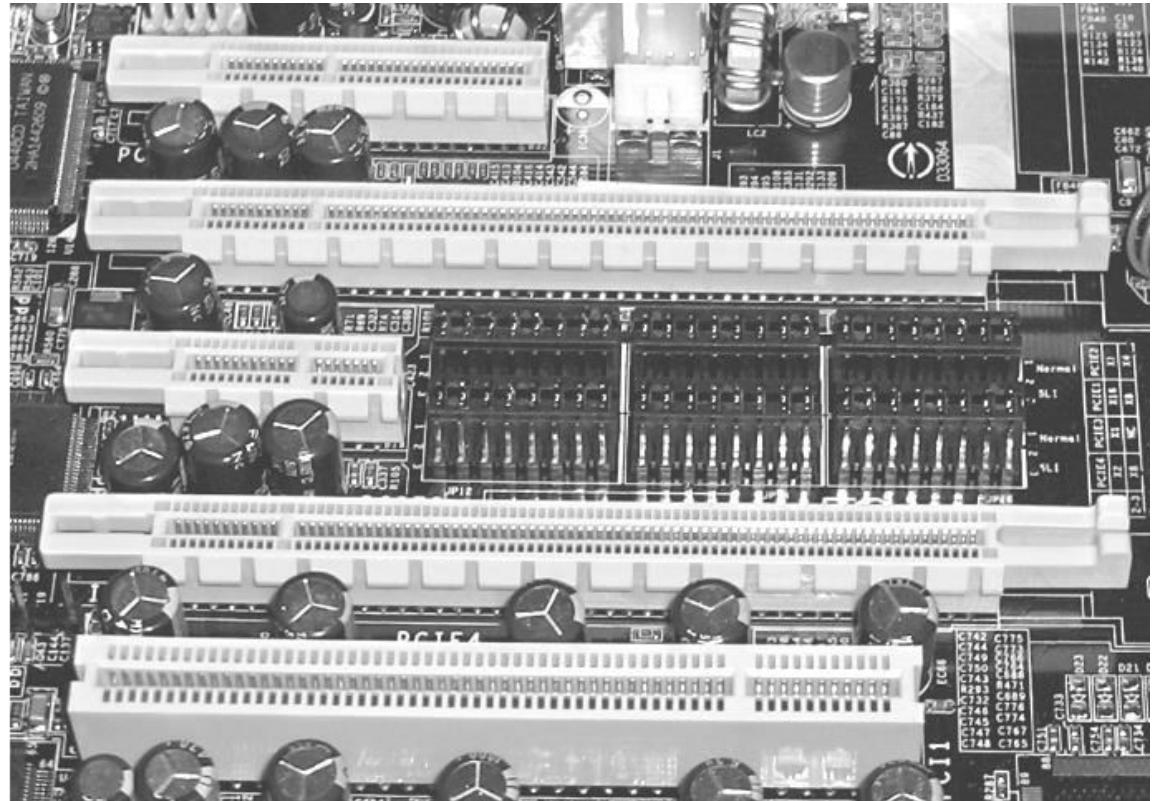
- Thunderbolt, also known as Light Peak, was introduced in 2011
- Thunderbolt 3 was released in 2015
 - Can provide a maximum throughput of 40 Gbit/s
 - Provide 100 W power to devices
 - Uses the USB Type-C connector
 - Backward compatible with USB 3.1

- Internal interfaces, typically some form of PCI, are located on the system board of the computer, inside the case, and connect expansion boards like network adapters and disk controllers
- Uses a shared parallel bus architecture
 - Only one shared communication path between two PCI devices can be active at any given time

- PCI Express (PCIe) uses a topology based on point-to-point serial links, rather than a shared parallel bus architecture
 - A connection between any two PCIe devices is known as a link
 - A collection of 1 or more links is called a lane
- Routed by a hub on the system board acting as a crossbar switch
 - The hub allows multiple pairs of devices to communicate with each other at the same time
- Despite the availability of the much faster PCIe, conventional PCI remains a very common interface in computers

PCI and PCIe

- PCIe x 4 →
- PCIe x 16 →
- PCIe x 1 →
- PCIe x 16 →
- PCI →

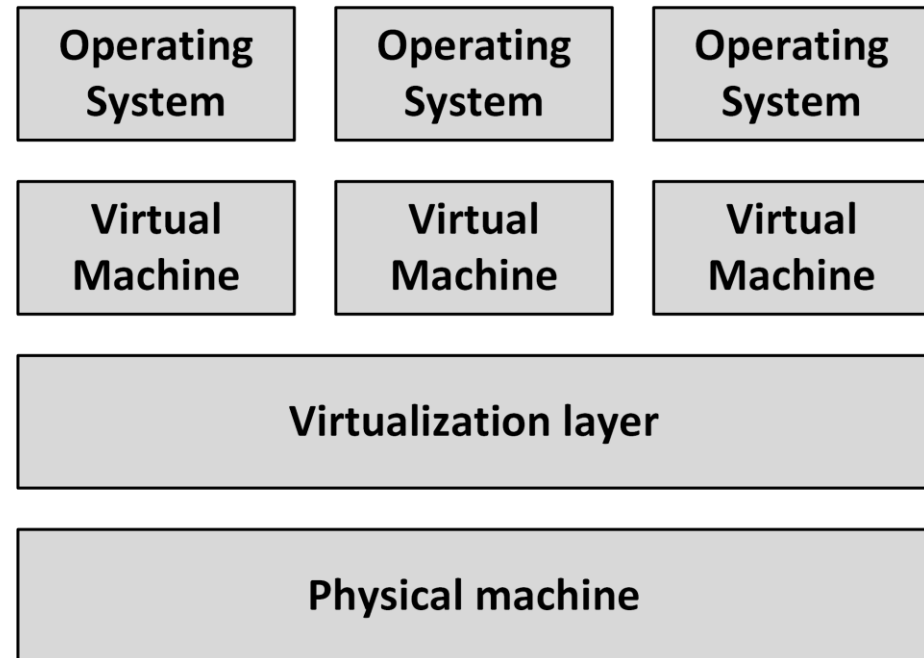


PCI speeds in Gbit/s

	Lanes					
	1	2	4	8	16	32
PCI 32-bit/33 MHz	1					
PCI 32-bit/66 MHz	2					
PCI 64-bit/33 MHz	2					
PCI 64-bit/66 MHz	4					
PCI 64-bit/100 MHz	6					
PCIe 1.0	2	4	8	16	32	64
PCIe 2.0	4	8	16	32	64	128
PCIe 3.0	8	16	32	64	128	256
PCIe 4.0	16	32	64	128	256	512

Compute Virtualization

- Compute virtualization is also known as:
 - Server virtualization
 - Software Defined Compute
- Introduces an abstraction layer between physical computer hardware and the operating system using that hardware
 - Allows multiple operating systems to run on a single physical machine
 - Decouples and isolates virtual machines from the physical machine and from other virtual machines



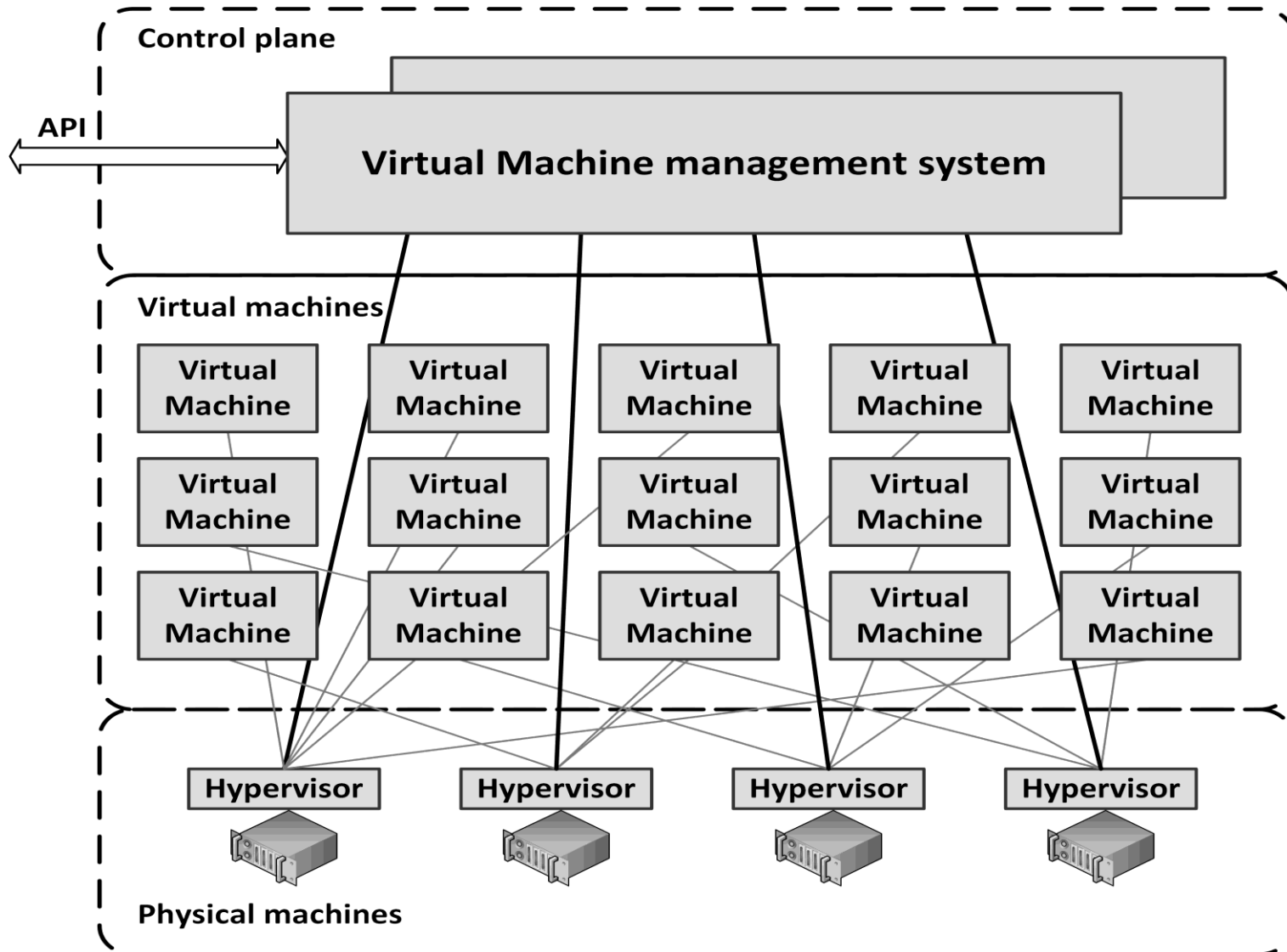
Compute Virtualization

- A virtual machine is a logical representation of a physical computer in software
- New virtual machines can be provisioned without the need for a hardware purchase
 - With a few mouse clicks or using an API
 - New virtual machines can be installed in minutes
- Costs can be saved on hardware, power, and cooling by consolidating many physical computers as virtual machines on fewer (bigger) physical machines
- Because fewer physical machines are needed, the cost of maintenance contracts can be reduced and the risk of hardware failure is reduced

Software Defined Compute (SDC)

- Virtual machines are typically managed using one redundant centralized virtual machine management system
 - Enables systems managers to manage more machines with the same number of staff
 - Allows managing the virtual machines using APIs
 - Server virtualization can therefore be seen as Software Defined Compute
- In SDC, all physical machines are running a hypervisor and all hypervisors are managed as one layer using management software

Software Defined Compute (SDC)



Software Defined Compute (SDC)

- Some virtualization platforms allow running virtual machines to be moved automatically between physical machines
- Benefits:
 - When a physical machine fails, all virtual machines that ran on the failed physical machine can be restarted automatically on other physical machines
 - Virtual machines can automatically be moved to the least busy physical machines
 - Some physical machines can get fully loaded while other physical machines can be automatically switched off, saving power and cooling cost
 - Enables hardware maintenance without downtime

Disadvantages of Computer Virtualization

- Because creating a new virtual machine is so easy, virtual machines tend to get created for all kinds of reasons
 - This effect is known as "virtual machine sprawl"
 - All VMs:
 - Must be managed
 - Use resources of the physical machine
 - Use power and cooling
 - Must be back-upped
 - Must be kept up to date by installing patches

Disadvantages of Computer Virtualization

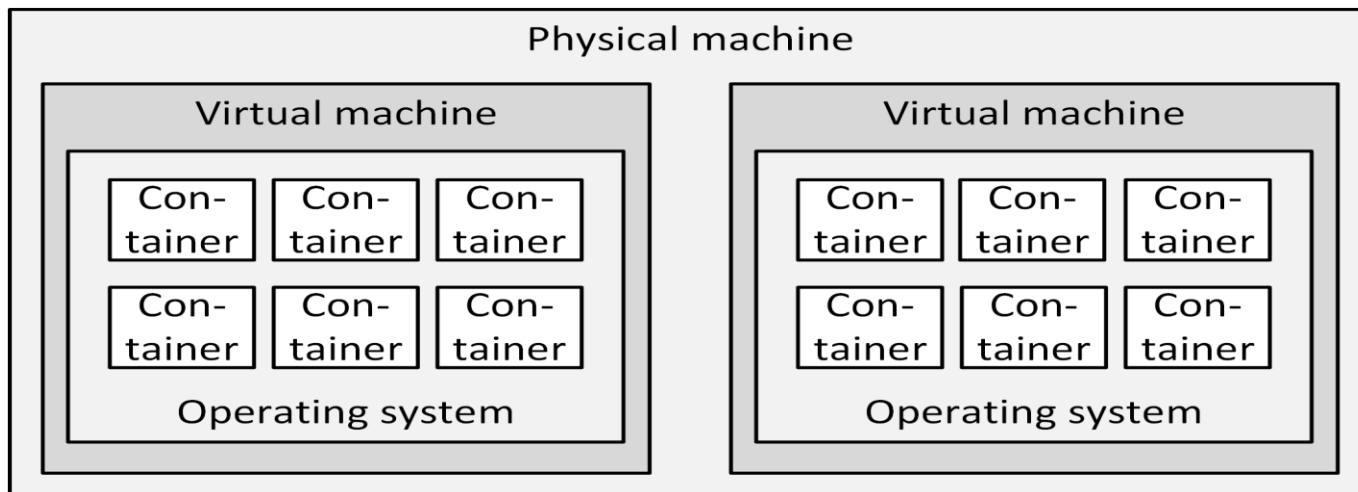
- Introduction of an extra layer in the infrastructure
 - License fees
 - Systems managers training
 - Installation and maintenance of additional tools
- Virtualization cannot be used on all servers
 - Some servers require additional specialized hardware, like modem cards, USB tokens or some form of high speed I/O like in real-time SCADA systems
- Virtualization is not supported by all application vendors
 - When the application experiences some problem, systems managers must reinstall the application on a physical machine before they get support

- Emulation
 - Can run programs on a computer, other than the one they were originally intended for
 - Run a mainframe operating system on a x86 server
- Logical Partitions (LPARs)
 - Hardware based
 - Used on mainframe and midrange systems

- Hypervisors
 - Control the physical computer's hardware and provide virtual machines with all the services of a physical system
 - Virtual CPUs
 - BIOS
 - Virtual devices
 - Virtualized memory management
 - Three types:
 - Binary translation
 - Paravirtualization
 - Hardware assisted virtualization (most used on x86 servers)

Container Technology

- Container technology is a server virtualization method in which the kernel of an operating system provides multiple isolated user-space instances, instead of just one
- Containers look and feel like a real server from the point of view of its owners and users, but they share the same operating system kernel
- Containers are part of the Linux kernel since 2008



Container Technology

- Containers have a number of benefits:
 - Isolation
 - Applications or application components can be encapsulated in containers, each operating independently and isolated from each other
 - Portability
 - Since containers typically contain all components the application needs to function, including libraries and patches, containers can be run on any infrastructure that is capable of running containers using the same kernel version
 - Easy deployment
 - Containers allow developers to quickly deploy new software versions, as their containers can be moved from the development environment to the production environment unaltered

Container Implementation

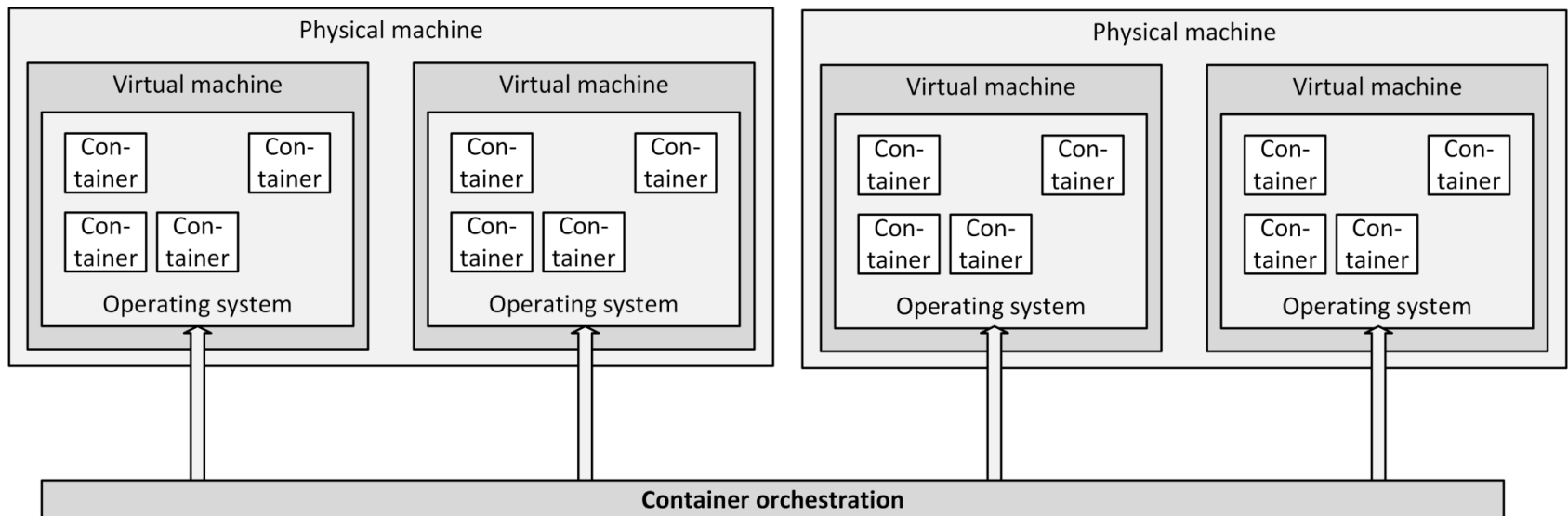
- Containers are based on 3 technologies that are all part of the Linux kernel:
 - Chroot (also known as a jail)
 - Changes the root directory for the current running process
 - Ensures processes cannot access files outside the designated directory tree
 - Namespaces
 - Allows complete isolation of an applications' view of the operating environment
 - Process trees, networking, user IDs and mounted file systems

Container Implementation

- Cgroups
 - Limits and isolates the resource usage of a collection of processes
 - PU, memory, disk I/O, network
- Linux Containers (LXC), introduced in 2008, is a combination of these
- Docker is a popular implementation of a container ecosystem

Container Orchestration

- Container orchestration abstracts the resources of a cluster of machines and provides services to containers
- A container orchestrator enables containers to be run anywhere on a cluster of machines
 - Schedules the containers to run on any machine that has resources available
 - Acts like a kernel for the combined resources of an entire datacenter



- A mainframe is a high-performance computer made for high-volume, I/O-intensive computing
 - Expensive
 - Mostly used for administrative processes
 - Optimized for handling high volumes of data
- IBM is the largest vendor – it has 90% market share
- The end of the mainframe is predicted for decades now, but mainframes are still widely used
- Today's mainframes are still large (the size of a few 19" racks), but they don't fill-up a room anymore

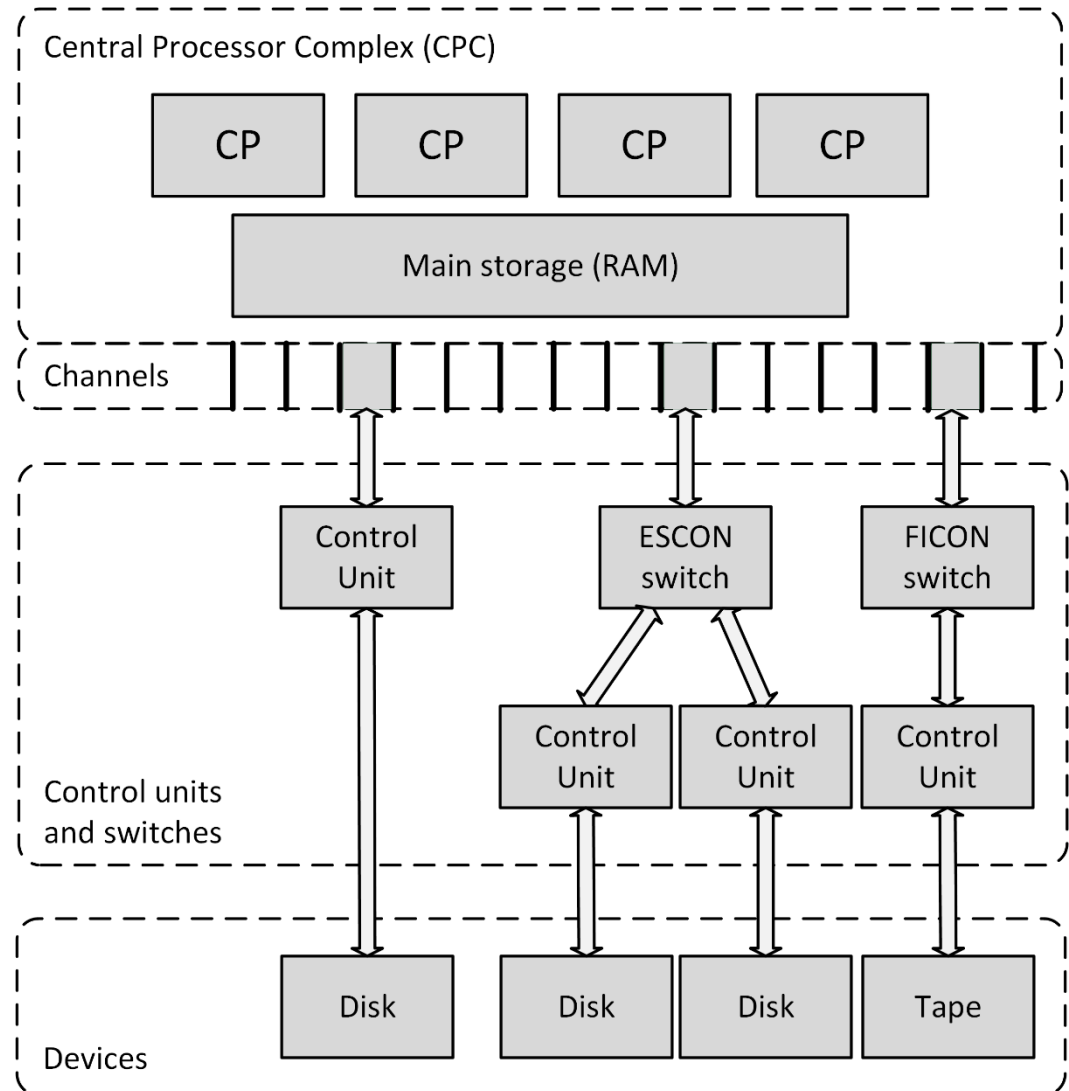
Mainframes

- Mainframes are highly reliable, typically running for years without downtime
- Much redundancy is built in
 - Hardware can be upgraded and repaired while the mainframe is operating without downtime
- The latest IBM z13 mainframe:
 - Introduced in 2015
 - Up to 10TB of memory
 - Up to 141 processors
 - Running at a 5GHz clock speed
 - Can run up to 8000 virtual machines simultaneously



Mainframe Architecture

- A mainframe consists of:
 - Processing units (PUs)
 - Memory
 - I/O channels
 - Control units
 - Devices, all placed in racks (frames)



Mainframe architecture – PU, Memory, and Disks

- In the mainframe world, the term PU (Processing Unit) is used instead of CPU
 - A mainframe has multiple PUs, so there is no central processing unit
 - The total of all PUs in a mainframe is called a Central Processor Complex (CPC)
- Each book package in the CPC cage contains from four to eight memory cards
 - For example, a fully loaded z9 mainframe has four book packages that can provide up to a total of 512 GB of memory
- Disks in mainframes are called DASD (Direct Attached Storage Device)
 - Comparable to a SAN in a midrange or x86 environment

