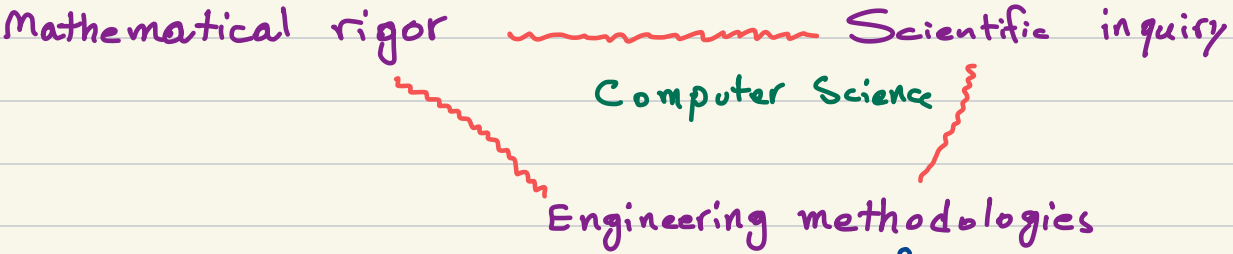
Defenition of computer Science:

What is computer Science?

- It is the study of algorithms, computation, and information processing → Scientific POV
* It's difficult to define because its evolving really fast and it covers a broad scope of diverse disciplines, and there is a deep interweaving of theory and applications
- Focus on designing and developing computer systems and applications. → Engineering POV



* programming is an essential part of Computer Science, but there is more to CS than programming

There are 3 major disciplines in computer Science:

Hardware → physical components of a computer

Software → programs that perform various tasks for users
- Application → programs that directly control computer hardware
- Systems ← programs used to create other software applications
- Development ←

Theory → Algorithms / Computability / cryptography

* In CS we are only limited by our creativity

Social Aspects of Computer Science:

Computer science is a powerful field that can offer many benefits to society. However, it also comes with many challenges and risks that need to be addressed ethically.

* CS requires organizing complex systems while handling unpredictable events and edge cases

CS involves Standardized (concrete) and non-standardized (abstract) concepts.

→ Specific, well-defined, precise and consistent

Generalized ideas
theoretical principles
Allows flexibility

An example for standardization is HTML → color ✓ colour ✗

Social Aspects of Computer Science 🌐🤝

- Addressing the societal impact of automation, privacy, and ethical considerations
- Ensuring technology benefits society while avoiding harmful effects

Introduction to Problem-solving:

Problem solving involves transforming an undesirable state (problem) into a desirable one (solution)

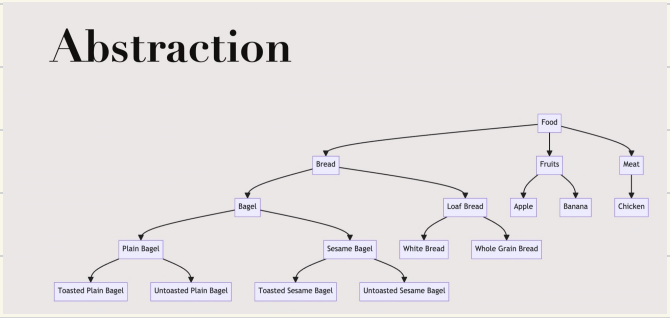
Pólya's Approach: Understand the problem → Devise a plan → Execute the plan → Review and extend the solution

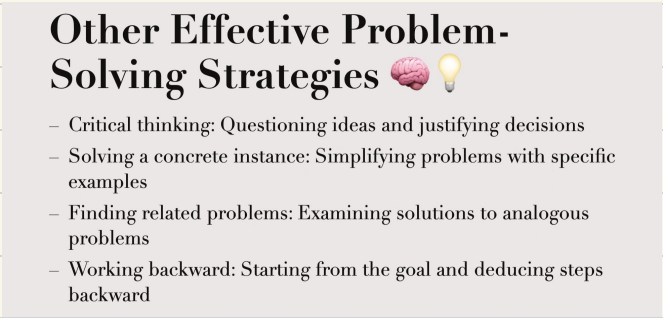
Decomposition: Breaking down complex problems into simpler parts

Heuristics: problem solving techniques yielding good enough answers

Divide and Conquer: Dividing the problem into several sub-problems and solving the sub-problems in order to solve the main problem

Abstraction: A way to simplify complex systems by focusing on the high-level overview rather than the nitty-gritty details. In short, it allows us to solve complex problems by removing unnecessary information.