Mainframe architecture – Channels and Control Units

- A channel provides a data and control path between I/O devices and memory
- Today's largest mainframes have 1024 channels
- A control unit is similar to an expansion card in an x86 or midrange system
 - Contains logic to work with a particular type of I/O device, like a printer or a tape drive

- Channel types:
 - OSA
 - Connectivity to various industry standard networking technologies, including Ethernet
 - FICON
 - The most flexible channel technology, based on fiber-optic technology
 - With FICON, input/output devices can be located many kilometers from the mainframe to which they are attached
 - ESCON
 - An earlier type of fiber-optic technology
 - Almost as fast as FICON channels, but at a shorter distance

Mainframe Virtualization

- Mainframes were designed for virtualization from the start
- Logical partitions (LPARs) are the default virtualization solution
- LPARs are equivalent to separate mainframes
- A common number of LPARs in use on a mainframe is less than ten
- The mainframe operating system running on each LPAR is designed to concurrently run a large number of applications and services, and can be connected to thousands of users at the same time
- Often one LPAR runs all production tasks while another runs the consolidated test environment

IT Infrastructure Architecture

Infrastructure Building Blocks and Concepts

Compute – Part 2
(chapter 10)

Midrange Systems

- The midrange platform is positioned between the mainframe platform and the x86 platform
- Built using parts from only one vendor, and run an operating system provided by that same vendor
- This makes the platform:
 - Stable
 - High available
 - Secure

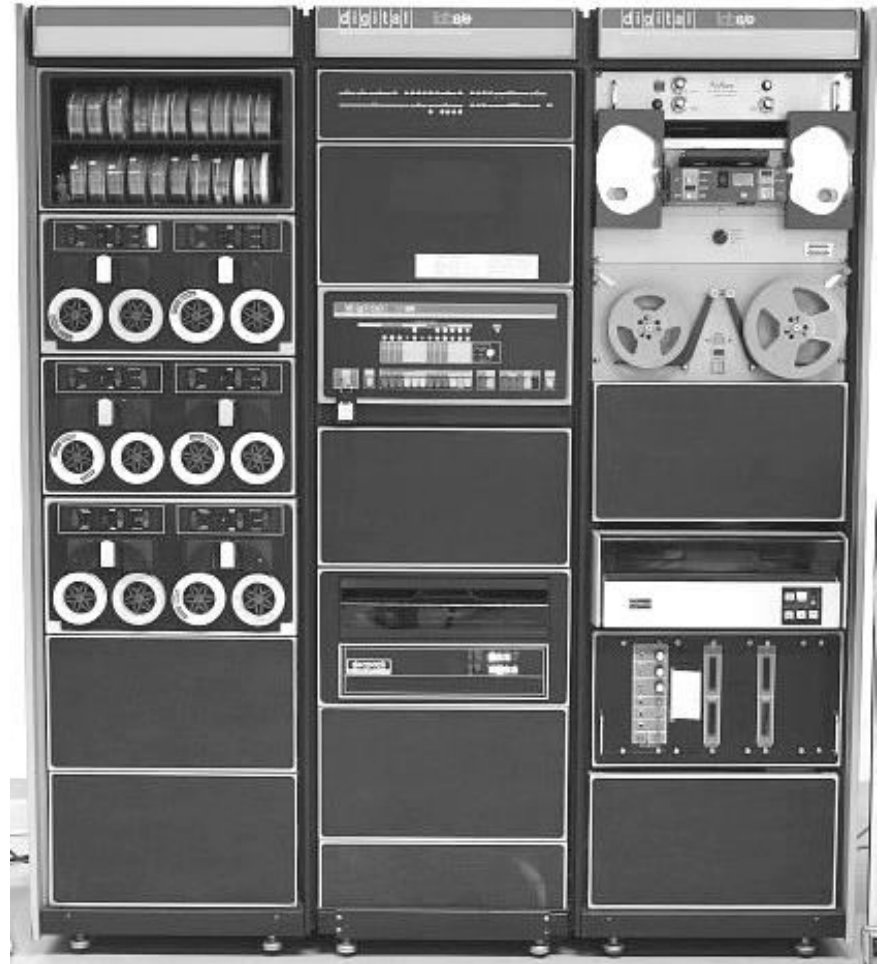


Midrange Systems

- Today midrange systems are produced by three vendors:
 - IBM
 - Power Systems series
 - Operating system: AIX UNIX, Linux and IBM i
 - Hewlett-Packard
 - HP Integrity systems
 - Operating system: HP-UX UNIX and OpenVMS
 - Oracle
 - Sun Microsystems's based SPARC servers
 - Operating system: Solaris UNIX

Midrange Systems - History

- Minicomputer evolved in the 1960s as small computers that became possible with the use of IC and core memory technologies
- Small was relative
 - A single minicomputer typically was housed in a few cabinets the size of a 19" rack
- The first commercially successful minicomputer was DEC PDP-8, launched in 1964



Midrange systems - History

- Minicomputers became powerful systems
 - They ran full multi-user, multitasking operating systems like OpenVMS and UNIX
- In the 1980s, minicomputers (a.k.a. midrange systems) became less popular
 - A result of the lower cost of PCs, and the emergence of LANs
- Still used in places where high availability, performance, and security are very important

Midrange Systems - Architecture

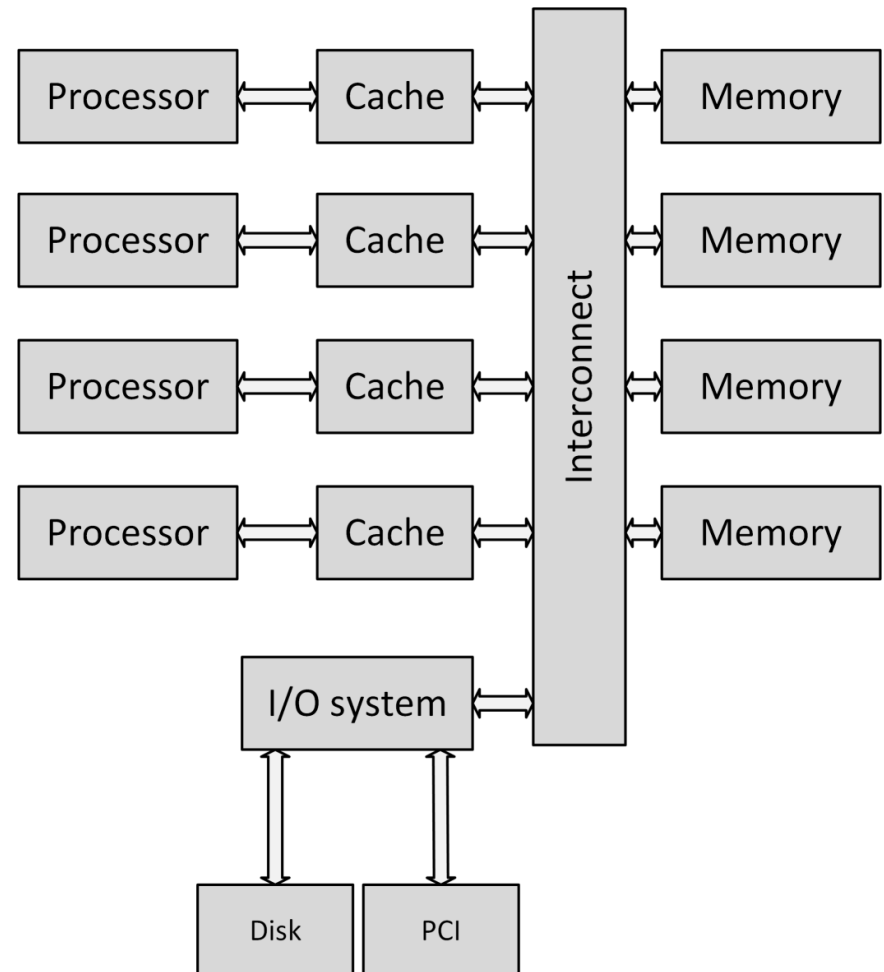
- The architecture of most midrange systems:
 - Uses multiple CPUs
 - Based on a shared memory architecture
- In a shared memory architecture, all CPUs in the system can access all installed memory blocks
 - Changes made in memory by one CPU are immediately seen by all other CPUs
 - A shared bus connects all CPUs and all RAM
 - The I/O system is also connected to the interconnection network

Midrange Systems - Architecture

- Shared memory architectures come in two flavors:
 - Uniform Memory Access (UMA)
 - Non-Uniform Memory Access (NUMA)
 - Cache coherence is needed
 - If one CPU writes to a location in shared memory, all other CPUs must update their caches to reflect the changed data
 - Cache coherent versions are known as ccUMA and ccNUMA

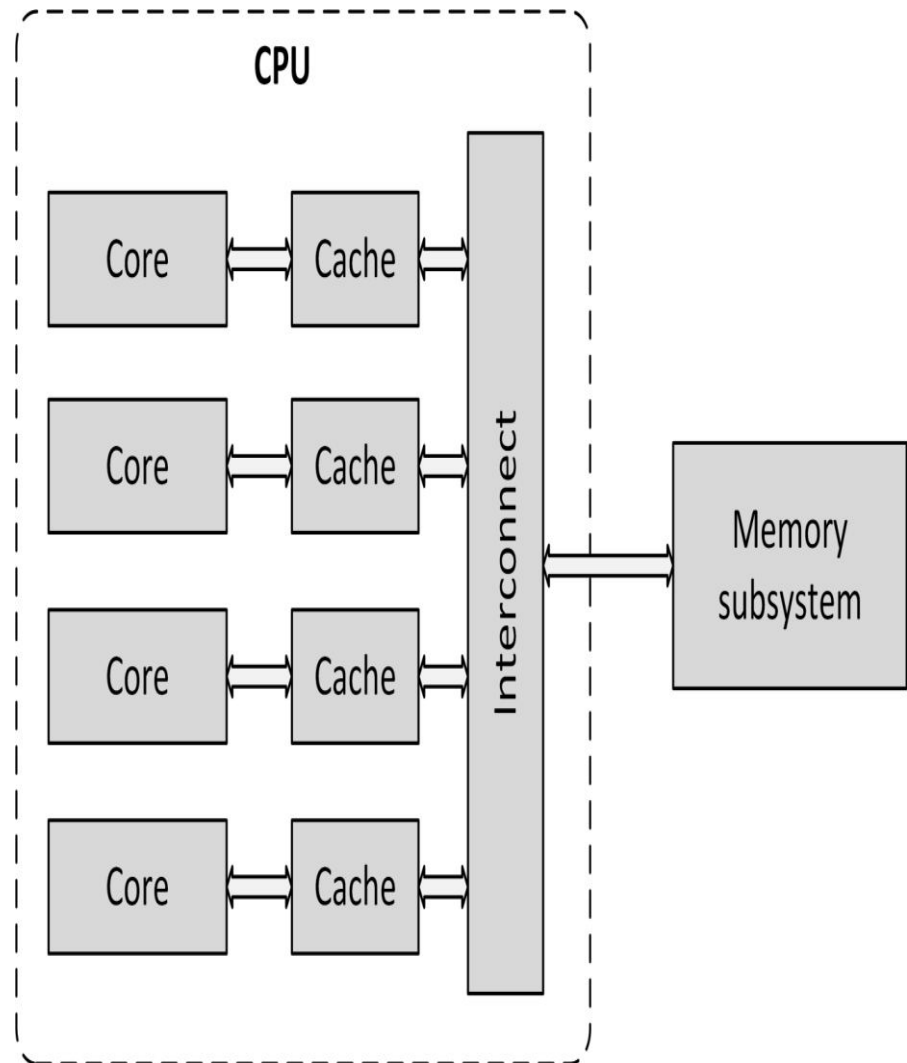
Midrange Systems – UMA Architecture

- The UMA architecture is one of the earliest styles of multi-CPU architectures, typically used in systems with no more than 8 CPUs
- The machine is organized into a series of nodes containing either a processor, or a memory block
- Nodes are interconnected, usually by a shared bus



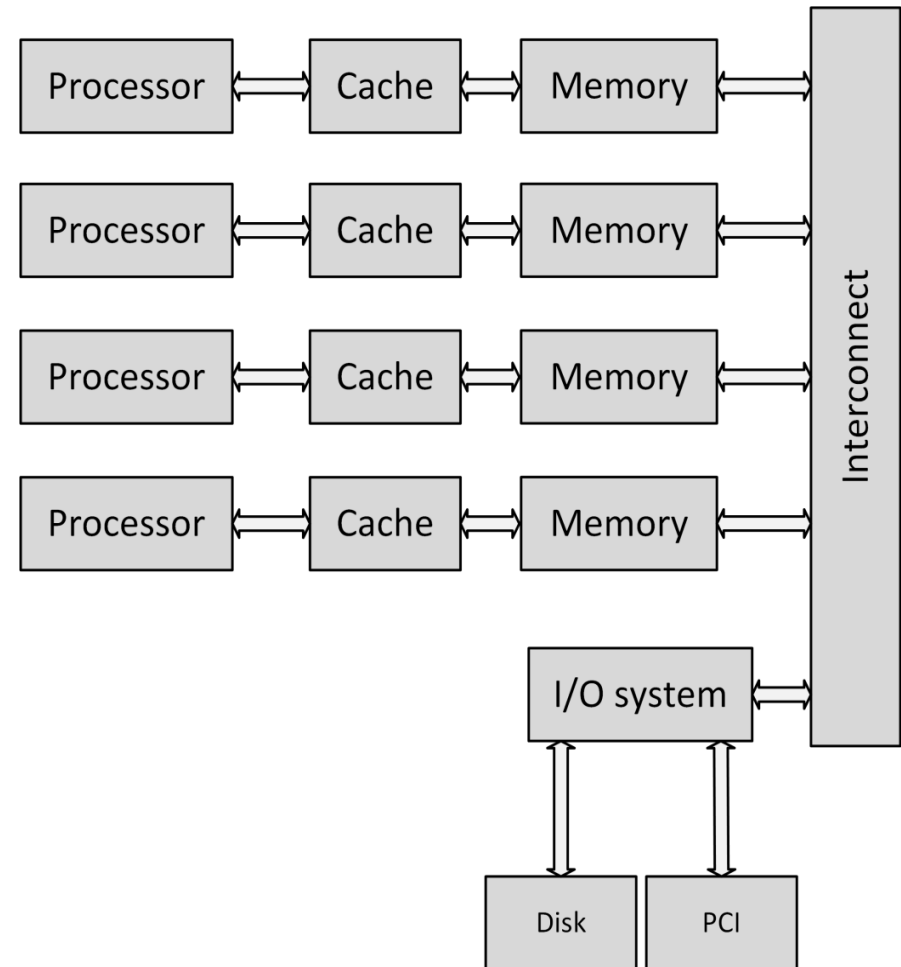
Midrange Systems – SMP Architecture

- UMA systems are also known as Symmetric Multi-Processor (SMP) systems
- SMP is used in x86 servers as well as early midrange systems
- Can be Implemented inside multi-core CPUs
 - The interconnect is implemented on-chip in the CPU
 - A single path to the main memory is provided between the chip and the memory subsystem



Midrange systems – NUMA architecture

- NUMA is a computer architecture in which the machine is organized into a series of nodes
- Each node contains processors and memory
- Nodes are interconnected using a crossbar interconnect
- When a processor accesses memory not within its own node, data must be transferred over the interconnect
 - Slower than accessing local memory
 - Memory access times are **non-uniform**



Midrange Virtualization

- Most midrange platform vendors provide virtualization based on LPARs
- LPARS are a type of hardware partitioning
 - IBM AIX: Workload/Working Partitions (WPARs)
 - HP: nPARs
 - Oracle Solaris: zones and containers

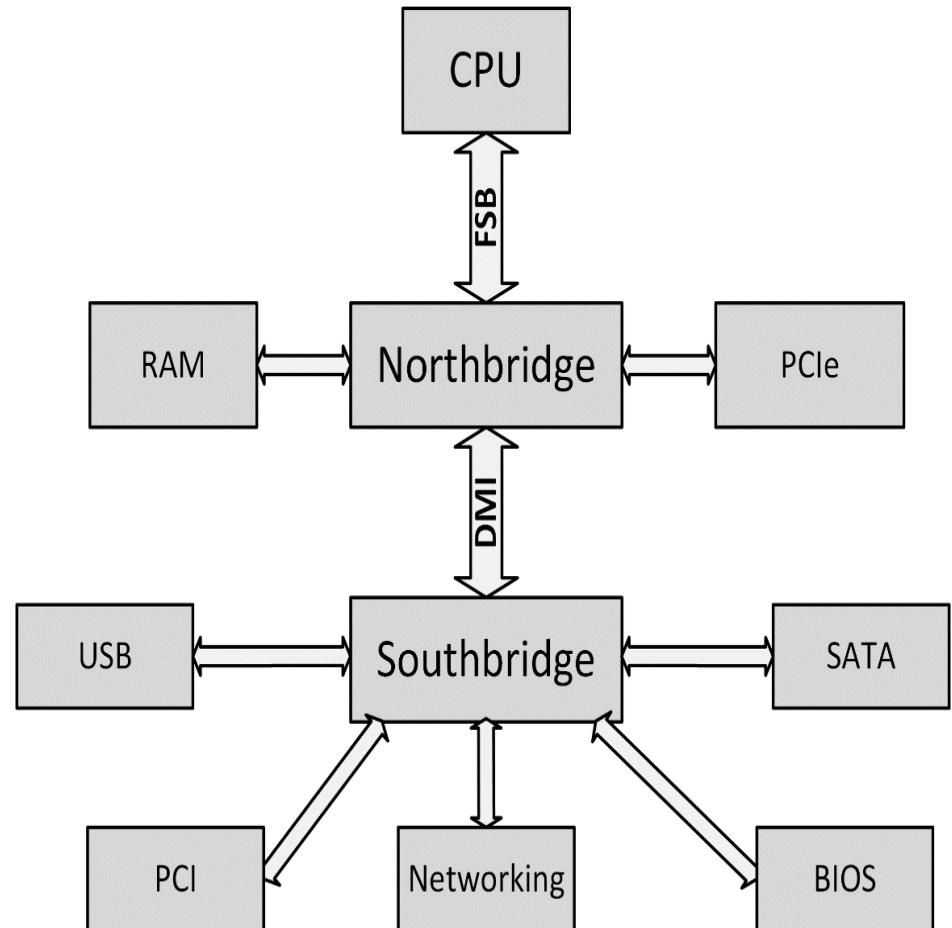
- The x86 platform is the most dominant server architecture today
- In the 1990s, x86 servers first started to appear.
 - They were basically big PCs, housed in 19” racks without dedicated keyboards and monitors
- x86 servers are produced by many vendors, like:
 - HP
 - Dell
 - Lenovo
 - HDS (Hitachi Data Systems)
 - Huawei
- Implementation of the platform is highly dependent on the vendor and the components available at a certain moment
- Most used operating systems are Microsoft Windows and Linux

X86-History

- Most servers in datacenters today are based on the x86 architecture.
- This x86 architecture (also known as PC architecture) is based on the original IBM PC.
- In the 1990s, x86 servers first started to appear.
- They were basically PCs, but were housed in 19" racks without dedicated keyboards and monitors.
- Over the years, x86 servers became the de-facto standard for servers.
- Their low cost, the fact that there are many manufacturers and their ability to run familiar operating systems like Microsoft Windows and Linux, made them extremely popular.

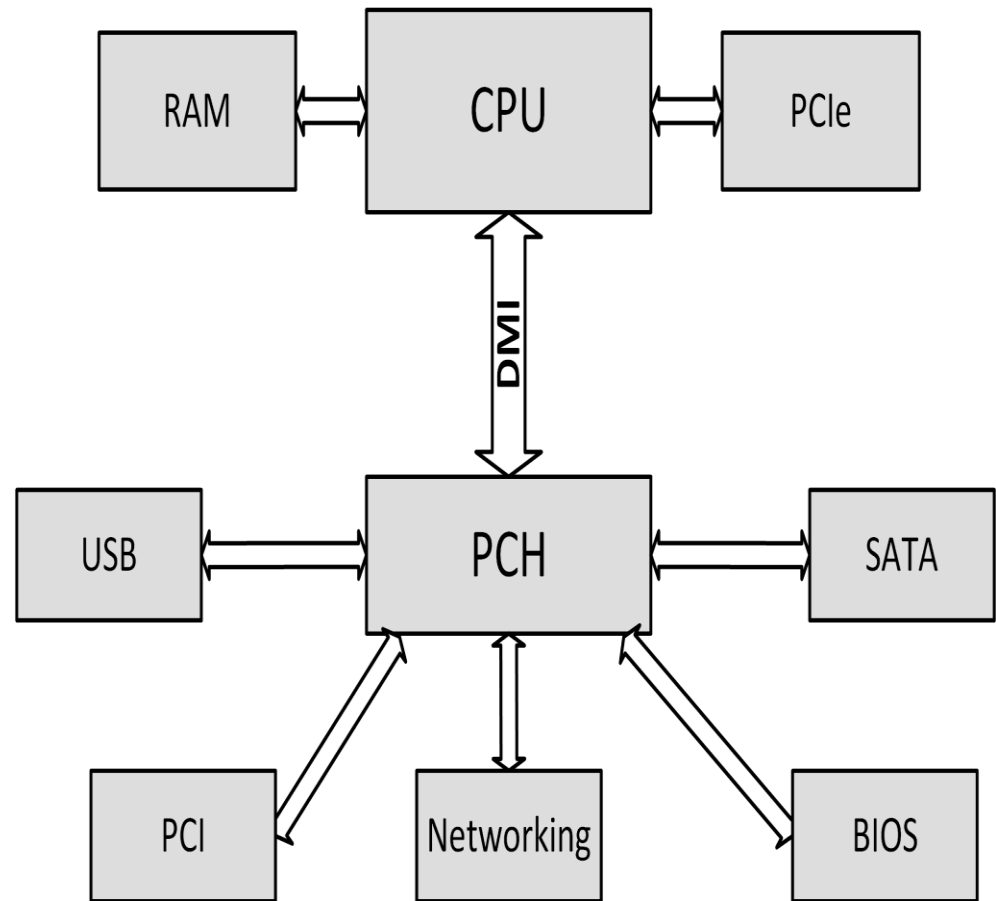
x86 Servers - Architecture

- x86 architectures are defined by a CPU from the x86 family, and building blocks, integrated in a number of specialized chips, known as an x86 chipset
- Earlier x86 systems utilized a Northbridge / Southbridge architecture
 - Front Side Bus (FSB)
 - Direct Media Interface (DMI)



x86 servers - Architecture

- In the PCH architecture, the RAM and PCIe data paths are directly connected to the CPU
- The Northbridge integrated in the CPU
- Intel introduces new architectures and chipsets roughly every two years
- Now full system on a chip (SoC)
- SOC's directly expose:
 - PCIe lanes
 - SATA
 - USB
 - High Definition Video



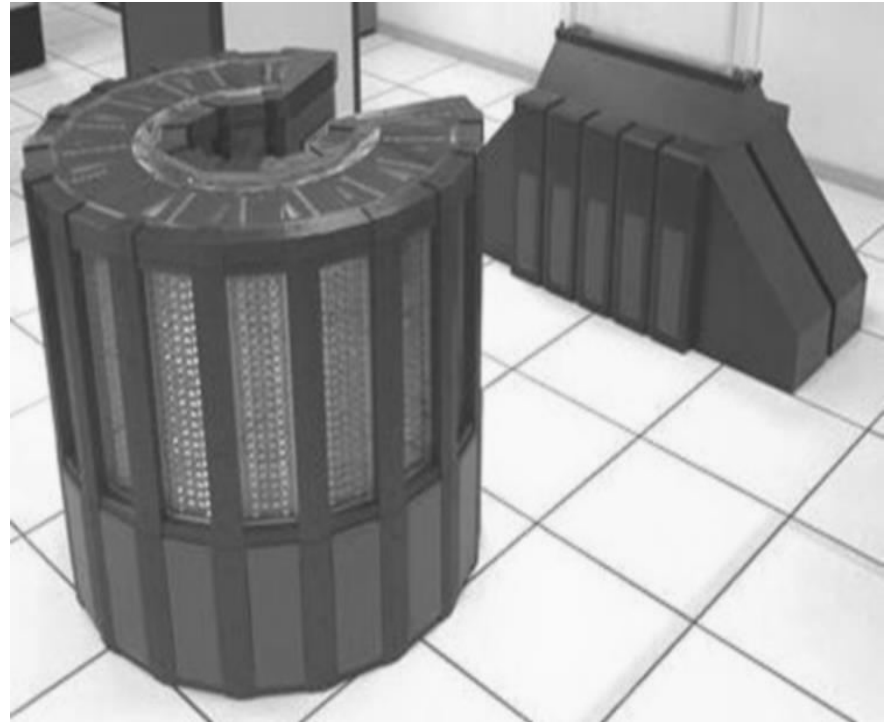
x86 Virtualization

- On x86 platforms, most servers only run one application each
 - A Windows server running Exchange will probably not also run SharePoint
 - A Linux server running Apache will probably not also run MySQL
 - This is the main reason x86 systems use their hardware much less efficient than midrange systems
- By running multiple operating systems – each in one virtual machine – on a large x86 server, resource utilization can be improved
- The most used products for virtualization on the x86 platform are:
 - VMware vSphere
 - Microsoft's Hyper-V
 - Citrix XenServer
 - Oracle VirtualBox
 - Red Hat RHEV

- A supercomputer is a computer architecture designed to maximize calculation speed
 - This in contrast with a mainframe, which is optimized for high I/O throughput
- Supercomputers are the fastest machines available at any given time
- Used for highly compute-intensive tasks requiring floating point calculations, like:
 - Weather forecast calculations
 - Oil reservoir simulations
 - Rendering of animation movies

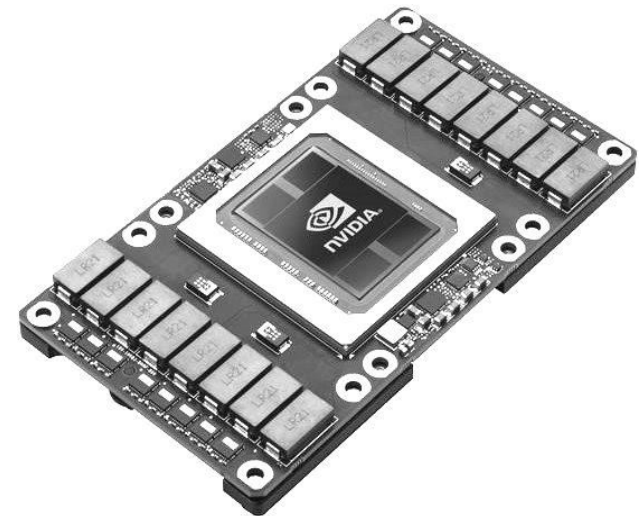
Supercomputers

- Originally, supercomputers were produced primarily by a company named Cray Research
 - Cray-1 (1976): 250 MFLOPS (Million Floating Point Operations per second)
 - Cray-2 (1985): 1,900 MFLOPS



Supercomputers

- Nowadays high performance computing is done mainly with large arrays of x86 systems
- Intel's Core i7 5960X CPU has a peak performance of 354,000 MFLOPS
- The fastest computer array is a cluster with more than 10,000,000 CPU cores, calculating at 125,435,000,000 MFLOPS, running Linux
- Graphics processing units (GPUs) can be used together with CPUs to accelerate specific calculations
- A GPU has a massively parallel architecture consisting of thousands of small, efficient cores designed for handling multiple tasks simultaneously
 - The NVIDIA Tesla GP100 GPU, introduced in 2016, has 3840 cores





Compute Availability

Hot Swappable Components

- Hot swappable components are server components that can be installed, replaced, or upgraded while the server is running
 - Memory
 - CPUs
 - Interface cards
 - Power supplies
- The virtualization and operating systems using the server hardware must be aware that components can be swapped on the fly
 - For instance, the operating system must be able to recognize that memory is added while the server operates and must allow the use of this extra memory without the need for a reboot

Parity Memory

- To detect memory failures, parity bits can be used as the simplest form of error detecting code
- Parity bits enable the detection of data errors
- They cannot correct the error, as it is unknown which bit has flipped

DATA	PARITY
1001 0110	0
1011 0110	1
0001 0110	0 -> ERROR: parity bit should have been 1!

- ECC memory not only detects errors, but is also able to correct them
- ECC Memory chips use Hamming Code or Triple Modular Redundancy (TMR) as the method of error detection and correction
- Memory errors are proportional to the amount of RAM in a computer as well as the duration of operation
 - Since servers typically contain many GBs of RAM and are in operation 24 hours a day, the likelihood of memory errors is relatively high and hence they require ECC memory

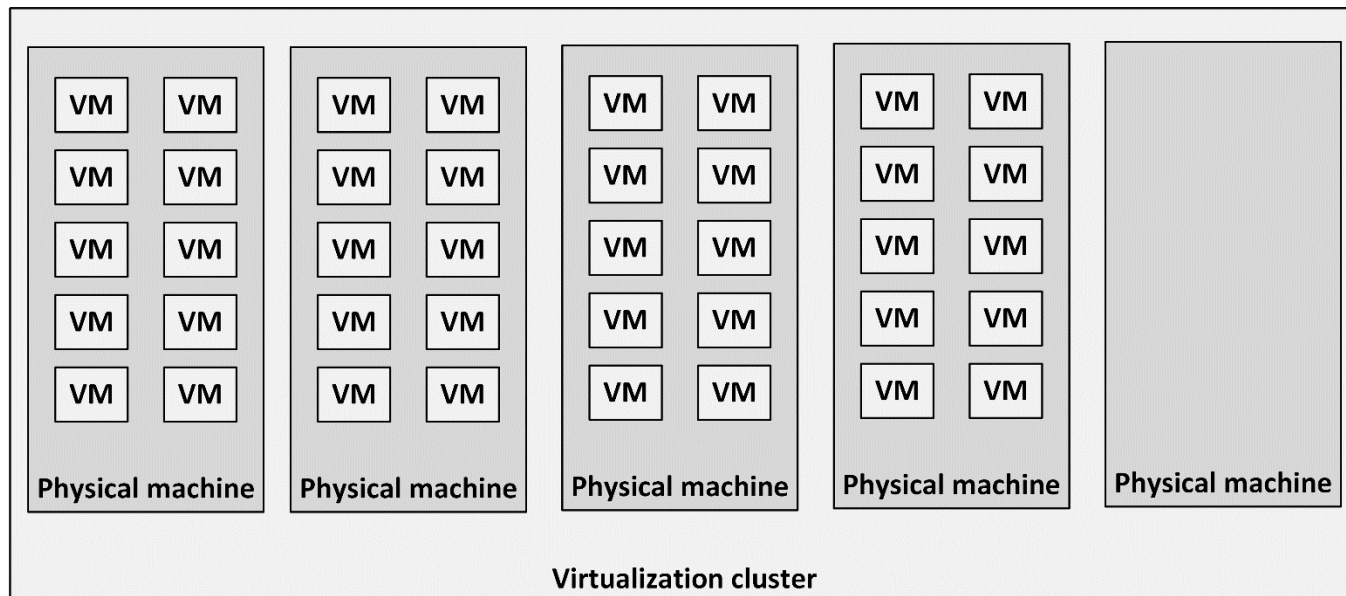
- Lockstepping is an error detection and correction technology for servers
- Multiple systems perform the same calculation, and the results of the calculations are compared
 - If the results are equal, the calculations were correctly performed
 - If there are different outcomes, one of the servers made an error
- Very expensive technology
 - Only used in systems that require extremely high reliability

- All virtualization products provide failover clustering
 - When a physical machine fails, the virtual machines running on that physical machine can be configured to restart automatically on other physical machines
 - When a virtual machine crashes, it can be restarted automatically on the same physical machine

- The virtualization layer has no knowledge of the applications running on the virtual machine's operating system
 - Failover clustering on virtualization level can only protect against two situations:
 - A physical hardware failure.
 - An operating system crash in a virtual machine

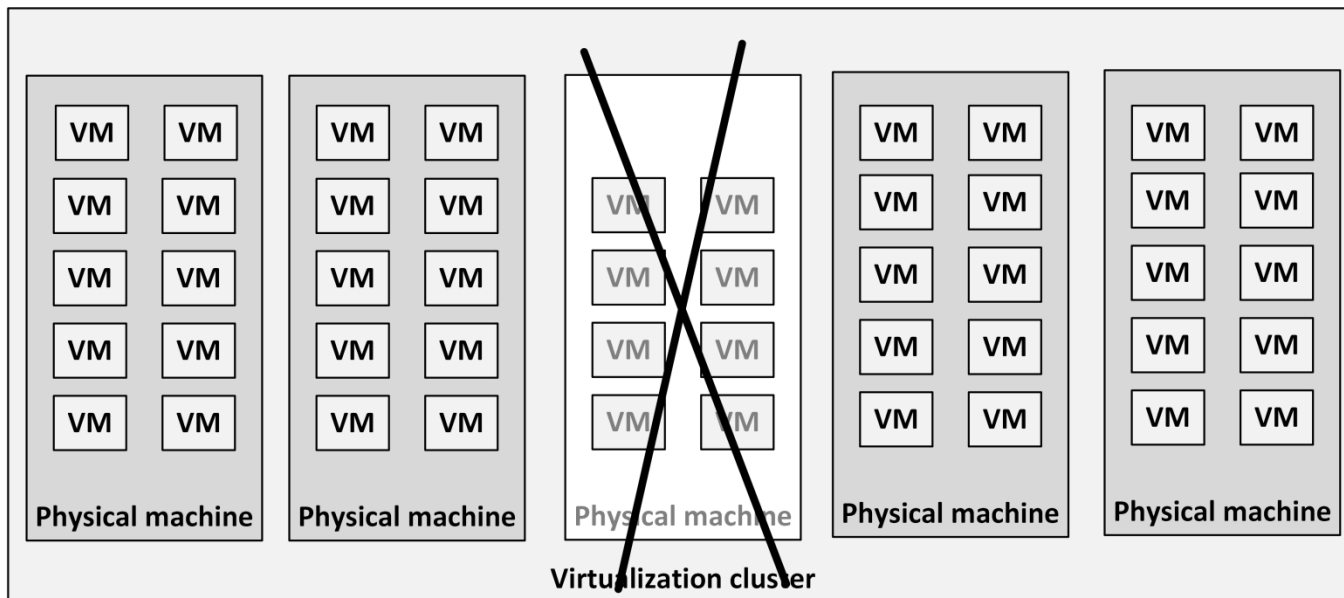
Virtualization Availability

- To cope with the effects of a failure of a physical machine, a spare physical machine is needed
 - All hypervisors are placed in a virtualization cluster
 - The hypervisors on the physical machines check the availability of the other hypervisors in the cluster
 - One physical machine is running as a spare to take over the load of any failing physical machine



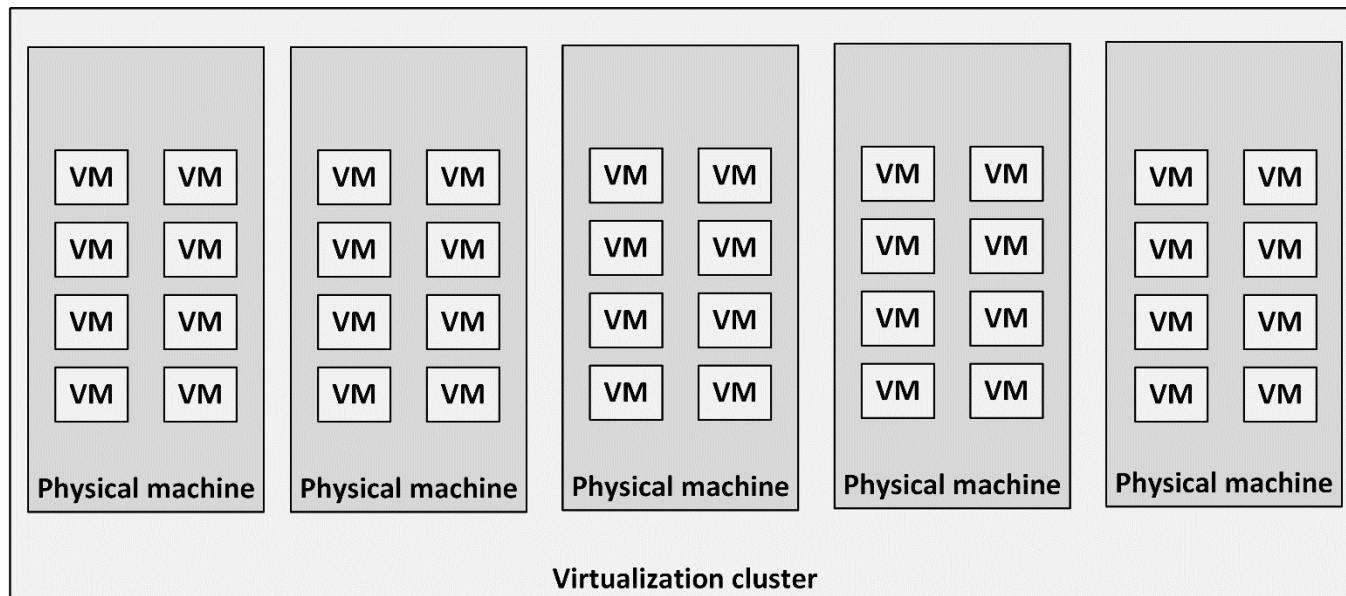
Virtualization Availability

- When a physical machine fails, the virtual machines running on it are automatically restarted on the spare physical machine



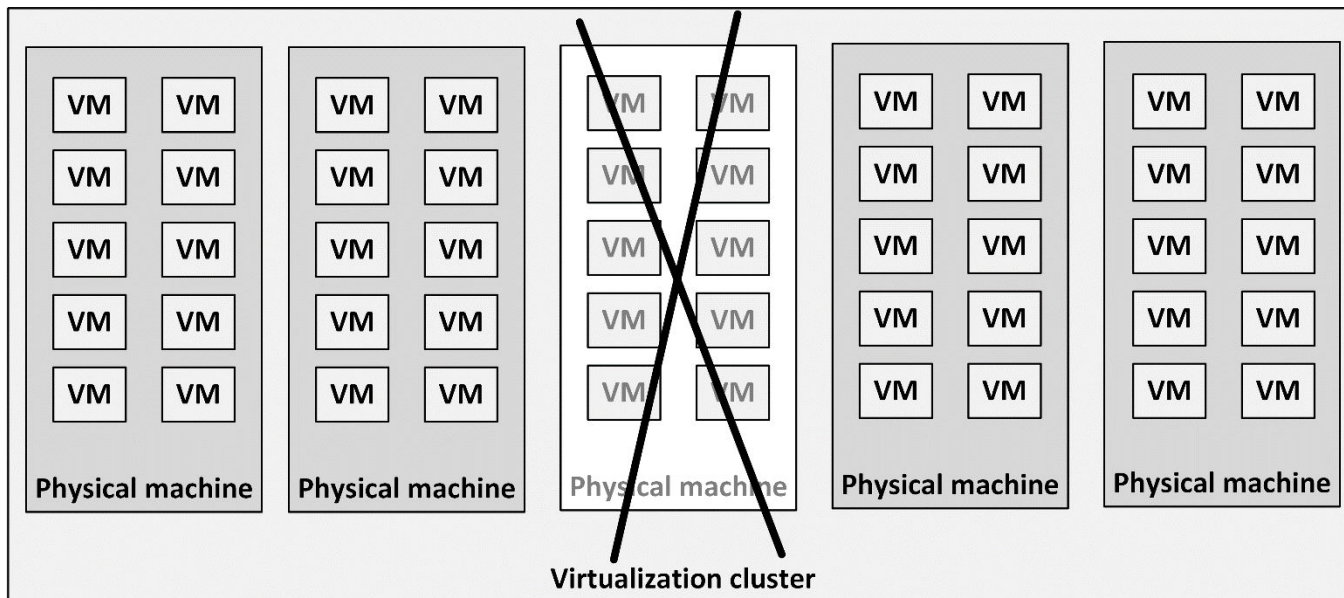
Virtualization Availability

- An alternative is to have all physical machines running at lower capacity



Virtualization Availability

- When a physical machine fails, the virtual machines running on it are automatically restarted on the other physical machine
- All machines now run on full capacity





Compute Performance

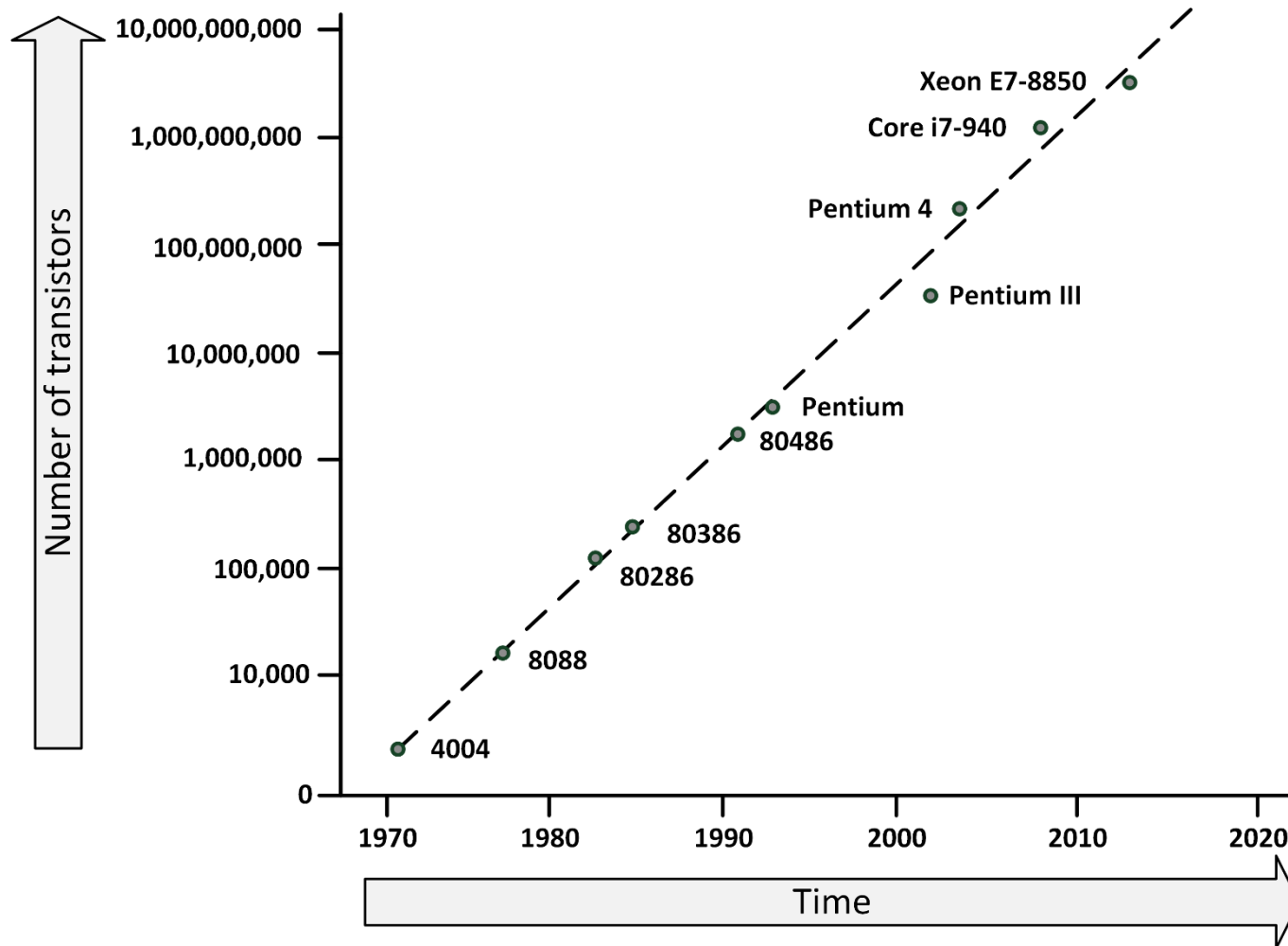
CPU: Moore's law

- In 1971, Intel released the world's first universal microprocessor, the 4004
 - Contained 2,300 transistors
 - Could perform 60,000 instructions per second
 - About as much as the ENIAC computer that filled a complete room and weighed several tons
- Since the introduction of the first CPU in 1971, the power of CPUs has increased exponentially



- Moore's law states:
 - The number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years
- This trend has continued for more than half a century now
- An Intel Broadwell-EP Xeon in 2017 contains 7,200,000,000 transistors
- An 3,100,000-fold increase in 45 years' time!

CPU: Moore's law



Please note that the vertical scale is logarithmic instead of linear, showing a 10-fold increase of the number of transistors in each step

CPU: Moore's law

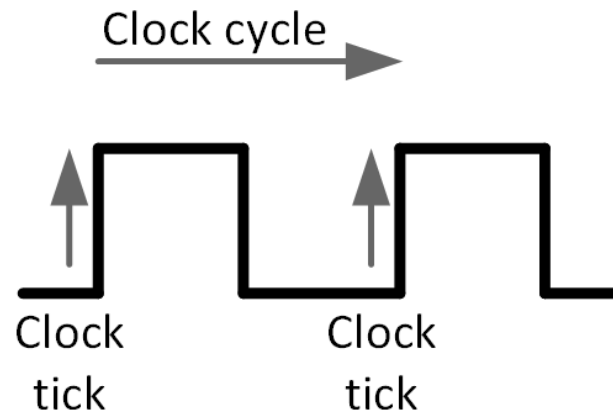
- Moore's law only speaks of the number of transistors; not the performance of the CPU
- The performance of a CPU is dependent on a number of variables, like:
 - Clock speed
 - Use of caches and pipelines
 - Width of the data bus
- Moore's law cannot continue forever, as there are physical limits to the number of transistors a single chip can hold
 - In 2017, connections used inside a high-end CPU had a physical width of 14 nm (nanometer), the size of 140 atoms
 - in 2020, 5 nm CPUs will be produced; traces on the chip are just 50 atoms wide

CPU: Increasing CPU and memory performance

- Various techniques have been invented to increase CPU performance, like:
 - Increasing the clock speed
 - Caching
 - Prefetching
 - Branch prediction
 - Pipelines
 - Use of multiple cores

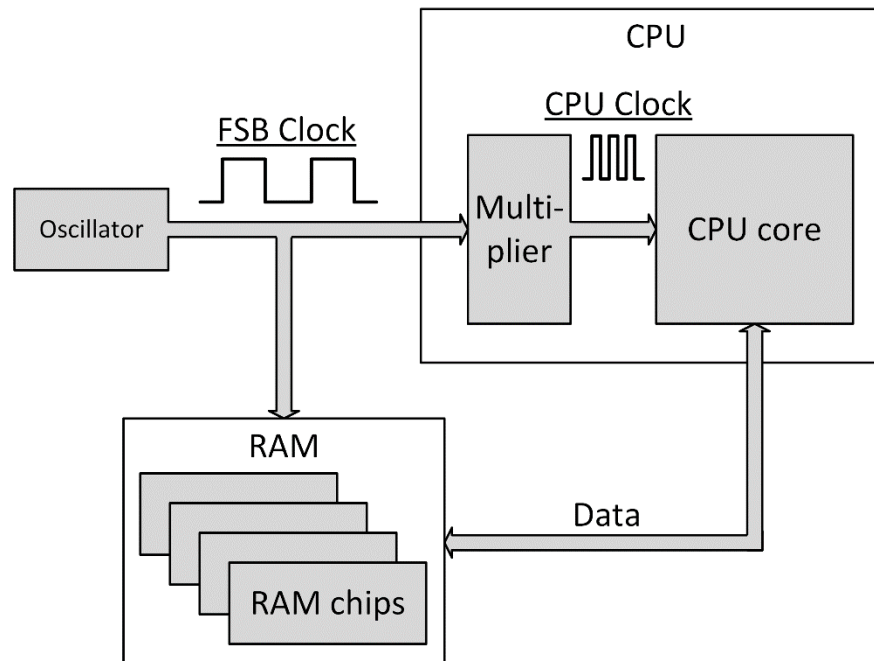
CPU: Increasing clock speed

- CPU instructions need to be fetched, decoded, executed, and the result must often be written back to memory
- Each step in the sequence is executed when an external clock signal is changed from 0 to 1 (the clock tick)
- The clock signal is supplied to the CPU by an external oscillator
- In the early years, CPU clock speed was the main performance indicator
- CPU clock speed is measured in Hertz (Hz) – clock ticks or cycles per second