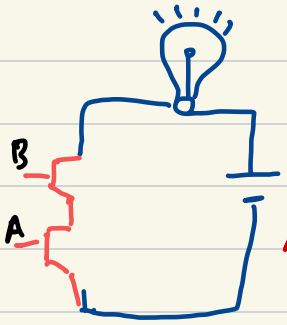




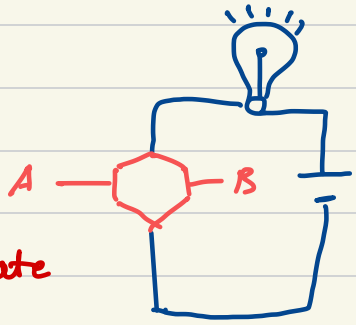
Computers are made from transistor that form circuits.
Transistors are made from semiconductors and they have 3 sections
→ Base, Emitter, Collector
* Semi Conductors are something between Conductors and insulators

Collector → V_{high}
Base → | a well between high and low voltage
Emitter → V_{low}

when current is not flowing, we have 0 and if it's flowing we have 1.



AND Gate

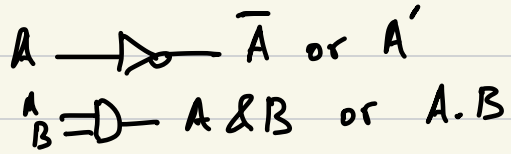


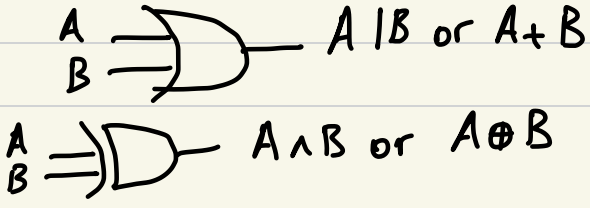
OR Gate

Both A and B have to be 1 to have light

Atleast one of A and B have to be 1 to have light

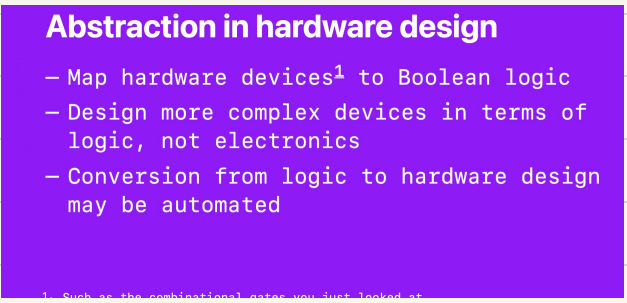
XOR of A and B shows if they have odd number of 1's or not
→ A=1, B=1 → A⊕B = 0 even

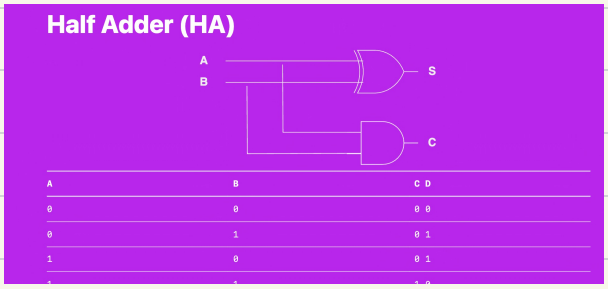




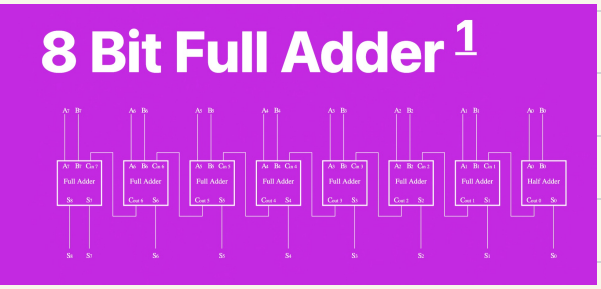
Combinational Circuits: Combining gates using Circuits

XOR is a combinational Circuit → $A \oplus B \equiv A \cdot B' + A' \cdot B$





↳ Gate level Circuit design



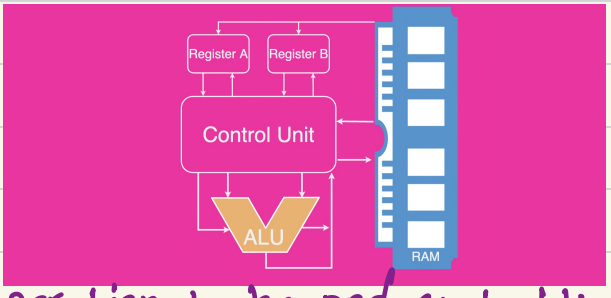
→ more abstraction makes hard problems easier to solve

Arithmetic Logic Unit: ALU gets several inputs (A and B) and a selector input (SEL) that will show which operation we will use on the inputs and then it gives an output.