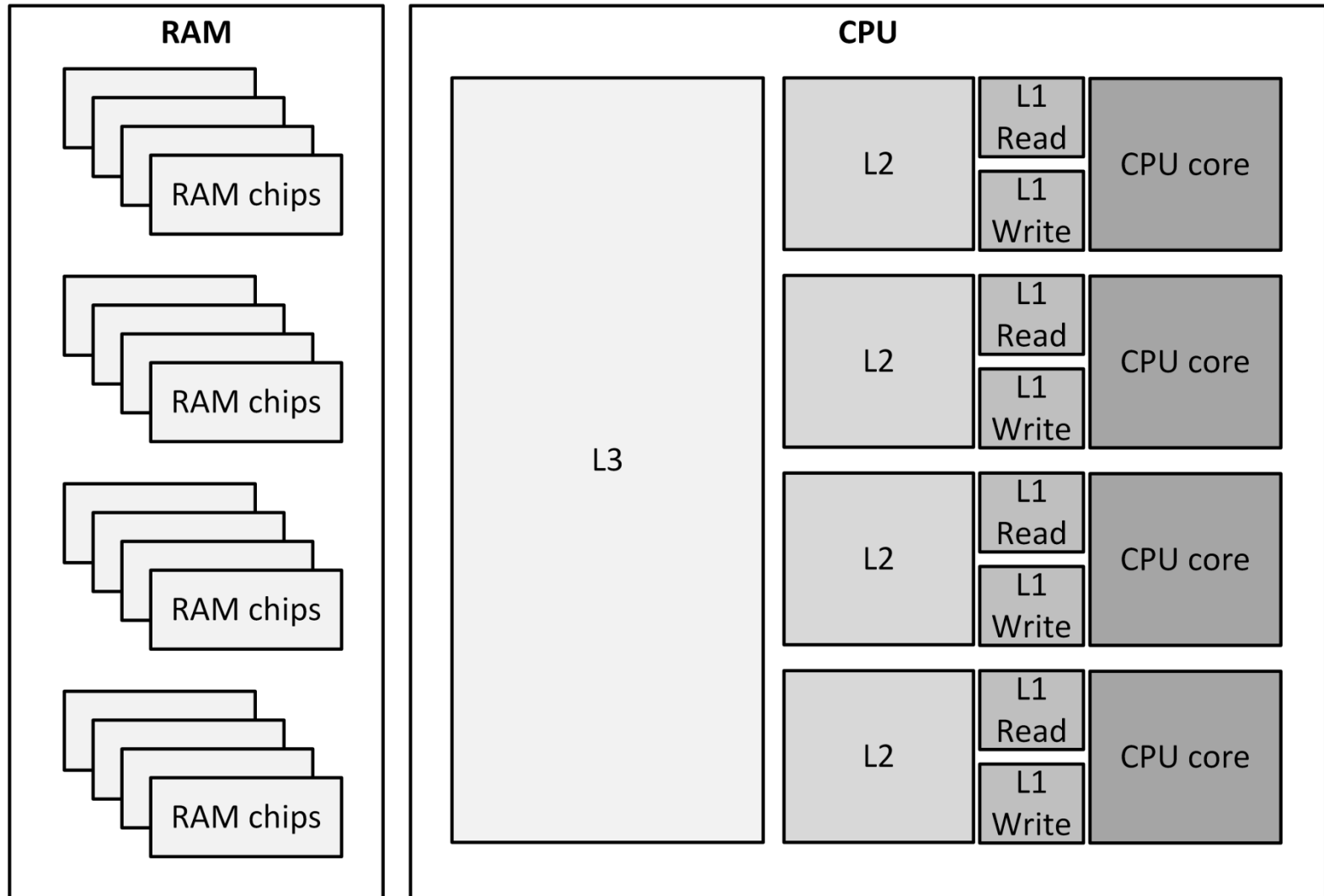
CPU: Increasing clock speed

- Today's CPUs use clock speeds as high as 3 GHz (3,000,000,000 clock ticks per second)
- Because of physical limitations, oscillators cannot run at this speed
- An oscillator with a lower frequency is used (for instance 400 MHz) and this clock rate is multiplied on the CPU chip
- The oscillator speed is known as the Front Side Bus (FSB) speed



- All CPUs in use today contain on-chip caches
- A cache is a relatively small piece of high speed static RAM on the CPU
 - Temporarily stores data received from slower main memory
 - Most CPUs contain two types of cache: level 1 and level 2 cache
 - Some multi-core CPUs also have a large level 3 cache; a cache shared by the cores
- Cache memory runs at full CPU speed (say 3 GHz), main memory runs at the CPU external clock speed (say 100 MHz, which is 30 times slower)

CPU: Caching

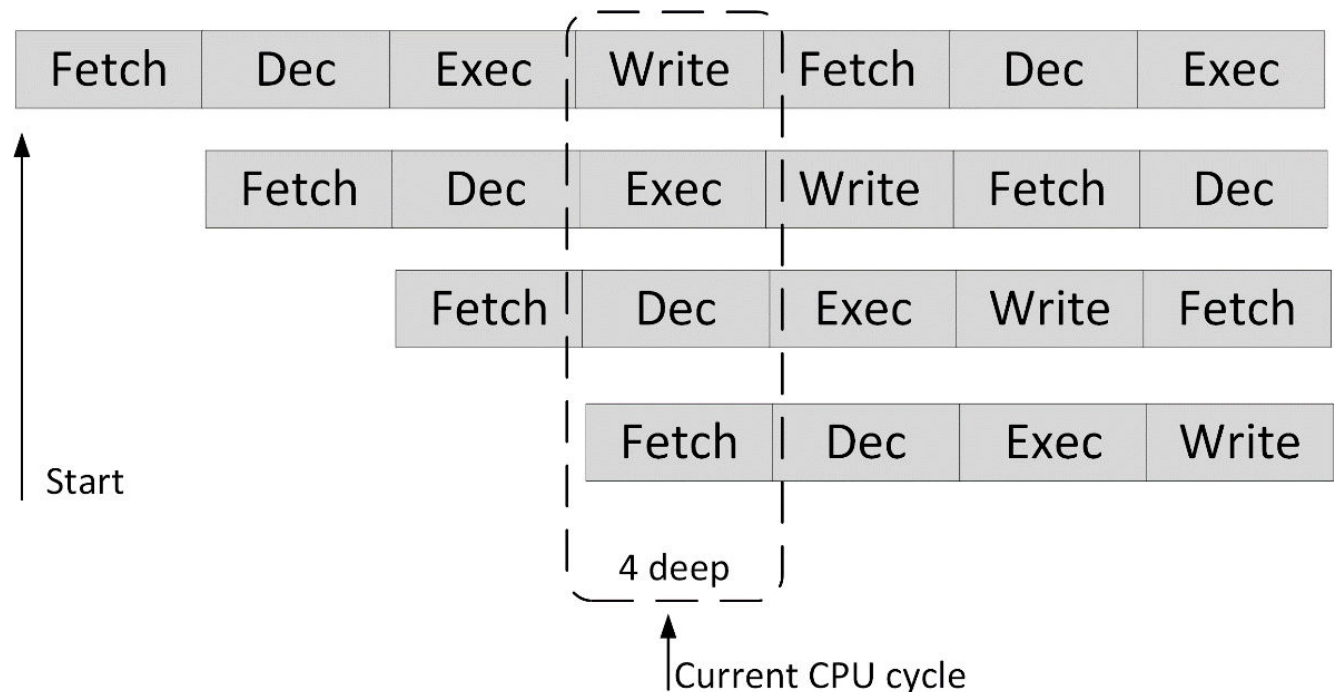


- Early processors first fetched an instruction, decoded it, then executed the fetched instruction, and wrote the result back before fetching the next instruction and starting the process over again



CPU: Pipelines

- Later CPUs used pipelines
- While the first instruction is being executed, the second instruction can be fetched (since that circuitry is idling anyway), creating instruction overlap



CPU: Prefetching and branch prediction

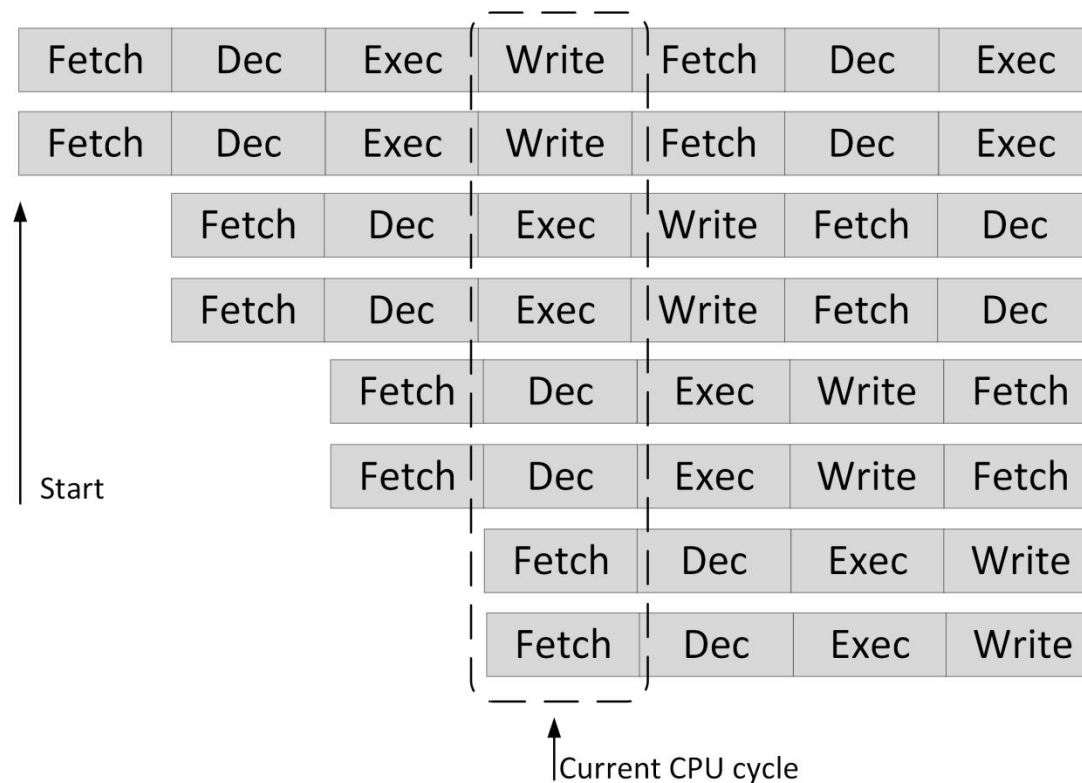
- Prefetching:
 - When an instruction is fetched from main memory, also the next instructions are fetched and stored in cache
 - When the CPU needs the next instruction it is already available in cache
- Unfortunately, most programs contain jumps (also known as branches), resulting in cache misses
 - The next instruction is not the next instruction in memory

CPU: Prefetching and branch prediction

- The cache system tries to predict the outcome of branch instructions before they are executed by the CPU (called branch prediction)
 - In practice more than 80% of the processor instructions are delivered to the CPU from cache memory using prefetching and branch prediction

CPU: Superscalar CPUs

- A superscalar CPU can process more than one instruction per clock tick
- This is done by simultaneously dispatching multiple instructions to redundant functional units on the processor



- The fastest commercial CPUs have been between running between 3 GHz and 4 GHz for a number of years now
- Reasons:
 - High clock speeds make connections on the circuit board work as a radio antenna
 - A frequency of 3 GHz means a wavelength of 10 cm. When signals travel for more than a few cm on a circuit board, the signal gets out of phase with the clock
 - The CPU can heat up tremendously at certain spots, which could lead to a meltdown

- A multi-core processor is a CPU with multiple separate cores
 - The equivalent of getting multiple processors in one package
- The cores in a multi-core CPU run at a lower frequency
 - Reduce power consumption
 - Reduce heat (no hot spots)
- Trend is to have CPUs with tens or even hundreds of cores

- Certain Intel CPUs contain a propriety technology called hyper-threading
 - For example the Core i3/i5/i7 and Xeon CPUs
- Hyper-threading makes a single processor core virtually work as a multi-core processor
- Hyper-threading can provide some increase in system performance by keeping the processor pipelines busier

Virtualization performance

- Consolidating multiple virtual machines on one physical machine increases CPU usage and reduces CPU idle time
 - This is a primary driver for the use of virtualization
- The physical machine needs to handle the disk and network I/O of *all* running virtual machines
 - This can easily lead to an I/O performance bottleneck

Virtualization performance

- When choosing a physical machine to host virtual machines, consider getting a machine with:
 - Much CPU and memory capacity
 - Capability of very high I/O throughput
- Virtualization introduces performance penalties:
 - Resources required to run the hypervisor
 - Operation transformations
- This is usually less than 10% of the total performance

- Databases generally require a lot of network bandwidth and high disk I/O performance
 - This makes databases less suitable for a virtualized environment
 - Raw Device Mapping allows a virtual machine exclusive access to a physical storage medium
 - This diminishes the performance hit of the hypervisor on storage to almost zero

- Often one physical server is needed per database
 - The physical server runs just one virtual machine
 - Many benefits of virtualization remain
 - Database servers can easily be migrated to other physical machines without downtime
 - Management of the servers is unified when all servers run hypervisors



Compute Security

Physical Security

- Disable external USB ports in the BIOS
- BIOS settings in an x86 server should be protected using a password,
 - Via the BIOS, external USB ports can be enabled, and other parameters can be set
- Some servers allow the detection of the physical opening of the server housing
 - Such an event can be sent to a central management console using for instance SNMP traps
 - If possible, enable this to detect unusual activities

- Virtual machines must be protected the same way as physical machines
- The use of virtualization introduces new security vulnerabilities of its own:
 - If possible, firewalls and Intrusion Detection Systems (IDSs) in the hypervisor should be deployed
 - The virtualization platform itself needs patching too
 - The size and complexity of the hypervisor should be kept to a minimum
- DMZ
 - Consider using separate physical machines that run all the virtual machines needed in the DMZ

- Systems management console
 - The systems management console connects to all hypervisors and virtual machines
 - When the security of the systems management console is breached, security is breached on all virtual machines
 - Not all systems managers should have access to all virtual machines
 - Special user accounts and passwords should be configured for high risk operations like shutting down physical machines or virtualized clusters
 - All user activity in the systems management console should be logged

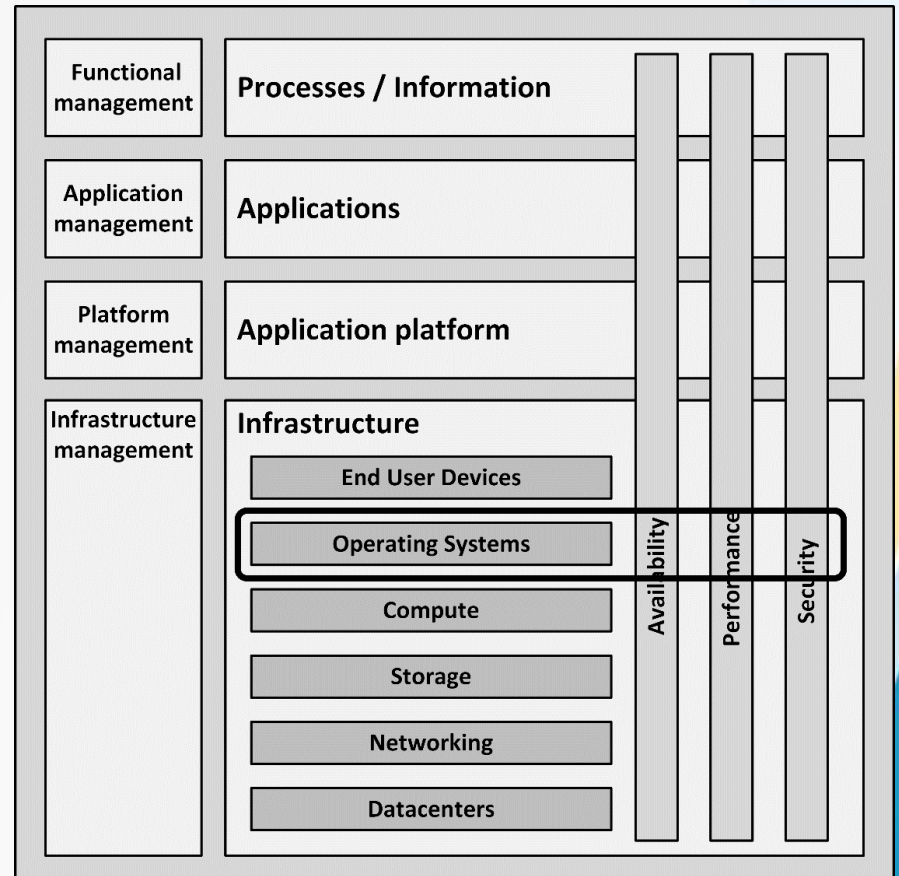
IT Infrastructure Architecture

Infrastructure Building Blocks and Concepts

Operating Systems – Part 1 (chapter 11)

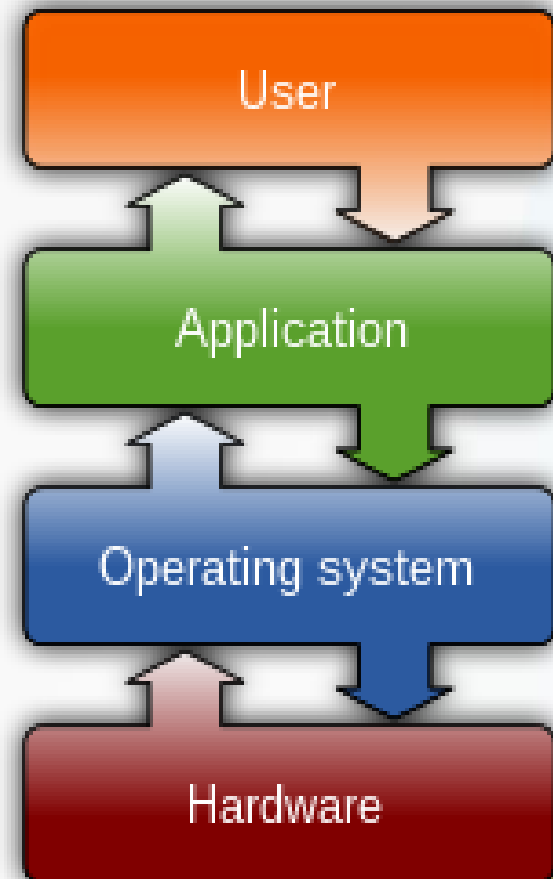
Introduction

- An operating system is the set of programs that, after being initially loaded into a computer by a boot program, controls all the other programs in a computer
- It is an intermediary between hardware and programs
- An **operating system** (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs.



Introduction

- Operating systems manage a computer's internal workings
 - Memory
 - Processors
 - Internal and peripheral devices
 - File system
- Operating systems are used on for instance::
 - PCs and laptops
 - Virtual machines
 - Servers
 - Tablets and mobile phones
 - Network routers
 - Storage arrays
 - Cars
 - Televisions



Introduction

- Operating systems provide an abstraction layer between (virtualized or physical) hardware and software applications
- As a service to applications, low level hardware management is handled by the operating system
 - Process management
 - Memory management
 - Interrupt handling
 - Multi user management
 - File locking and file sharing

Introduction

- Operating systems provide services to applications in the form of Application Programming Interfaces (APIs)
- For example for:
 - File management
 - I/O interfaces (like video and keyboard)
 - Hardware drivers (like printer drivers)

History

- Early operating systems could execute one program at a time
- At any given time, one user had sole use of the computer
- Through the 1950s, many major features were pioneered in the field of operating systems, including multitasking
- During the 1960s, IBM introduced a single operating system (OS/360) for all of its mainframes
- In 1969, UNIX was created
 - UNIX used a file system with directories
 - Standard POSIX system calls

History

- An early operating systems for personal computers was CP/M
 - CP/M introduced drive letters (c:), but no directories
- Parts of CP/M and UNIX were imitated in MS-DOS
 - Drive letters and directory structures
 - System calls
 - MS-DOS became extremely popular when chosen as the default operating system for the IBM PC
- The most popular operating systems running on servers today are Microsoft Windows, Linux, and UNIX

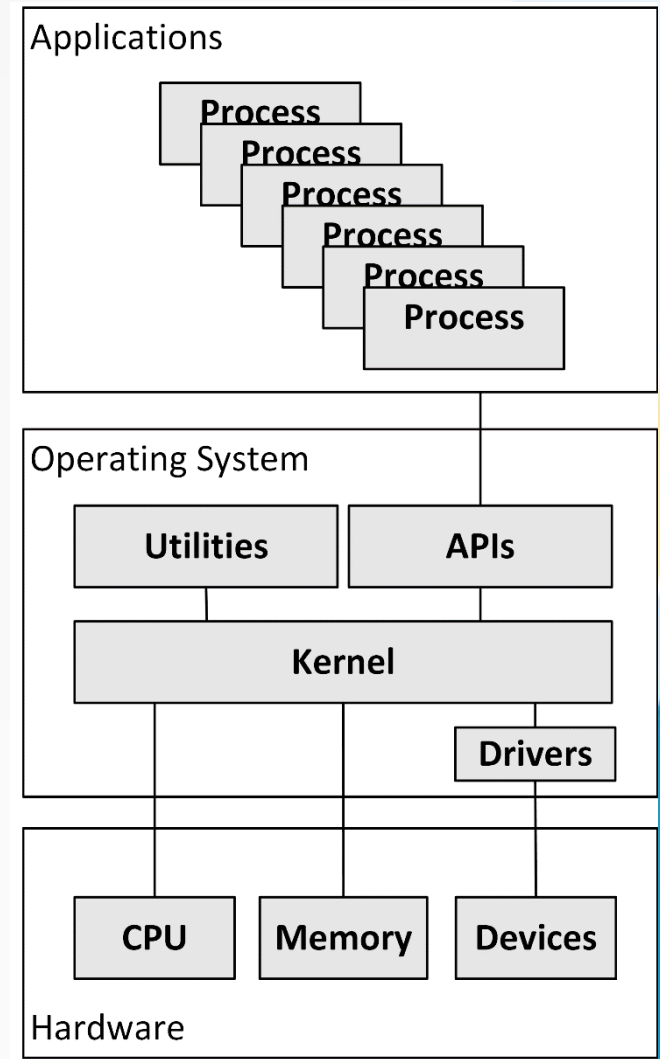
Operating System Building Blocks

Operating System Building Blocks

- An operating system basically performs two basic operations:
 - It enables multiple users, multiple processes, and multiple applications to run together on a single piece of hardware
 - It hides the technical complexities of the underlying hardware from the applications running on top of the operating system

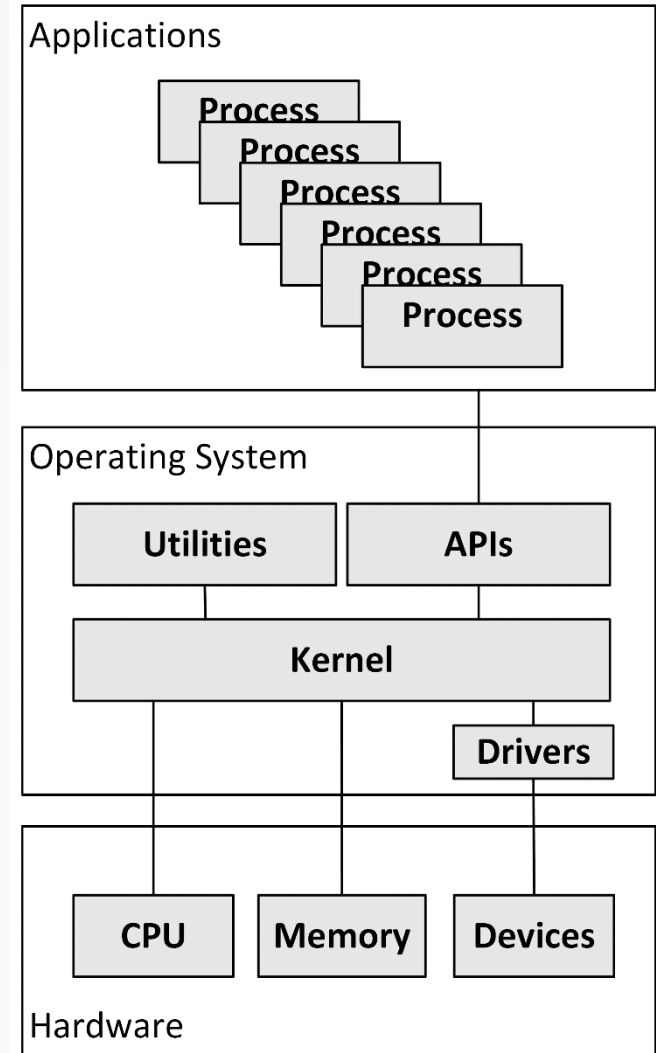
Operating System Building Blocks

- The kernel is the heart of an operating system
 - Starts and stops programs
 - Manages the file system
 - Performs low level tasks that most programs need
 - The kernel schedules access to hardware to avoid conflicts if two programs try to access the same resource or device simultaneously



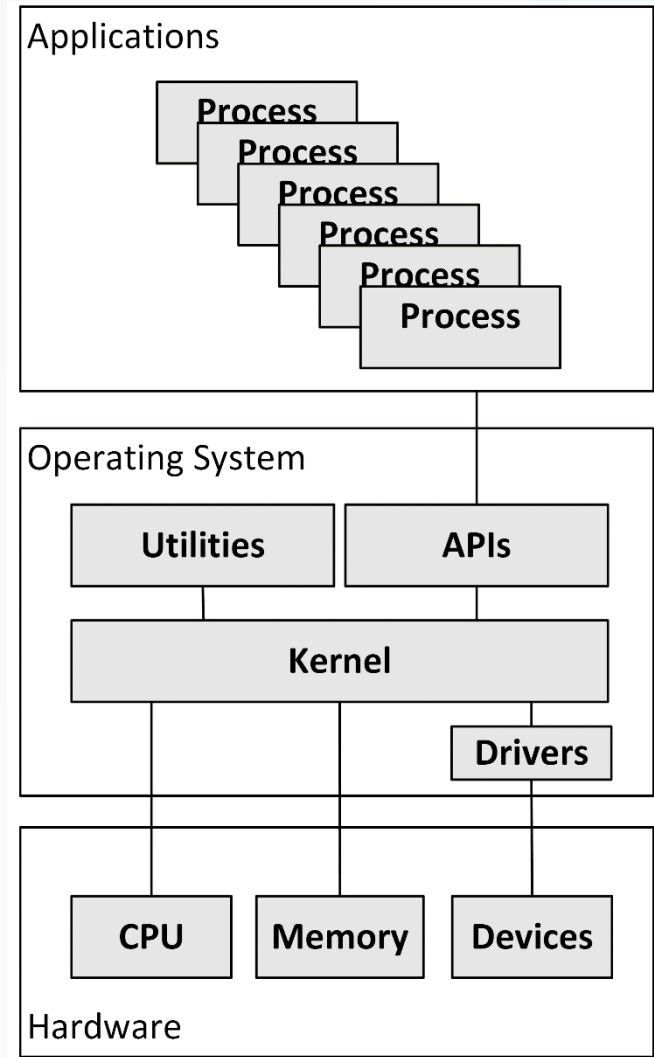
Operating System building blocks

- Drivers are small applications that connect specific hardware devices to the kernel
 - Printers
 - Network cards
 - Keyboard and mouse
 - Video screens
- Utilities are applications that are considered part of the operating system
 - User interfaces
 - Logging tools
 - Editors
 - System update processes



Operating System building blocks

- Applications consist of one or more processes that communicate with the operating system using system calls that are invoked through Application Programming Interfaces (APIs)



Process Scheduling

- Operating systems create the illusion of multiple running processes in parallel by scheduling each process to run only during a short time frame
 - This principle is also known as preemptive multitasking
 - Periodically, the operating system decides if a running process is to be suspended in favor of another process, or if the running process can keep on running for a while
- Process scheduling is fairly complex
 - Must be well-balanced
 - Switching processes introduces some overhead
 - The scheduling algorithm guarantees each process gets its fair share of CPU time
 - Because operating systems have evolved over decades, scheduling algorithms are very sophisticated

File Systems

- The operating system provides a file system to applications
 - File systems usually consist of directories (also known as folders) with files or other directories
- The operating system hides complexity
 - Handling individual disk blocks or communication with a SAN or NAS
 - Managing the files and the directory structure
 - Security – permission to read, write, create, and delete files and directories

File Systems

- Most operating systems can handle multiple types of file systems on multiple disks at the same time
- Some popular file systems are:
 - FAT (File Allocation Table), vFAT, and FAT32, used in MS-DOS, older versions of Windows, and removable storage devices like USB memory sticks
 - NTFS (New Technology File System) used in Windows
 - UFS (Universal File System) and VxFS (Veritas File System) used in most UNIX flavors
 - Ext (and Ext2, Ext3, Ext4) - used in Linux

File Systems

- Journaling file systems keep track of changes made to files in a journal log before committing them to the main file system
 - Higher availability
 - Fast recovery in case of a malfunction
- File systems must be mounted before they can be used by the operating system
 - A disk and the file system on it must be recognized by the operating system and attached to it
- After mounting, the file system is typically given either:
 - A drive letter (Windows)
 - A drive name (OpenVMS)
 - A mount point in the global directory tree (UNIX and Linux)

File Systems

- Most operating systems provide file sharing functionality
 - File sharing enables files on one system to be accessed by (users on) other systems
 - File sharing protocols:
 - NFS: originates from UNIX
 - SMB/CIFS: originates from Windows

APIs and System Calls

- System calls are programming functions
 - Provide a hardware-independent interface to tasks the operating system can perform for applications
- Example:

```
int read(int handle, void *buffer, int  
nbyte) ;
```

translates into the system call

```
READ (FILEHANDLE, DESTINATION DATA POINTER,  
NUMBER_OF_BYTES)
```

APIs and System Calls

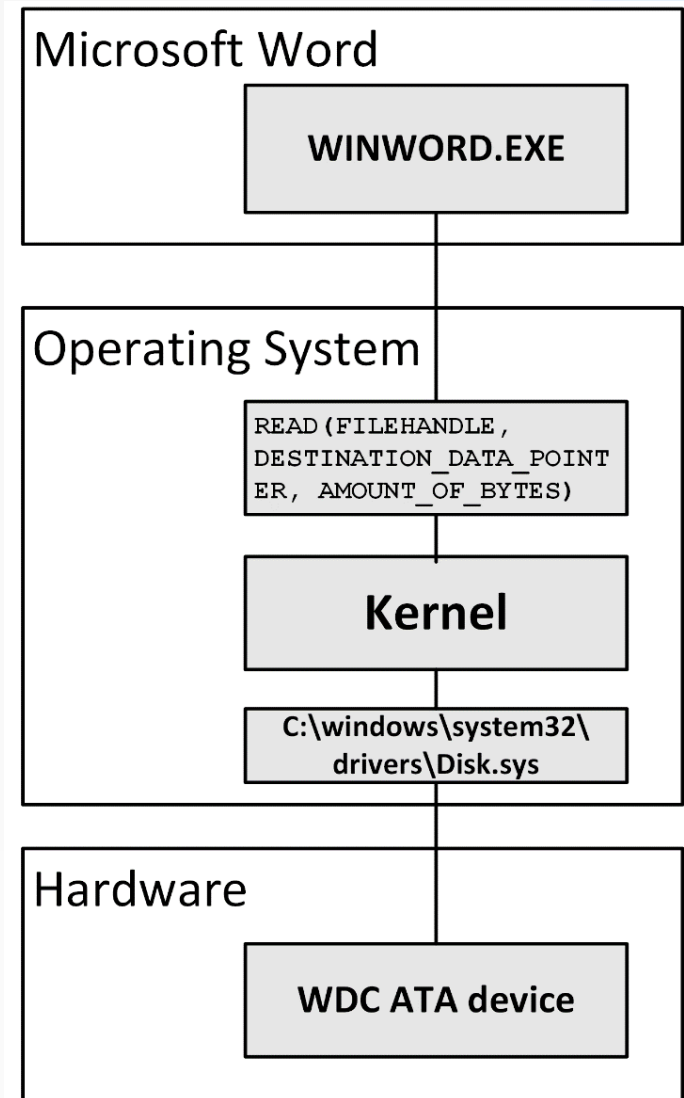
- The operating system takes care of:
 - Looking-up the file in a file allocation table
 - Looking up the disk blocks on disk
 - Instructing the disk controller to fetch the needed disk blocks
 - Copy the disk blocks to memory
 - Providing a pointer to the disk blocks in memory

APIs and System Calls

- System calls are grouped and presented to application processes as Application Programming Interfaces (APIs)
- APIs describe the available system calls in an operating system and how they can be used by programmers in their applications
- Each operating system has its own API
 - UNIX and Linux use the POSIX standard
 - Windows has its own API

Device Drivers

- The operating system manages all hardware
- I/O devices are controlled using device drivers
 - Pieces of software that interact with the device's hardware (like an Ethernet network adapter or a SAS disk adapter)
 - They provide an Application Programming Interface (API) to the operating system



Memory Management

- The operating system:
 - Allocates and de-allocates memory on behalf of applications
 - Manages what happens when the amount of requested memory exceeds the physical amount of memory
- Memory management includes:
 - Cache management
 - Paging
 - High volume data transfers
 - Memory management units (MMUs)
 - Thin memory provisioning (memory overcommitting)
 - Direct Memory Access (DMA)
- The operating system takes care of all of this and just provides chunks of memory to applications

Shells, CLIs and GUIs

- A shell provides a user interface to the operating system
 - The primary purpose of shells is to launch other programs, started by end users or scripts
- Two types of shells:
 - Command-Line Interfaces (CLIs)
 - The user types commands on a keyboard on a command-prompt
 - Examples: UNIX shells (bash, sh, csh) and Windows' cmd.exe (also known as a DOS box)
 - Graphical User Interfaces (GUIs)
 - The user uses a mouse to click on icons or buttons
 - Examples Microsoft Windows and X Windows (UNIX and Linux)

Operating System Configuration

- The configuration of an operating system is stored in an operating system specific database or in text files
- Examples:
 - Windows registry
 - Files in the Linux `/etc` directory
 - AIX Object Data Manager (ODM) database
- For most used configuration parameters, user-friendly tools are provided
 - These tools still edit the text files, but that is hidden from the user

Popular Operating Systems

IBM z/OS

- The most used mainframe operating system
- Extreme backward compatibility is one of z/OS's main design philosophies
 - Programs written for MVS in 1974 can still run on today's z/OS without modification
- Typical use of z/OS:
 - Batch processing: reading and writing large amounts of data and performing relatively simple calculations on it
 - Interactive users: supports thousands of interactive users

IBM i (OS/400)

- IBM i is an operating system only used on IBM's Power Systems midrange systems
 - The operating system was previously known as OS/400
 - The midrange system was previously known as AS/400
- One of the biggest advantages of IBM i is its completeness, it includes:
 - Communications
 - Transaction processing
 - Relational database manager
 - Features for the implementation and maintenance of data security
- The latest version is "IBM i 7.2", released in 2014

OpenVMS

- OpenVMS is an operating system developed by DEC
 - VMS means Virtual Memory System
 - OpenVMS is now maintained by VMS Software, Inc.
- OpenVMS is a multi-user operating system designed for:
 - Time sharing
 - Batch processing
 - Real-time processing
 - Transaction processing
- OpenVMS is not open source software
 - The source listings are available for purchase

OpenVMS

- OpenVMS is typically used in environments where system uptime and data access is critical
- It is used for various applications, including:
 - Mail servers
 - Network services
 - Manufacturing
 - Transportation control and monitoring
 - Critical applications
 - Databases
- OpenVMS for HP Integrity servers based on Intel Itanium 9500 processors was released in 2015

UNIX - History

- UNIX is a multitasking, multi-user operating system
- Created by Bell Labs (now AT&T) in 1969
- In 1973, UNIX was rewritten in the new C programming language
 - C was created by the same people that created UNIX
 - This made UNIX portable to multiple types of computer hardware
- In 1982, UNIX was licensed to a number of computer manufacturers
 - Most marketed their own UNIX versions based on the original UNIX source code
 - They adapted the code to meet their own hardware and software requirements

UNIX – vendor versions

- Vendors came up with different names for their UNIX flavors
- These versions are 90% the same, but have some minor differences
 - Wording of error messages
 - The order of commands used to start-up the machine
 - The location of certain files

Vendor	UNIX flavor
IBM	AIX
Oracle/Sun	Solaris
HP	HP-UX
Apple	Mac OS X (built on FreeBSD, discussed in the next section)

UNIX – vendor versions

- Each flavor needs specific hardware
 - HP-UX only runs on HP Integrity systems
 - HP systems cannot run for example IBM's AIX
- Applications running on a particular flavor of UNIX cannot run on another flavor without (at least) recompiling
 - Software vendors must provide separate versions of their applications for each flavor of UNIX

UNIX – vendor versions

- UNIX has been ported to a wider variety of machine architectures than any other operating system
 - UNIX is written almost entirely in the C programming language
 - Source code is published
- Many business software today is released on Linux before being released on the various flavors of UNIX
 - Linux runs on many hardware platforms without recompiling
 - Linux is now more widely used than UNIX

UNIX - filesystem

- UNIX popularized the hierarchical file system with nested subdirectories – the directory tree
- All files and directories appear under the so-called root directory "/"
 - Even if they are stored on different physical disks
- UNIX has no concept of drive letters, like Windows or DOS
 - Drives are mounted on a branch in the directory tree, providing disk space for that particular branch
 - Also removable drives must be mounted in the tree

UNIX – system tools

- The UNIX philosophy is to use a large set of small tools that do only one thing, and do it very well
- To perform complicated tasks, commands can be combined using a system called pipes
 - Pipes feed the output of one command to the input of another command, without storing the intermediate result
 - For instance, the UNIX command: `ls | sort` prints a sorted list of files on the screen
 - The pipe sign “|” sends the output of the “ls” command as input to the “sort” command
- In practice, these chains of piped commands can get very long and complex

Linux - history

- Linux is a free UNIX-like operating system for the x86 platform
- It is not derived from the UNIX source code
- In 1991, Linus Torvalds wanted to explore the multitasking possibilities of the new Intel 80386 CPU in his PC
- He decided to create a small multitasking, multi-user operating system with the help of the internet community
- Because of the open source nature of Linux many developers contributed
 - Kernel patches
 - Device drivers
 - Multilingual keyboards
 - Disk drivers
 - Support for video card devices

Linux

- Linux commands and scripts are almost similar to those of UNIX
- Linux also uses the same commands, file structure, scripting language, pipes, etc. as UNIX
- Today Linux is a very mature operating system, used in:
 - Servers
 - Workstations
 - Mobile devices
 - All Android smartphones
 - Appliances like set-top boxes, firewalls and NAS devices
- Ninety-five per cent of the supercomputers listed in the top 500 list of the fastest computers in the world are running Linux
- Almost all internet services run on Linux

Linux – GNU/Linux

- The GNU project was launched in 1984 by Richard Stallman
 - Goal was to develop a free UNIX-like operating system
 - GNU is a recursive acronym for “GNU's Not UNIX!”
 - By 1990, the GNU project had recreated all the major components of the UNIX-like system except one – the kernel
- Combining Linux with the GNU system resulted in a complete operating system: the GNU/Linux system
 - It is important to understand that Linux is actually only an operating system *kernel*
 - What we call Linux, is actually GNU/Linux
- Linux and the GNU tools are licensed under the GNU General Public License
 - Ensures all source code will be free for all to copy, study, and to change

Linux – distributions

- Vendors compiled the Linux source code, added some tools and configurations of their own, and releasing it in a distributable format
- Some of the best-known Linux distributions:
 - Red Hat
 - SuSe
 - Ubuntu
 - Debian

Linux - Support

- Linux can be downloaded from the internet for free
- Most organizations demand professional support for their software
- Professional support is not free
- Most Linux distribution vendors, like Red Hat and SuSe, and some independent vendors, offer support contracts for Linux

Linux on mainframes

- Some Linux distributions can be used on IBM mainframes, running in virtual machines
- Linux uses X-Windows emulators on PCs to connect to the mainframe
- The emulated LAN on the mainframe can be used to connect multiple Linux virtual machines and to provide an external LAN route

BSD

- Berkeley Software Distribution (BSD) is a UNIX operating system derivative
 - Developed and distributed by the University of California, Berkeley
- BSD was the basis for three open source development projects:
 - FreeBSD
 - Most widely used
 - FreeBSD is a complete operating system (Linux is only a kernel!)
 - NetBSD
 - Ported to 57 hardware platforms across 15 different processor architectures
 - Often used in embedded systems

BSD

- OpenBSD
 - Most secure BSD version
 - Has a tradition in which developers audit the source code for software bugs and security problems
 - In the 10+ years of its existence, only three security bugs have been found in OpenBSD
- Darwin, the system on which Apple's Mac OS X is built, is a derivative of FreeBSD

IT Infrastructure Architecture

Infrastructure Building Blocks and Concepts

Operating Systems – Part 2
(chapter 11)

Windows - introduction

- * Windows is a popular x86 operating system, used on PCs and servers
- * Because of Window's popularity, a large collection of software is available
 - * Microsoft provides a fairly complete stack of business solutions like SharePoint, BizTalk, SQL Server, and Exchange
 - * They also provide a development environment (Visual Studio and the .Net framework)
 - * Microsoft Azure cloud runs on a slimmed down version of Windows
- * Many organizations have a "Microsoft unless" strategy
 - * Software is purchased from Microsoft or built using Microsoft tools, unless there is no solution from Microsoft available

Windows for desktops

- * The first version of Microsoft Windows was released in 1985
- * Early Windows versions ran as an application on top of MS-DOS
 - * Windows was no real operating system
- * In 1990, Microsoft Windows 3.0 was the first successful Windows version
- * In late 1995, Microsoft released Windows 95, positioned as the new operating system for desktops
 - * Windows 95 introduced the "start" button
- * Windows targeted at workstations include:
 - * Windows XP
 - * Windows Vista
 - * Windows 7, 8 and 10

Windows for servers

- * In 1992, Windows NT was the first version of Windows designed to run on servers
 - * A real operating system, not running on top of MS-DOS
- * Windows NT 4
 - * Included the Windows 95 style GUI
 - * Companies started the switch from Novell servers to Windows NT 4
 - * Some UNIX systems were being replaced by Windows NT 4 systems
- * Windows 2000 introduced an implementation of LDAP directory services, called Active Directory
- * The server operating systems were named after the year of release: Windows server 2003, 2008, 2012, and 2016

Windows – Stability

- * While NetWare and UNIX would run for at least a year without crashing, it was not uncommon that a Windows server crashed once a day
- * Causes:
 - * The need for backwards compatibility of Windows
 - * Every version of Windows needed to be able to run all already developed software without recompilation
 - * Windows runs on all kinds of hardware
 - * As opposed to UNIX or Apple systems, which are designed for specific hardware
 - * The quality of third-party drivers was not always guaranteed

Windows – Security

- * Windows security was weak
 - * Windows was based on MS-DOS – a single user / single tasking operating system
 - * Multi-user features and concurrently running multiple applications was built in later
 - * Most Windows applications were not designed with multi user usage in mind
 - * Applications had to run with the highest possible user permissions (administrator rights)
 - * This led to the rise of viruses and worms attacking Windows

Windows – Security

- * The Trustworthy Computing Initiative
 - * In 2002, spent several months' full-time effort of all developers to update the Windows code base to make it more stable
 - * As a result, today's Windows versions are reasonably stable and secure

Windows – Support

- * Windows is closed source software
 - * Only Microsoft has access to the source code and knows how Windows works internally
 - * Users are dependent on Microsoft for support and updates
- * Users must follow updates and software upgrades to get support
 - * Extended support is sometimes possible, but at a price
 - * This leads to frequent (and usually costly) upgrade projects

End user operating systems

- * Some operating systems are exclusively designed to be used on end user devices
- * Some examples:
 - * **Windows** XP, Vista, Windows 7, Windows 8, and Windows 10 - Microsoft's PC operating system
 - * **Mac OS** - Apple's operating system for laptops and desktops, based on BSD
 - * **Ubuntu** - Linux distribution specially designed for laptops and desktops
 - * **iOS** - Apple's operating system for mobile devices (iPhone, iPad, etc.), based on BSD
 - * **Android** - Google's operating system for mobile devices, based on Linux

Special purpose operating systems

- * Some operating systems are created for special purposes, like:
 - * Firewalls
 - * Intrusion detection and prevention systems
 - * Routers
 - * Phones
 - * ATM machines
 - * Media centers
- * Typically based on existing operating systems
 - * Usually based on Linux or Windows
 - * Stripped of all unneeded features

Special purpose operating systems

- * A special type of operating system is a real-time operating system (RTOS)
 - * Guarantee to perform tasks in a predefined amount of time
 - * Used where handling events within a predefined time is critical
 - * Factories
 - * Power plants
 - * Vehicles
 - * Example: QNX



Operating System Availability

Failover clustering

- * A failover cluster is:
 - * A group of independent servers running identical operating systems (known as “nodes”)
 - * Connected via a network
 - * Controlled by cluster software running on the nodes
- * Every active application has a standby counterpart available on a passive node
 - * It sits idle until a failover is needed
 - * After a failover, this standby application becomes active and provides service to clients
- * A failover cluster provides high availability to applications
 - * It manages each running application within a node as a package of application components, called a resource pool or an application package

Failover clustering

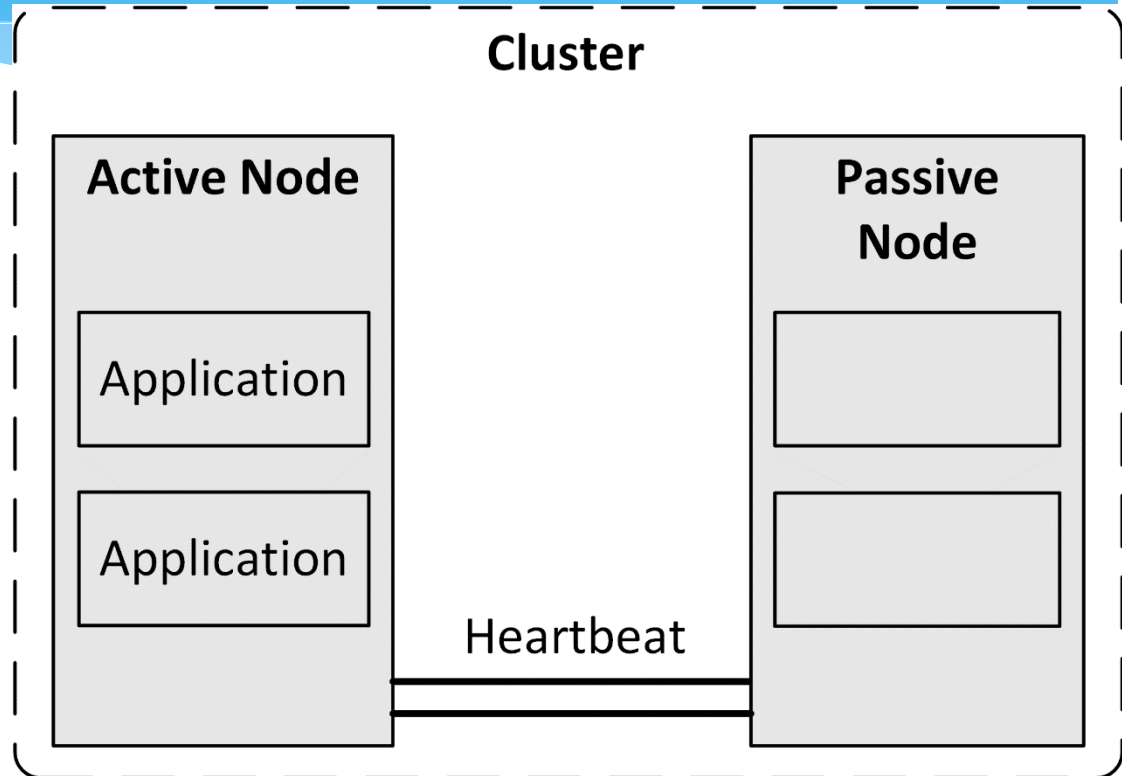
- * Examples of cluster software products are:
 - * Parallel Sysplex
 - * For IBM mainframes
 - * HACMP
 - * For IBM AIX UNIX
 - * MC/Service Guard
 - * For HP-UX UNIX
 - * Windows Cluster Service
 - * For Microsoft Windows
 - * Heartbeat and Pacemaker
 - * For Linux

Failover clustering

- * A resource pool is the single unit of failover within a cluster. It typically contains:
 - * **Application name** and identifier
 - * **Start script** for the application
 - * **Stop script** for the application
 - * **Monitor script** for the application
 - * Continuously checks the status of the application
 - * If the application does not work as expected, a restart or failover is initiated
 - * **Virtual IP address** the application can be addressed with
 - * **Mount points** for storage – the disks that must be available to the application

Failover clustering

- * A cluster network typically consists of redundant physical Ethernet connections
- * Carries heartbeats between all nodes in the cluster
- * A heartbeat allows nodes to detect the unavailability of nodes by regularly sending packets to each other's network interfaces



Failover clustering

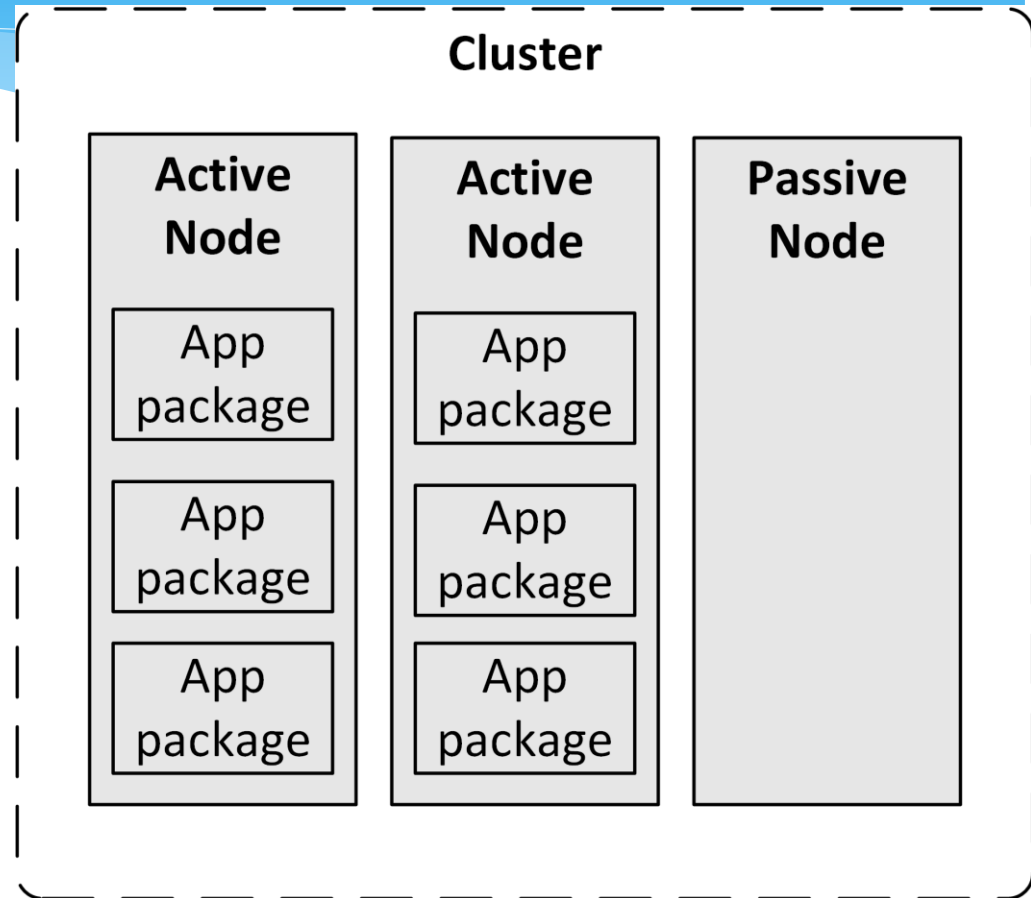
- * All nodes are able to access data on shared storage
 - * Every individual disk is mounted to one active application only at any given time
 - * This usage of shared storage is also called ‘shared nothing clustering’
- * Distributed Lock Management (DLM) clustering:
 - * Each cluster node can access the same resource, for instance a disk, *at the same time*
 - * A lock mechanism is responsible to manage data to avoid corruption

Failover clustering

- * In case of for instance a server crash or a power outage, all applications running on that server node will not be brought down cleanly
 - * When the applications are restarted on another node in the cluster, standard crash recovery should take place
 - * The file system must take care of performing file system checks before mounting
 - * The application must perform its standard recovery on startup
- * Application recovery in case of a failover is identical to an application startup following a server power failure

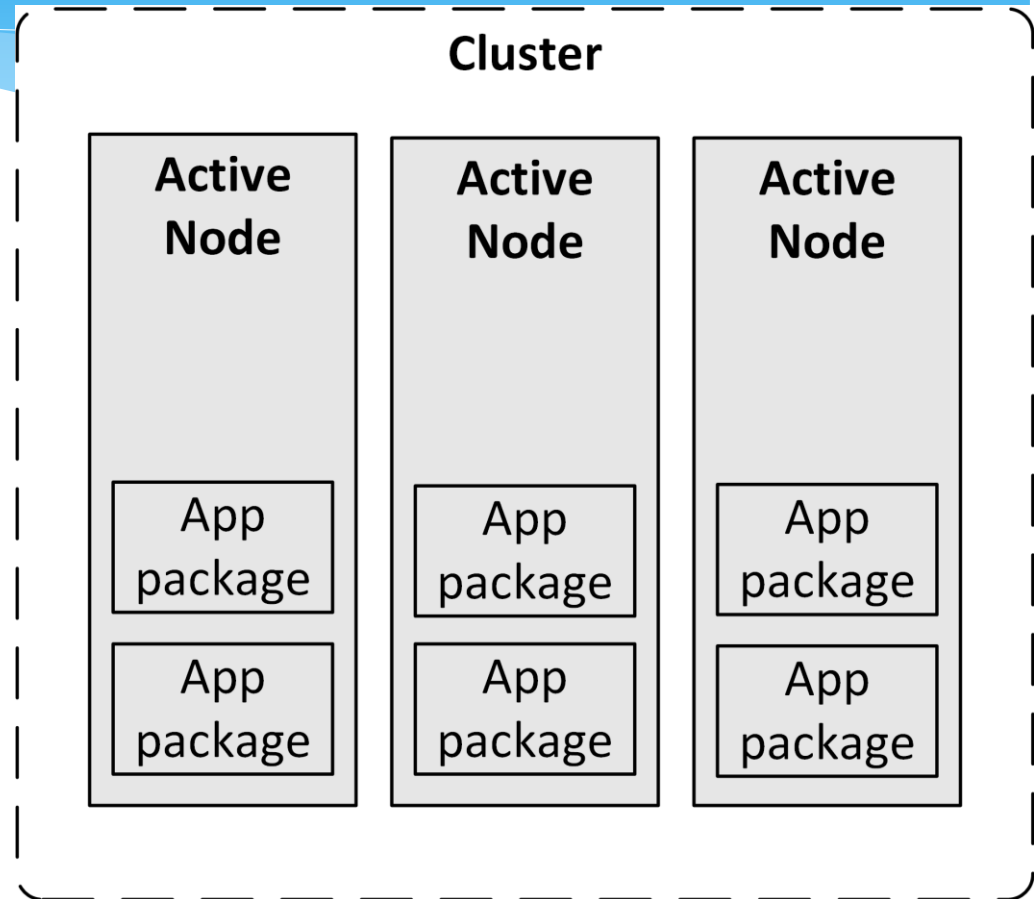
Failover clustering

- * A spare node could be added to a cluster to handle failovers
- * This is called a N+1 cluster
 - * N represents the number of nodes with active applications
- * N+2 or N+3 can provide more redundancy



Failover clustering

- * An alternative is an N to N cluster
- * There is no spare idle node
- * Each node has some spare capacity



Failover clustering

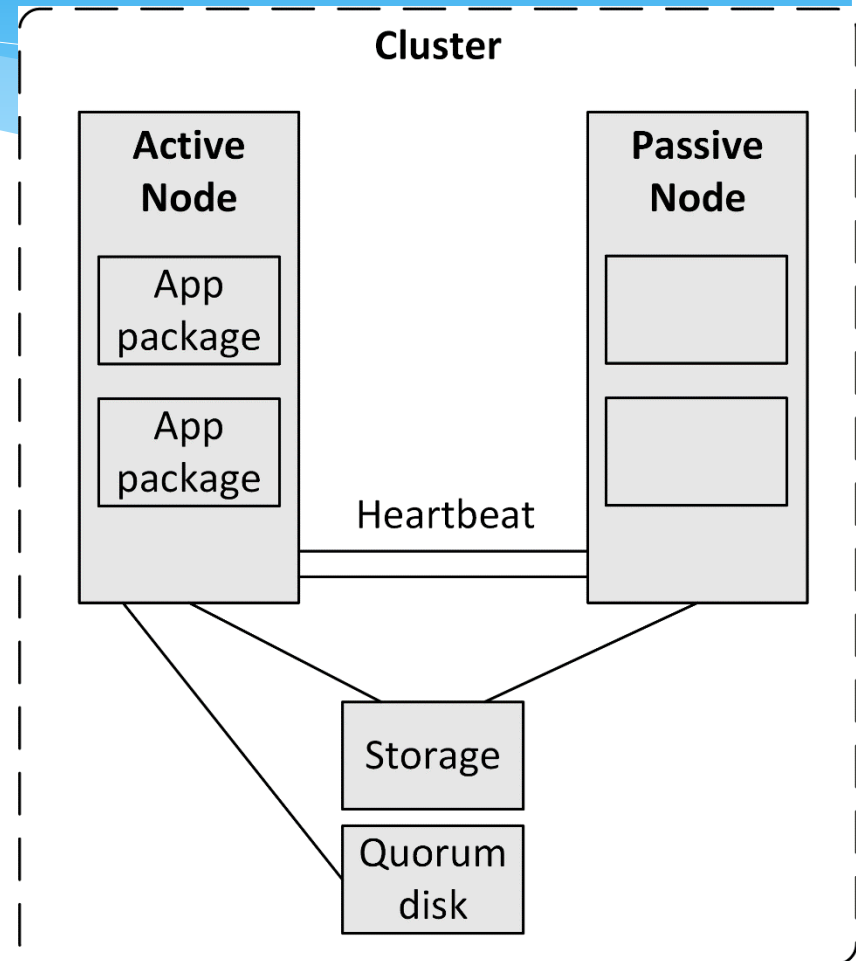
- * The advantage of an N+N cluster is that the available hardware is always used
- * All memory and CPU cycles in the operating system can be used by all running applications
 - * When a failover occurs, less memory and CPU cycles are available to the applications, possibly leading to some performance degradation

Voting and quorum disks

- * Most clusters contain two nodes
- * In a cluster with an even number of nodes, if the nodes are disconnected from each other, the status of the other nodes is unknown to each node
- * One of two situations occurs:
 - * Each node decides that it has lost contact with the active cluster
 - * Both nodes decide to stop (effectively bringing down the cluster)
 - * Each node decides that the other node must be down
 - * Each node decides to be the new active node in the cluster
 - * Known as a split-brain situation

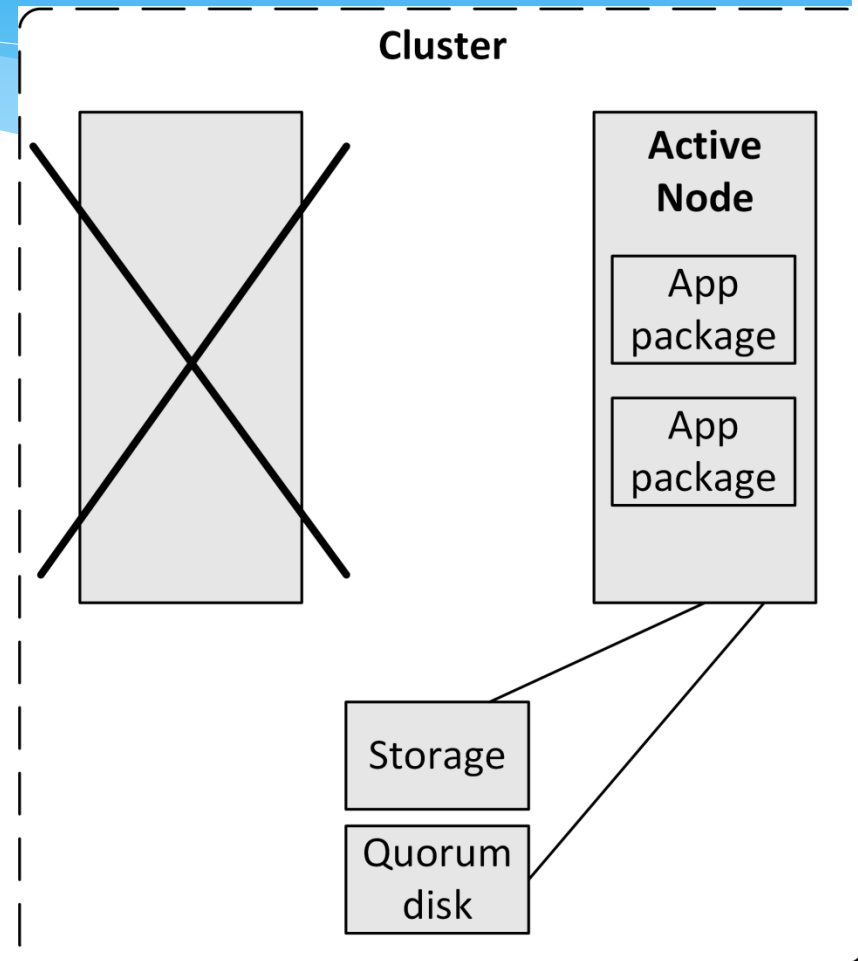
Voting and quorum disks

- * A voting mechanism determines which part of the cluster is faulty and which part of the cluster is working properly
- * In a two-node cluster there is no majority possible in a voting system
- * Therefore a virtual third node is used
 - * Usually in the form of a shared disk called a quorum disk
 - * Installed at a third location



Voting and quorum disks

- * The quorum acts as one vote in the voting system
- * The quorum disk is always assigned to one (and only one) node at any time
- * A faulty node releases its quorum assignment automatically
- * The working node gets two votes: one from itself, the other from the quorum disk
- * The faulty node will stop working, because it has only one vote



Cluster-aware applications

- * Cluster-aware applications run active instances on multiple nodes
- * Examples:
 - * Oracle RAC (Real Application Cluster)
 - * Microsoft SQL Server Always On Failover Cluster
 - * Microsoft Exchange Server
- * Enhances switch-over times in case of a failure
 - * In case of a failure, the application does not need to be started on another node before it can service clients
- * Cluster-aware applications provide scalability in addition to high availability
 - * Client requests can be distributed among multiple cluster nodes
 - * Handle increased demand and traffic by adding additional nodes to the cluster



Operating system performance

Operating system performance

- * The performance of an operating system is dependent on:
 - * The performance of the underlying hardware
 - * The type of load generated by the applications
 - * The configuration of the operating system itself
- * Some operating system performance can be gained by:
 - * Increasing memory
 - * Decreasing the kernel size

Increasing memory

- * An operating system should have enough memory to run all applications needed at any time
- * When an application needs more than the available memory, memory is freed:
 - * Moving less used memory pages to disk
 - * Paging
 - * Some paging is not bad

Increasing memory

- * When memory is really low, moving an entire application's allocated memory to disk
 - * Swapping
 - * Ruins the performance of an operating system
 - * Data stored on disk is at least three orders of magnitude slower than data stored in RAM memory
 - * Swapping must be avoided at all times
 - * Increase memory
 - * Run less (demanding) applications

Increasing memory

- * Increasing memory benefits the operating systems' performance
- * All memory not used by applications is used to cache disk blocks
 - * This is the main reason why the performance of operating systems usually increases when memory is added
- * Operating systems use highly sophisticated algorithms to optimize disk caching
- * In general, tweaking the memory management system of an operating system provides little benefits

Decreasing kernel size

- * Some operating systems (like UNIX and Linux) allow tuning kernel parameters of the operating system
 - * Unused features (like support for IPv6 or floppy disk drives) can be switched off, leading to a smaller kernel size
- * To create a smaller kernel, the kernel must be recompiled or re-linked
 - * This is a highly automated, low risk operation on most UNIX and Linux systems
 - * A restart of the operating system is needed after a kernel rebuild
- * Not all operating systems allow rebuilding the kernel
 - * For instance, the Windows kernel cannot be rebuilt

Decreasing kernel size

- * A smaller kernel has the following benefits:
 - * It simplifies the kernel:
 - * Lower risk of crashes
 - * Smaller security attack surface
 - * The kernel must be in memory at all times
 - * It cannot be paged or swapped-out
 - * A smaller kernel will free up memory for applications and disk caching
 - * Switched-off features don't need patching to keep them up-to-date
 - * The operating system starts faster when the kernel is small



Operating System Security

Patching

- * All operating system vendors provide small software updates called patches:
 - * Fixing bugs or design flaws
 - * Closing security holes
 - * Small improvements

Patching

- * In general, patches come in three categories:
 - * Regular patches
 - * Meant to fix low priority software bugs
 - * Some regular patches fix multiple bugs at once
 - * Hot-fixes
 - * Repairs a bug or flaw in the operating system that needs to be fixed fast
 - * Used to close a security hole or to fix an error introduced by another patch or service pack
 - * Hot-fixes should be installed as soon as possible
 - * Service packs
 - * Also known as support packs or patch packs
 - * A collection of patches and hot-fixes that are packed together and can be installed in one deployment
 - * Sometimes service packs also introduce new functionality to the operating system

Patching

- * It is good practice to install all patches, hot-fixes, and service packs as soon as possible
- * Test them before deploying in production
 - * They could introduce unwanted effects in the infrastructure
- * Patches hot-fixes, and service packs are usually provided with release notes
 - * They describe what changes are made to the operating system
 - * Read release notes before installing the patch!
 - * When a patch or hot fix does not have impact on a specific deployment it can be discarded

Hardening

- * Hardening is a step by step process of configuring an operating system to protect it against security threats
- * The operating system is stripped down to support only essential services and processes
 - * Unnecessary protocols and subsystems are switched off
 - * Unused user accounts are removed or disabled
 - * All new and relevant hot-fixes, patches, and service packs are applied
- * Harden all operating systems in the infrastructure using a hardened operating system configuration template
 - * This template is used to instantiate new operating systems
 - * Ensure security is optimal and is consistent in all deployments

Virus scanning

- * Windows, Linux and end user operating systems are vulnerable to viruses
 - * It is good practice to install a virus scanner
- * Virus scanners can have an impact on the performance of the operating system
 - * The virus scanner must be configured to only scan high risk files and directories based on a risk analysis
 - * For instance, it makes no sense to protect a database table file with a virus scanner

Host-based firewalls

- * Most operating systems, including Windows, Linux, and UNIX, provide a built-in host-based firewall
- * A host-based firewall is a software firewall
 - * Part of an operating system
 - * Protecting an individual host from unwanted network traffic
- * Host-based firewalls typically block all incoming network traffic

Host-based firewalls

- * Rule sets define which type of traffic is allowed to communicate with the operating system, based on:
 - * Source and destination IP address
 - * TCP or UDP port
 - * The running process sending and/or receiving the network traffic
- * It is good practice to enable host-based firewalls on all machines
 - * Servers
 - * End user devices

Limiting user accounts

- * Operating systems have local user accounts that can login to the operating system
- * Most operating systems also have a special super user account called "root", "supervisor", "admin", or "administrator"
 - * These accounts have almost unlimited power
 - * They should be used only to provide permissions to user accounts bound to a physical person
 - * Under normal circumstances, these accounts should never be used
 - * It should be possible to do all work using a user-bound account with sufficient rights

Hashed passwords

- * Operating systems should only store hashed passwords
 - * When a user logs in, her password is hashed
 - * The hashed password is compared to the stored hash
 - * If the two are equal the login succeeds
- * There is no way to calculate or extract the original password from the hashed one
 - * The hashed passwords should never be disclosed
 - * When weak passwords are used, brute force or dictionary attacks can be used to find the passwords

IT Infrastructure Architecture

Hosting and deployment options
(chapter- 14)

Hosting options

- Hosting options define *where* infrastructure is deployed
- Deployment models define *how* infrastructure is deployed
- Infrastructure can be hosted:
 - On-premises
 - In a colocation
 - Outsourced

On-premises hosting

- Infrastructure components run on the premises of the organization using the infrastructure
 - In the datacenter of an existing building
 - In a dedicated, specially designed datacenter building
- The building must have:
 - Enough space
 - An uninterruptable power supply (UPS)
 - Options to install sufficient cooling
 - Fire prevention and detection
 - External redundant network capabilities with enough bandwidth
 - Sufficient floor loading capacity

On-premises hosting

- Drawbacks:
 - On-premises datacenters don't scale well, as they are embedded in existing (office) buildings
 - The organization must have enough knowledge and staff available to manage the datacenter

Colocation

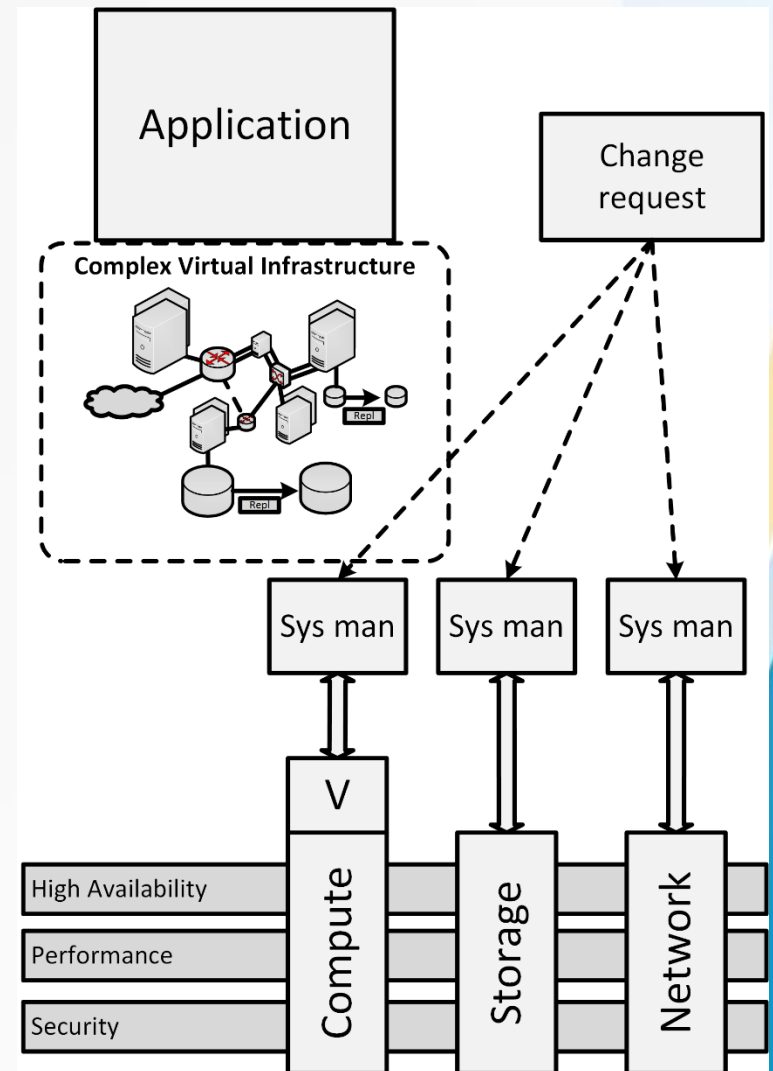
- A third party dedicated datacenter where racks, floor space, and network bandwidth can be rented
- Hosts and connects customer owned infrastructure components
- Provides:
 - Power
 - Cooling
 - Physical security
- Racks are empty – all infrastructure components must be provided and managed by the organization renting the colocation racks

Outsourcing

- Full infrastructure outsourcing is a subcontracting service in which some third-party purchases, deploys, hosts, and manages the infrastructure, and performs its lifecycle management
 - Managed using Service Level Agreements
 - Very rigid change management process
- Frees the organization from investing in hardware
 - Only operational cost
- The outsourcing organization must have:
 - A demand organization
 - A process to manage the outsourcing party

Enterprise infrastructure deployment

- A traditional enterprise infrastructure deployment
- Enterprise grade hardware delivers three infrastructure resources:
 - Compute
 - Storage
 - Network
- Provides:
 - High availability
 - High performance
 - Some security controls



Enterprise infrastructure deployment

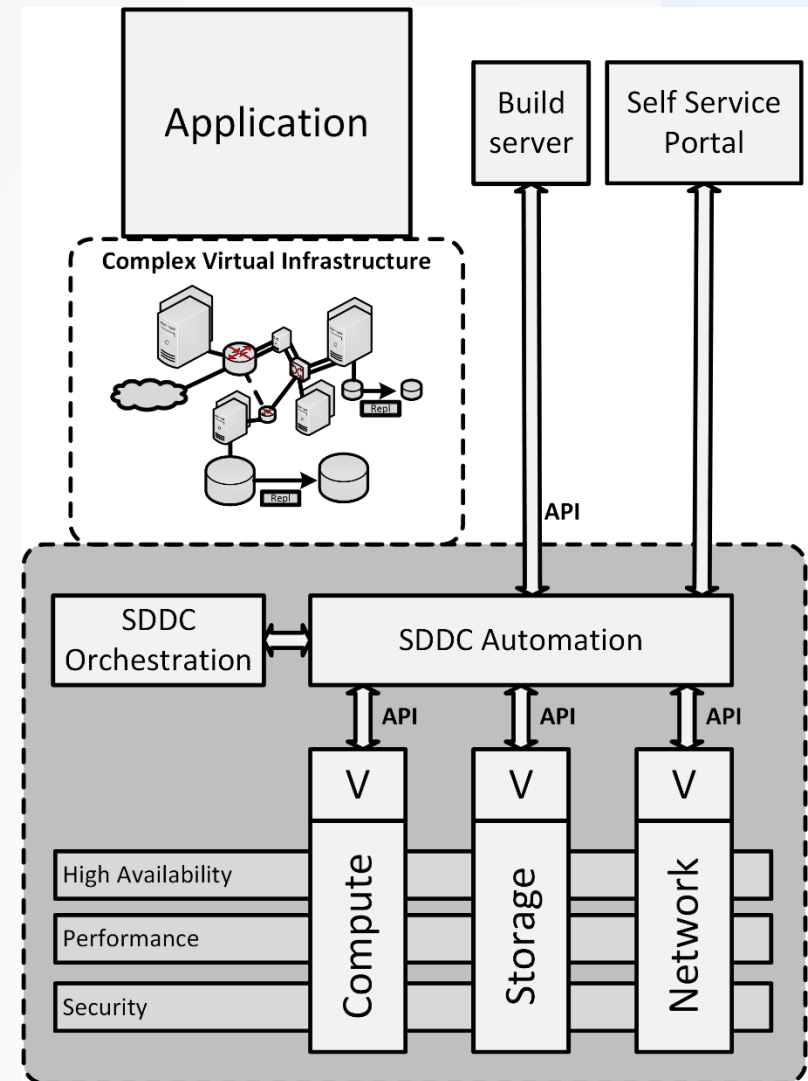
- Most enterprise infrastructure deployments implement compute virtualization to provide virtual machines
- Each resource is managed by a team of dedicated systems managers
- Changes are managed by a workflow based change process
 - Each systems manager is responsible to manually perform his part of the change

Enterprise infrastructure deployment

- Enterprise infrastructures can deliver complex virtual infrastructures to applications
- The virtual infrastructure is designed to meet the requirements of applications
- Applications depend on the high availability, performance and security controls delivered by the infrastructure

Software-defined datacenter - SDDC

- A software-defined datacenter (SDDC) is an architecture in which all infrastructure resources – compute, storage and networking – are virtualized, and can be configured using software APIs
- An SDDC is an extension of an enterprise infrastructure
 - All resources are virtualized and managed by SDDC automation and orchestration software



Software-defined datacenter - SDDC

- An SDDC is characterized by:
 - Automation
 - Orchestration
 - Abstraction of resources into software and code
- Changes are managed by an automated workflow
- An orchestrated change can lead to a number of automated changes in various resources

Software-defined datacenter - SDDC

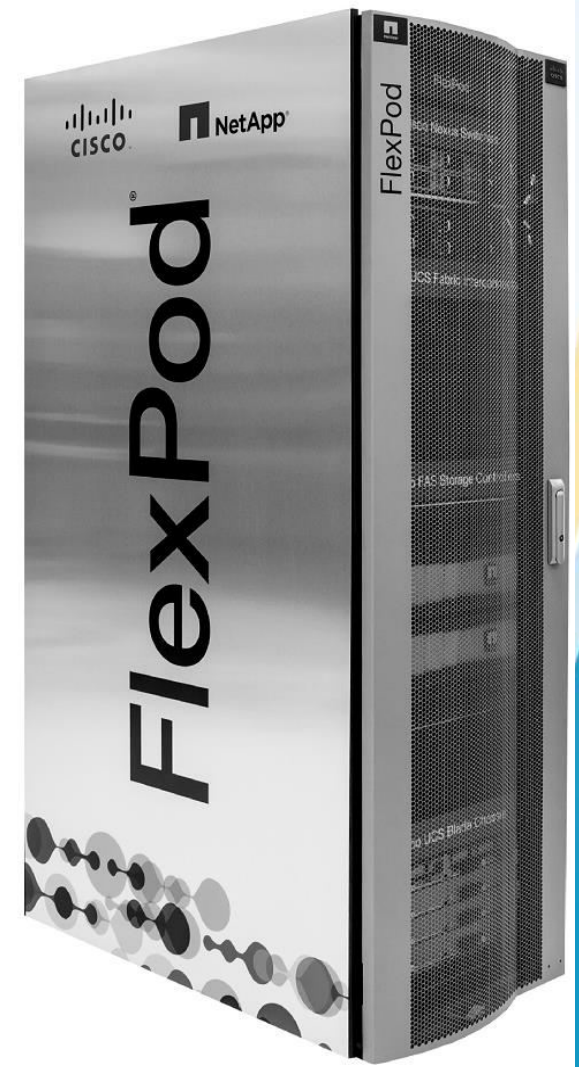
- An SDDC is the foundation for cloud computing
- Developers, DevOps teams and systems managers can create and deploy new infrastructures using:
 - A manual self-service portal
 - A combination of a build server and APIs
- Allows the user to request:
 - Desired infrastructure components
 - Their sizing to meet performance demands
 - Their required availability

Software-defined datacenter - SDDC

- SDDC provides tools for:
 - Costing
 - Logging
 - Reporting
 - Scaling (up and down)
 - Decommissioning of the infrastructure resources
- Examples
 - IBM Cloud Orchestrator
 - OpenStack Horizon
 - VMware vRealize

Converged Infrastructure

- Compute, storage, and network components are designed, assembled, and delivered by one vendor and managed as one system, typically deployed in one or more racks



Hyperconverged Infrastructure

- Compute, storage, and network components are implemented within a single server node
- A hyperconverged infrastructure comprises:
 - A large number of identical physical servers from one vendor
 - Direct attached storage in the server
 - Special software that manages all servers, storage, and networks as one cluster running virtual machines



Hyperconverged Infrastructure

- The technology is easy to expand on-demand, by adding nodes to the hyperconverged cluster
- Hyperconverged systems are an ideal candidate for deploying VDI environments
 - Storage is close to compute (in the same node)
 - Scales well with the rise in the number of users (add more nodes)

(Hyper)converged Infrastructure

- Advantages:
 - Only one vendor
 - All updates for compute, storage and networking in one service pack
 - Easy scalable
- Disadvantages:
 - Vendor lock-in – the solution is only beneficial if all infrastructure is from the same vendor
 - Scaling can only be done in fixed building blocks
 - If more storage is needed, compute must also be purchased

Cloud computing

- Cloud computing is one of the most important paradigm shifts in computing in recent years
- Definition NIST:

*Cloud computing is a model for enabling ubiquitous, convenient, **on-demand network access** to a **shared pool** of configurable computing resources that can be **rapidly provisioned** and released with **minimal management effort** or service provider interaction*

- Cloud computing is an outsourcing model, in which IT services are provided and paid based on actual on-demand use

Cloud characteristics

- **On demand self-service** – End users can configure, deploy, start and stop systems or applications on demand
- **Rapid elasticity** – A cloud is able to quickly scale-up and scale-down resources, leading to elasticity of resources
- **Resource pooling** – Instead of providing each application with a fixed amount of processing power and storage, cloud computing provides applications with resources from a shared pool
- **Measured service** – The actual resource usage is measured and billed. There are no capital expenses, only operational expenses
- **Broad network access** – Capabilities are available over the network and accessed through standard mechanisms

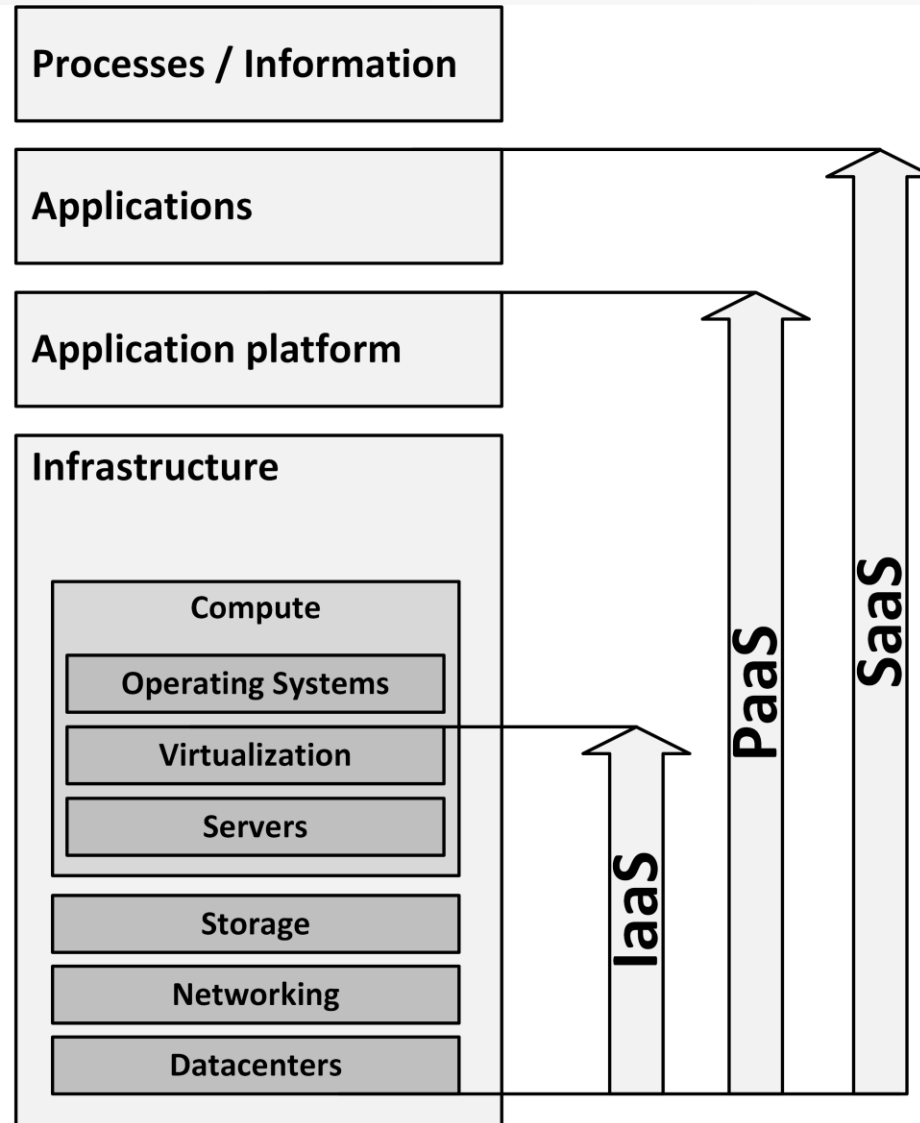
Cloud deployment models

- A **public cloud** is accessible through the internet, and available to the general public
- A **private cloud** is operated solely for a single organization, whether managed internally or by a third-party, and hosted either on premises or external
- A **community cloud** is much like a private cloud, but shared with a community of organizations that have shared concerns
- In a **hybrid cloud** deployment, a service or application is provided by a combination of a public cloud, and a community cloud and/or a private cloud

Cloud service models

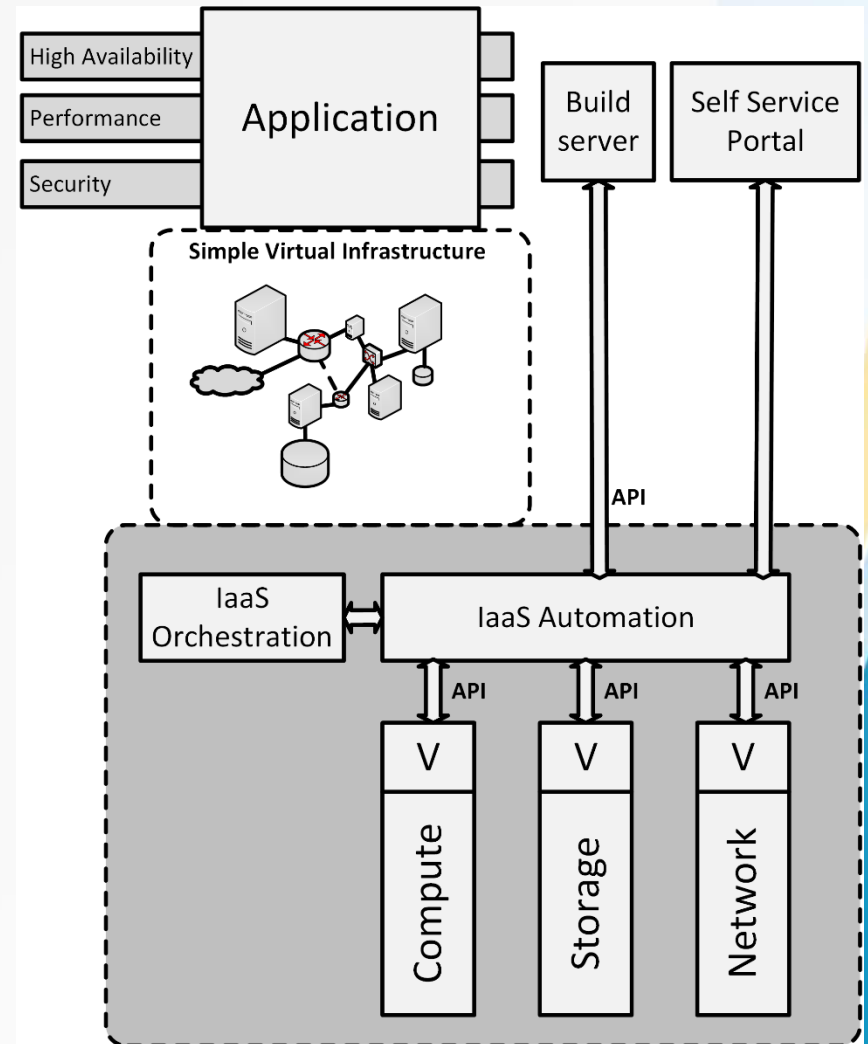
- **Software-as-a-Service (SaaS)** delivers full applications that need little or no configuration
 - Microsoft Office365, LinkedIn, Facebook, Twitter, and Salesforce.com
- **Platform-as-a-Service (PaaS)** delivers a scalable, high available, open programming platform that can be used by developers to build bespoke applications that run on the PaaS platform
 - Microsoft Azure Cloud Service and Google App Engine
- **Infrastructure-as-a-Service (IaaS)** delivers virtual machines, networking, and storage
 - Amazon Elastic Cloud (EC2 and S3) and Microsoft Azure IaaS

Cloud service models



Infrastructure as a Service (IaaS)

- Provides:
 - Virtual machines
 - Virtualized storage
 - Virtualized networking
 - Systems management tools to manage them



Infrastructure as a Service (IaaS)

- Failure of the infrastructure is acceptable:
 - Based on cheap commodity white label hardware
 - The philosophy is to keep the cost down by allowing the hardware to fail every now and then
 - Failed components are either replaced or simply removed from the pool of available resources

Infrastructure as a Service (IaaS)

- IaaS provides simple, highly standardized building blocks to applications
 - Does not provide high availability, guaranteed performance or extensive security controls
 - Applications should be:
 - Robust to allow for failing hardware
 - Horizontally scalable to increase performance

Infrastructure as a Service (IaaS)

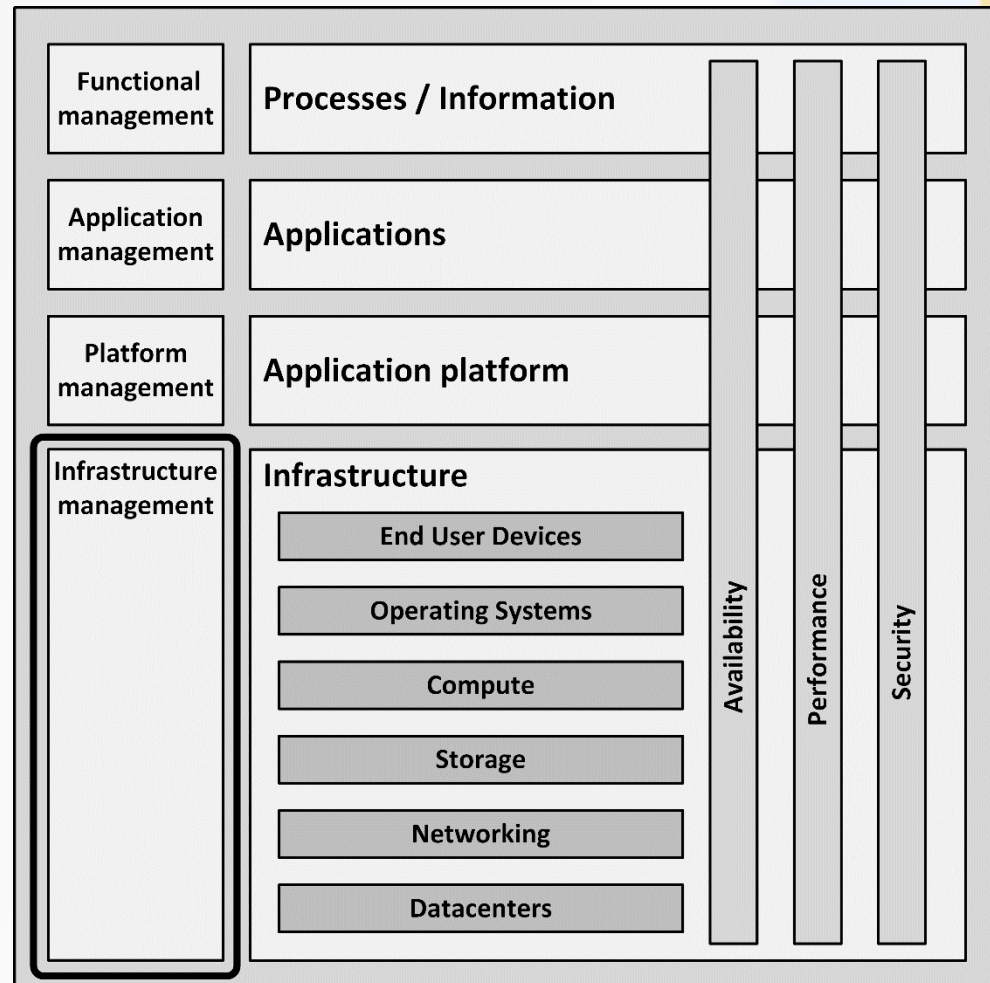
- To use IaaS, users must create and start a new server, and then install an operating system and their applications
- The cloud provider only provides basic services, like billing and monitoring
- The user is responsible for patching and maintaining the operating systems and application software
- Not all operating systems and applications can be used in an IaaS cloud
 - Many software licenses prohibit the use of a fully scalable, virtual environment like IaaS

IT Infrastructure Architecture

Infrastructure lifecycle
(chapter 15 - 16)

Introduction

- The infrastructure lifecycle encompasses the following steps:
 - Planning and Design
 - Purchasing infrastructure and services
 - Build-up, testing and go-live
 - Maintaining the infrastructure
 - Deploying applications
 - Decommissioning the infrastructure



Planning and Design

- The Planning stage involves:
 - Feasibility study
 - Site Selection
 - Conceptual Design
 - Develop Comprehensive Design Plan

Purchasing infrastructure and services

- Most large-scale IT projects require procurement of hardware, software, or services
- The purchase process entails:
 - Determining what is needed
 - Getting an offer
 - Ordering
 - Delivery
 - Warranty
 - Renewal

Determine what is needed

- Before any purchase can be made, it must be crystal clear what is actually needed
- A Bill of Materials (BoM) is made that includes part numbers of all items
 - If allowed by the purchasing rules, work with suppliers to get the BoM
- Also a Statement of Work (SoW) is made
 - A SoW describes what the supplier will do, apart from delivering the goods
 - It must be clear from the start who does what
- The supplier can have specific requirements, like:
 - Is a loading dock available to deliver goods to the datacentre?
 - Is the elevator large enough to lift the equipment to the final destination?

Getting an offer

- In a large organization, the lead time for the internal procurement process can be several weeks, or even longer
 - Find a supplier
 - Handle contract issues and/or to get signatures from management to formally place the order
 - Procurement will try to get discounts
- It typically takes four to eight weeks for the supplier to deliver the goods

Choice of suppliers

- Most organizations use preferred suppliers for standard purchases
- Having a small number of preferred suppliers makes the purchase process easier
 - Contracts are already in place
 - Discounts can be negotiated because of large volume purchases
- Organizations often choose for a predefined purchase policy for their software stack
 - When possible all software is either purchased from preferred suppliers, or built based on technology from preferred suppliers

Choice of suppliers

- The alternative is using a best of breed policy
 - Each component is chosen based on the best quality or the most comprehensive feature set
- Combination also possible:
 - A standard software stack for commodity components
 - Best of breed products for highly specific tasks, that are close to the core business processes

Choice of suppliers

- With hardware, predefined choices are typically made as well
 - For instance, a policy could be to buy all network equipment from Cisco and all servers from HP
 - Easier management of support contracts
 - When a hardware component fails, one telephone number to the supplier's support desk is all it takes to start the repair process
 - Limit the knowledge needed in the organization to manage components

Choice of suppliers

- Having preferred suppliers can lead to a vendor lock-in
 - After some time, for practical reasons, it becomes unfeasible to change suppliers
 - The resulting lack of competition can lead to
 - A higher price level in the long run
 - A decrease in service levels

Bidding and tendering

- Getting an offer may involve a formal bidding process, also known as tendering
- In general, a bidding process comprises the following steps:
 - RFI – Request for Information
 - A large group of suppliers is asked to inform the purchase department if they are capable of providing the required goods or service

Bidding and tendering

– Short list

- Based on the RFI responses, the purchase department creates a short list of suppliers
- Which are most likely to be able to deliver the goods for a good price and with good service?

– RFP – Request for Proposal

- Suppliers in the short list are requested to make a proposal for the delivery

– Questions and clarification

- Planned between the publication of the RFP and the supplier's responses
- Suppliers are given the opportunity to ask questions about the RFP (in writing)

Bidding and tendering

- Offer
 - Suppliers provide the answers to the RFP
 - Includes an initial offer
- Terms and conditions negotiations
 - The purchase department starts negotiations with the suppliers that provided the best response to the RFP
- BAFO - Best and final offer
 - Preferred suppliers make a final price and SoW
- Award
 - Based on the BAFO, the purchase department awards the supplier with the deal

Ordering and delivery

- Ordering
 - Typically done by the purchasing department
 - Because the delivery time is often weeks after the purchase order is placed, it makes sense to start ordering the goods as early as possible
- Delivery
 - Beware that the person that physically receives the goods, is not always the one formally accepting the delivery
 - Before signing for delivery, check the boxes for any damage and check for completeness of the delivery
- Warranty period
 - During the warranty period, defects will be fixed without additional cost

Renewal

- When purchased goods are used for some time, they might need renewal
 - Hardware is often used for five years before it is replaced
 - Software typically has major releases every few years
 - Service contracts are also often agreed upon for a fixed number of years
- Often a renewal of the hardware, software licenses, or service contracts leads to a new purchase process
- Systems management should have a Life Cycle Management (LCM) process implemented to handle timely renewal

Deploying the infrastructure



Assembling the infrastructure

- Infrastructure can be assembled using the following checklist:
 - Build up the physical datacenter room
 - Raised floors
 - Uninterruptable power supply
 - Cooling facilities
 - Fire prevention and detection
 - Physical security
 - Install redundant power cabling

Assembling the infrastructure

- Install racks

- Typically, separate racks are installed for network, storage, and compute components
- Check if the computer floor can still be opened (no racks placed on multiple tiles)

- Test the facilities:

- Test alarms that should respond to a power failure or heating, ventilating and air conditioning (HVAC) failure
- Perform a visual check of the facilities
- Ensure cables are properly fixed in the racks and labelled
- Measure the airflow, temperature, and humidity
- Test the physical security controls

Assembling the infrastructure

- Install the server, networking, and storage hardware
 - Ensure not too much equipment is placed in one rack to prevent:
 - Racks from falling over
 - Using too much power
 - Obstructing cooling air flow
 - Ensure:
 - Cabling is properly installed and labelled
 - Equipment can be sled out of the racks without damaging cabling
 - Check if maintenance on a component can be done without interfering with the other components

Assembling the infrastructure

- Allow for a burn-in period to ensure the equipment is not “dead on arrival” (DOA) or fails within the first day
 - Switch the equipment off and on several times to see if it doesn’t break
- Check the power and cooling usage of the equipment

Assembling the infrastructure

- Configure the infrastructure components
 - Configure routers, switches, and storage LUNs
 - Install virtualization and operating systems
 - Configure DNS and NTP
 - Configure security configurations like network zoning and firewalls
 - Perform a basic test to check network connectivity and storage availability

Assembling the infrastructure

- Install systems management tools
 - Backup and recovery
 - Monitoring
 - Logging
 - IDS/IPS
- Test systems management processes.
 - Create incidents, create and handle changes, etc.
- Provide as-built documentation to the systems managers.
 - Populate the Configuration Management Database (CMDB)

Testing the infrastructure

- After assembling the infrastructure, it should be tested
- Each test type has a predefined scope:
 - Functional tests
 - Ensure the infrastructure delivers the required functionality
 - Performance tests
 - Load, stress, and endurance tests
 - Prove the infrastructure has enough resources to run applications with the required performance
 - Security tests
 - Penetration tests and vulnerability scans
 - Prove security controls are in place and are functioning as designed

Testing the infrastructure

- Availability of the infrastructure can be tested by:
 - Physical actions
 - Pulling cables from infrastructure components
 - Unexpectedly rebooting machines
 - A failover test, a fallback test and a disaster recovery test
 - Testing backup and recovery processes and disaster recovery plans

Test stages

- System integration test
 - The combination of components is tested including their interfaces
 - A system integration test checks both functional and non-functional requirements
- Fallback test
 - Fallback from the main datacenter to the secondary datacenter is checked on a technical level
- Migration test
 - Ensures applications are installed without errors and data from previous systems can be migrated to the new system as designed

Test stages

- Acceptance test
 - The final check of the delivered infrastructure
 - Consists mainly of verifying that all tests are performed and that defects found in previous tests are either solved or accepted
 - Leads to a discharge of the project for the delivery of the infrastructure

Go live scenarios

- Big Bang
 - At a set time, the existing system is switched off and the new system is immediately put in production, possibly after a short data migration run.
- Disadvantage:
 - It may be impossible to roll back to the old system after the system is live for some time
 - Downtime can occur when something goes wrong during the switchover

Go live scenarios

- Parallel changeover
 - Both the new and the existing system run simultaneously for some time (typically weeks)
 - Allows for testing the new system on both functionality and non-functional attributes
 - Ensuring it works with live production data before switching off the existing system
 - Switching back is possible at any time, minimizing risk
- Disadvantage:
 - The cost of maintaining both systems
 - Possible extra work to keep both systems in sync
 - Many system designs don't allow running two systems in parallel, for instance, if the system has many data interfaces with other systems

Go live scenarios

- Phased changeover
 - Individual components or functionalities of the existing system are taken over by the new system, one by one
 - The changeover can be done gradually and controlled
- Disadvantage:
 - Can be costly, as many interfaces between the existing and the new system must be created and maintained
 - New interfaces introduce new risk to the scenario
 - The existing system must be kept online until the last component or functionality is moved to the new system

Go live scenarios

- The go-live should be very well prepared
- A step-by-step plan must be created describing each step in detail
- This plan must be reviewed, tested and improved multiple times
 - Well in advance of the go-live date to eliminate possible surprises and to minimize risk
- The scenario should include:
 - Intermediate tests
 - Multiple “go/no go” milestones
 - A defined point of no return

Go live scenarios

- At the go-live date, high alert is needed from:
 - The project team
 - Systems managers
 - Service desk
 - Senior management
- After the new system is live, on-site support should be available for some predefined time

IT Infrastructure Architecture

Infrastructure lifecycle (Part 2) (chapter 17 - 19)

Infrastructure lifecycle Steps

Previous lecture

- Planning and Design
- Purchasing infrastructure and services
- Build-up, testing and go-live

Today's Lecture

- Maintaining the infrastructure
- Deploying applications
- Decommissioning the infrastructure

Maintaining the infrastructure

Systems management processes

- Systems management is important
 - A typical infrastructure project takes a couple of months to complete
 - The infrastructure is often used in operation for many years, sometimes even decades
- Examples of systems management processes:
 - Incident management
 - Change management
 - Configuration management

Systems management processes - COBIT

- COBIT provides a structure for setting up:
 - IT Governance
 - IT organization
 - IT architecture
- Describes thirty-four IT processes
 - Management objectives
 - Associated measures
 - Performance indicators
 - Maturity levels
- Focused on management issues
- COBIT is a framework created by ISACA

Systems management processes - ITIL

- ITIL is the most used approach to implementing systems management processes
- Full life cycle of IT management
 - IT organization
 - 37 processes
 - Most implemented processes:
 - Incident management
 - Change management
 - Problem management
 - Release management
 - Capacity management

DevOps for infrastructure

- DevOps is a contraction of "developer" and "system operator"
- DevOps is typically used by teams developing and running functional software
- DevOps teams consist of:
 - Developers
 - Testers
 - Application systems managers
- Each team is responsible for developing and running one or more business applications or services

DevOps for infrastructure

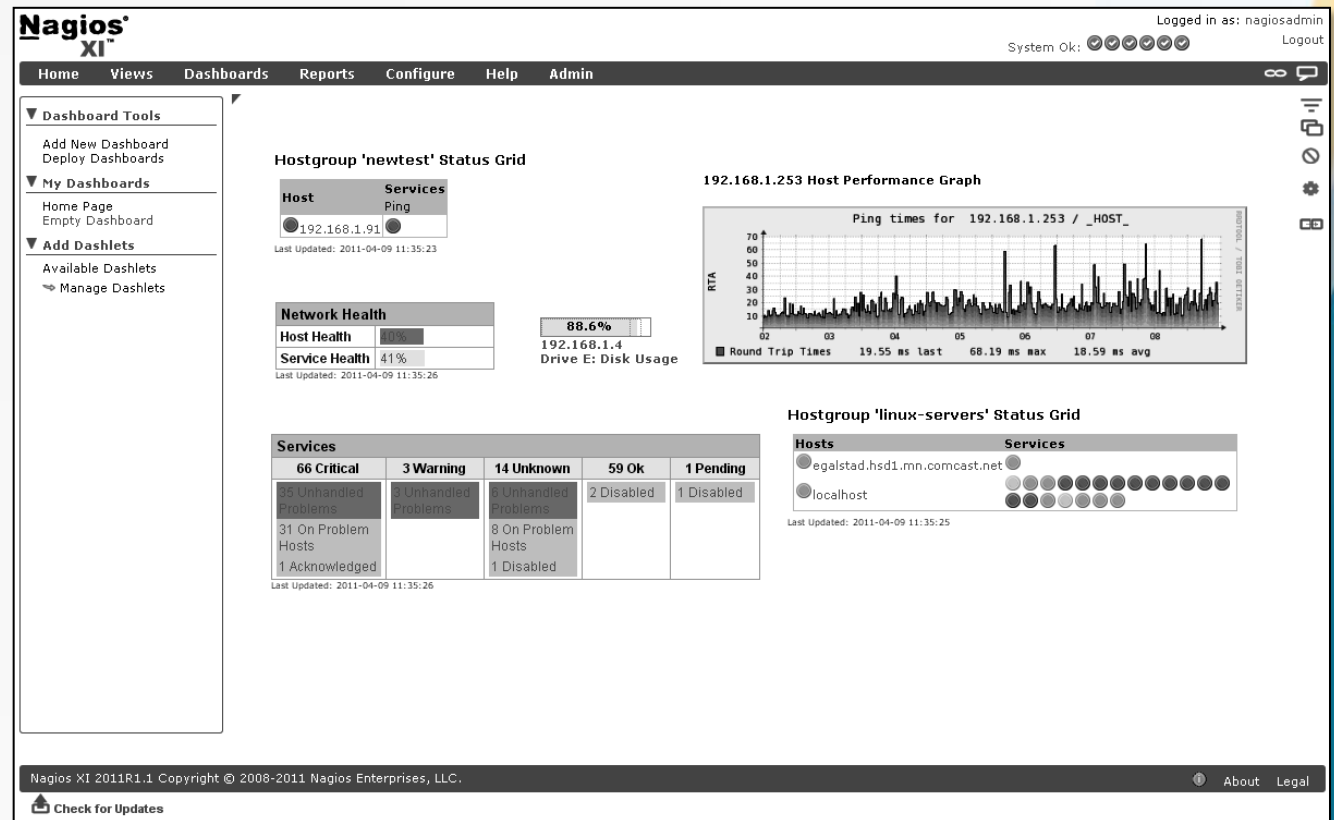
- DevOps can also be used to develop and run an infrastructure platform
- In an infrastructure DevOps team:
 - Infrastructure developers design, test, and build the infrastructure platforms and manage their lifecycle
 - Infrastructure operators keep the platform running smoothly, fix incidents, and apply small changes

Monitoring

- Monitoring continuously inspects IT components for events
- Examples of events:
 - Error conditions
 - Signs of (upcoming) failures
 - A disk with only little free space left
 - Unusually high CPU utilization
 - Extreme network bandwidth usage
- Alarms can be sent if a certain threshold is reached

Monitoring

- Monitoring systems provide real-time dashboards with overviews of an entire infrastructure landscape
- Nagios
- Zabbix
- HP Operations Manager
- BMC Patrol



Management using SNMP

- Simple Network Management Protocol
- SNMP can be used to:
 - Remotely change or update configurations
 - Collect statistics and performance information
- Devices that support SNMP include:
 - Routers
 - Switches
 - Servers
 - Workstations
 - Printers

Management using SNMP

- SNMP uses a management/agent model
 - The agent runs on the monitored device
 - Has local knowledge of the system it resides on
 - Translates information to the SNMP protocol
 - A management server collects information from all agents
 - The Network Management System – NMS
 - Monitors and controls managed devices via the agents
 - Reading of values is done in regular polling intervals (like every 30 seconds)

Management using SNMP

- SNMP traps
 - A trap is an alarm that is sent to the NMS
 - A trap is sent immediately – no polling
- Security in SNMP is implemented using a shared secret
 - Called the community string
 - Provides access to agent functionality
- SNMP version 3 provides strong security features:
 - Encryption
 - Strong authentication
 - Integrity measures

Logging

- Most infrastructure components generate log data
- Examples:
 - Network routers and switches
 - Operating systems
 - Applications
 - Databases
 - Firewalls
 - Intrusion detection systems
- Log data can be used to:
 - Correlate events
 - Identify sources of application issues
 - Identify trends to predict or even prevent unavailability
 - Find security vulnerabilities or security breaches

Capacity management

- Capacity Management ensures the timely availability of sufficient infrastructural capacity to process, transport, and store data now and in the future

Capacity management

- The following input is needed:
 - Monitoring of resources to detect trends
 - Reduced free disk capacity provides insight in when to purchase or free-up disk capacity
 - Business plans to anticipate on business changes that might have impact on the infrastructure
 - A marketing campaign during the summer time could justify temporary adding server capacity
 - Developments in technology
 - Upgrading servers when a higher capacity server blade becomes available

Deploying applications



DTAP environments

- DTAP:
 - Development
 - New software is developed or existing software is modified
 - Test
 - Software is tested by independent testers
 - Acceptance
 - Software is accepted by a delegation of the user population
 - Production
 - When all tests are successful, the software is deployed in the production environment
- DTAP environments are used in the software development process

DTAP environments

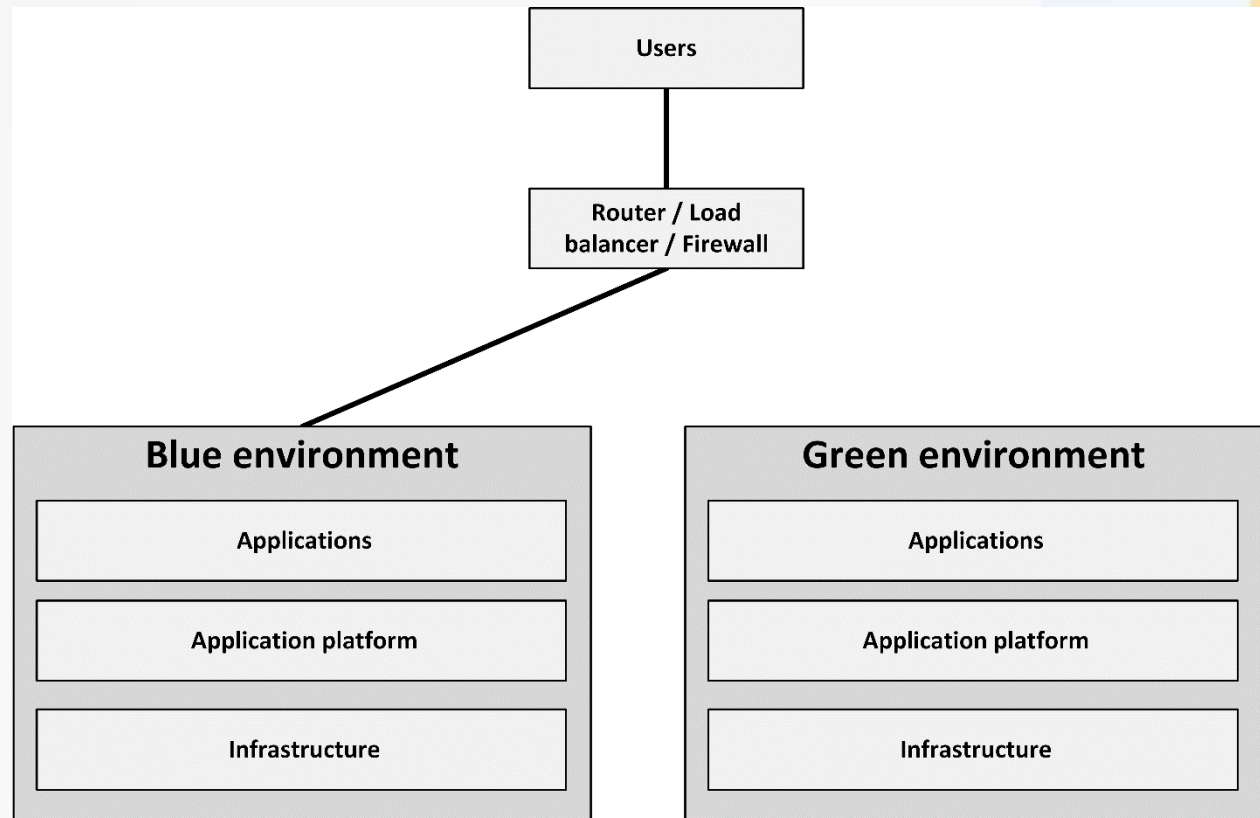
- DTAP suggests four environments, but often more environments are used
- A real-world DTAP environment could contain a:
 - Sandbox environment
 - Pre-development environment, where preliminary tests can be performed on new technology or solutions
 - Development environment
 - To develop new software and configurations
 - Test environment
 - Functionally test new software releases
 - User Acceptance test environment
 - Allow end users to functionally test new releases

DTAP environments

- Non-Functional acceptance environment
 - Setup to be identical to the production environment to enable reliable performance, availability, and security testing
- Hot Fix environment
 - Find fixes for production problems
 - Test fixes before they are deployed to production
- Production environment
 - Runs the actual software for end users
- A Systems Management environment is often used to manage the other environments

Blue-Green deployment

- In a blue-green deployment:
 - The hot fix environment is used in conjunction with the production environment
 - Both environments switch roles when new software is deployed



Blue-Green deployment

- At any time:
 - One environment is running production
 - The other environment is used for:
 - Running the previous production environment (for rollback)
 - Hot-fix testing
 - Final test stage for new software

Blue-Green deployment

- Once new software is working in the non-production environment, the routing is switched so that it becomes the production environment
- The previously running production environment is now idle
 - If anything goes wrong, routing can be switched back
 - After some rollback time it becomes the new hot-fix environment

Continuous Delivery

- Continuous delivery speeds up the time and minimizes the effort to deploy changed and new applications in production
- The primary goal of continuous delivery is to make software deployments:
 - Painless
 - Predictable
 - Low-risk
 - Routine events

Decommissioning



Decommissioning

- At the end of its lifecycle, infrastructure must be decommissioned
- The decommissioning process can be broken down into:
 - Preparation
 - Execution
 - Cleanup

Preparation

- Prepare a plan (interview specialists, plan a date)
- Communicate that the system will go down well in advance
- Check for interdependencies with other systems and remove any dependency
- Determine if and how long backup or archived data must be retained
- Check if the system is really not used anymore
- Ask for vendor assistance if needed
- Inform the floor manager of the datacenter

Execution

- Create a final backup
- Remove the system from the monitoring and alerting system
- Remove the system from the backup schedule
- Close the network communications
- Switch off the system and stand by to redeploy it immediately if any dependency pops up
- Physically remove hardware
- Remove cabling and patching related to the system

Cleanup

- Check if no SLAs and licenses are active and paid for decommissioned systems
- Remove firewall rules that are no longer needed
- Remove unneeded installation software from the software vault
- Update documentation and remove redundant documentation
- Wipe all data and/or destroy data media like disks and tapes that were part of the decommissioned infrastructure
- Remove databases, database schemes, or database tables used by the decommissioned system

Cleanup

- Remove DNS records and administered IP addresses
- Remove all user credentials and system roles from the Identity and Access Management system
- Inform the finance department that the system is decommissioned (because of bookkeeping)
- Remove all decommissioned components from the CMDB database

IT Infrastructure Architecture

Ethics for IT infrastructure

The bottom of the slide features several overlapping, wavy lines in shades of blue, orange, and light blue, creating a modern, abstract design.

Definition - Ethics

“Ethics” is broadly defined as well based standards of right and wrong that prescribe our rights, obligations and benefits to society.

In general, ethics is about what is good or right in our day to day lives, and how we make decisions.



Ethics for IT infrastructure

Ethics for IT infrastructure involves considering the moral and societal implications of the design, implementation, and use of technology that underpins an organization's information technology systems. Here are **some key ethical considerations** for IT infrastructure:

- Privacy and Data Protection
- Transparency and Accountability
- Accessibility
- Security and Cybersecurity
- Sustainability and Environmental Impact
- Fairness and Bias Mitigation
- Intellectual Property
- Responsibility in AI and Automation
- Long-Term Planning and Scalability
- Human-Centered Design
- Ethical Decision-Making
- Compliance and Legal Adherence

Privacy and Data Protection

- Ensure that personal and sensitive data is collected, stored, and processed in a secure and transparent manner.
- Respect user privacy rights and comply with relevant data protection regulations such as GDPR (General Data Protection Regulation) or CCPA (California Consumer Privacy Act).
- **Additional points:**
 - Minimization of Data Collection
 - Informed Consent
 - Data Security
 - Data Retention and Deletion
 - Cross-Border Data Transfer
 - Data Breach Response
 - Data Protection Impact Assessments (DPIAs)
 - Data Governance
 - Data Transparency

Transparency and Accountability

- Transparency in IT infrastructure practices involves being open about policies, procedures, and data usage. It also includes taking responsibility for any errors or security breaches that occur and working to rectify them promptly.
- **Key points to emphasize:**
 - Clear Data Policies
 - Data Usage Notifications
 - Data Sharing and Third-Party Disclosures
 - Regular Transparency Reports
 - Accountability Mechanisms
 - User Access to Data
 - Incident Reporting
 - User Education
 - Ethical Data Use for Research

Accessibility

- Accessibility is a critical aspect of ethical IT infrastructure design and development. Ensuring that technology is inclusive and accessible to all individuals, regardless of their abilities or characteristics, is essential for promoting equality and preventing discrimination.
- **Key considerations:**
 - Universal Design
 - Accessibility Guidelines
 - User Testing with Diverse Audiences
 - Assistive Technologies Support
 - Accessible Forms
 - Accessible Documentation
 - Continuous Improvement

Security and Cybersecurity

- Fundamental aspects of ethical IT infrastructure management. Safeguarding data and infrastructure from cyber threats and unauthorized access is essential to protect user privacy, prevent data breaches, and maintain the trust of stakeholders.
- **Key considerations for prioritizing security:**
 - Risk Assessment
 - Data Encryption
 - Secure Authentication
 - Access Controls
 - Regular Security Audit
 - Patch Management
 - Incident Response Plan
 - Backup and Disaster Recovery

Sustainability and Environmental Impact

- Ethical IT infrastructure considers the environmental impact of technology. Organizations should adopt green IT practices, optimize energy consumption, and reduce electronic waste generation.
- **Key practices to incorporate sustainability:**
 - Energy Efficiency
 - Virtualization and Consolidation
 - Renewable Energy Sources
 - Green Data Centers
 - Energy Monitoring and Management
 - E-waste Management

Fairness and Bias Mitigation

- Critical ethical considerations in the development and implementation of IT infrastructure, especially when it involves automated systems and algorithms.
- **Key points to address fairness and bias:**
 - Data Collection and Preparation
 - Bias Identification and Mitigation
 - Diverse Development Teams
 - Bias Testing and Evaluation
 - Feedback Mechanisms
 - Ethical Guidelines and Policies

Intellectual property

- Essential aspect of ethical IT infrastructure management. Intellectual property rights protect the creations and innovations of individuals and organizations, and it is crucial to abide by the laws and regulations that govern these rights.
- **Key points:**
 - Software Licenses
 - Copyright Compliance
 - Open Source Software
 - Patent Respect
 - Trademark Protection
 - Intellectual Property Audit

AI and Automation

- Ensure that technologies are deployed ethically and do not have adverse effects on individuals, society, or employment.
 - Some key considerations:
 - Ethical Design and Development
 - Human-Centered Approach
 - Privacy Protection

Long-term planning and Scalability

- Build a sustainable and efficient IT infrastructure that can adapt to future needs and growth.
- **Key points to consider in long-term planning and scalability:**
 - Scalability Assessment
 - Flexible Architecture
 - Cloud Solutions
 - Virtualization and Containerization
 - Automation and Orchestration
 - Future-Proof Technologies
 - Performance Monitoring and Optimization

Human-centered design (HCD)

- Prioritizes the needs, experiences, and feedback of users and stakeholders.
- **Key aspects of human-centered design:**
 - User Research
 - User Personas
 - Iterative Design
 - Prototyping and Testing
 - Accessibility and Inclusivity
 - User-Friendly Interfaces
 - User Engagement
 - Empathy and Empowerment
 - Human-Centered Metrics

Ethical Decision-making

- By fostering a culture that encourages employees and IT professionals to consider ethical implications and speak up about potential concerns, organizations can ensure that their technology aligns with ethical principles and values.
- **Key Strategies:**
 - Ethics Training and Education
 - Ethical Frameworks and Guidelines
 - Whistleblower Protection

Compliance and Legal Adherence

- Compliance and legal adherence are critical aspects of IT infrastructure management, particularly concerning data protection, privacy, and IT security. Organizations are bound by various laws, regulations, and industry standards that dictate how they handle sensitive information, safeguard data, and ensure the security of their IT systems