Opcode shows which operation we are using by converting each operation into a n-bit binary code → (2^n different operations at most)

CPU → Central Processing Unit

Control Unit: The control unit recieves instructions from memory and controls the flow of data within the CPU



↳ it interprets opcode to determine the operation to be performed by ALU or memory

RAM → temporary stores data and instruction for the CPU to assess during execution (data goes from the RAM into CPU's registers)

Instruction Sets are collections of binary-code that CPU's can execute. Two main types → RISC → {simple instructions (less) Faster
CISC → {more complex instructions
Reduced program size

Let's write your first program

Program to add two numbers

1. Load numbers into registers from RAM
   1.1 Locate the number in RAM (use LOAD_A & LOAD_B Opcode)
   `LOAD_A + address 1` -> 0110 1110
   `LOAD_B + address 2` -> 0111 1111
2. Add the values at register A and B
   `Add opcode + 2 register IDs` -> 0001 01 10
3. Save our result into the RAM
   `STORE_B + memory address` -> 1011 1101
4. Stop the program
   `HALT` -> 0100

Congratulations! You just wrote your first program in machine language (code)

```
0110 1110
0111 1111
0001 01 10
1011 1101
0100
```

Moore's law: The number of transistors in a dense integrated circuit (IC) doubles every two years. → after 2Y years
↳ 2^Y times

---

why are computers general-purpose machines?

Because they can handle a broad spectrum of computational tasks thanks to their programmability; unlike "specific-purpose machines" like calculators that are designed for a single task

what are some advantages and disadvantages of biometric-authentication

strong security, Convenience, Accountability → clear audit trail of who accessed what

Privacy concerns, Cannot change, False positives & negatives

* Computer Science is a mix between Sciences and Engineering with theory of computation, Algorithms,... being the Science part and experimenting and testing being the engineering part
So, it's not either Engineering or Science but something in between

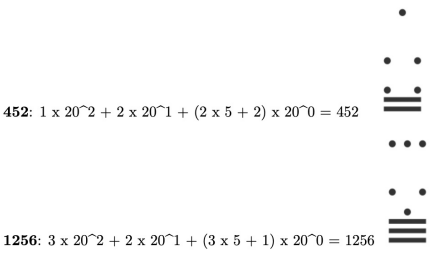
Importance of Standardization

- **Question 4**: Why do we need standardization in computer Science?

Standardization in computer science ensures that systems and software operate consistently and efficiently across different platforms and environments. It facilitates interoperability, meaning different systems can work together seamlessly.

For example, consider HTML, the language used to create web pages. Without a standardized way to write and interpret HTML, every web browser (like Chrome, Firefox, or Safari) might

Mayan numeral system ←
base 20



* if you run out of RAM, your computer tries to use your permanent storage instead and it will result in worse performance and reduced speed. If you run out of permanent storage, you can no longer store and save files.

GB → 10^9 Bytes MB → 10^6 Bytes KB → 10^3 Bytes
GHz → 10^9 cycles per second → measure of speed

kibi bytes → 1024 Bytes

Theory of Computation

- Knowing what a computer can and cannot do helps us solve problems more efficiently
- Some problems cannot be solved by a computer

* You can't write an automated program to check whether a code stops or loops forever in some cases

An Automaton (plural: Automata) is a mathematical model of a computing device.

* We build models for 2 reasons → Mathematical simplicity (Abstraction)
  → Intellectual robustness (making theorems and ...)

Finite State Automata ⟶ FSA

They have 2 classes ⟶ 1. DFA 2. NFA

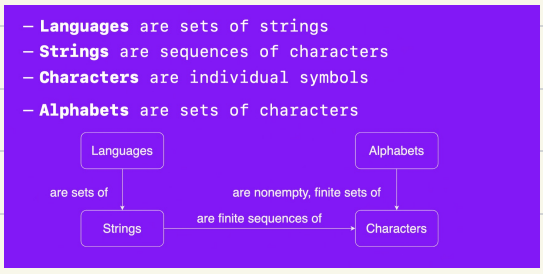
Strings: Sequence of symbols (characters)

Alphabet: Finite set of symbols called characters → $\Sigma = \{a, b, \}$

* A string over the alphabet Σ is a finite sequence of characters drawn from Σ

* empty string is denoted as ε

Languages: A formal language is a set of strings

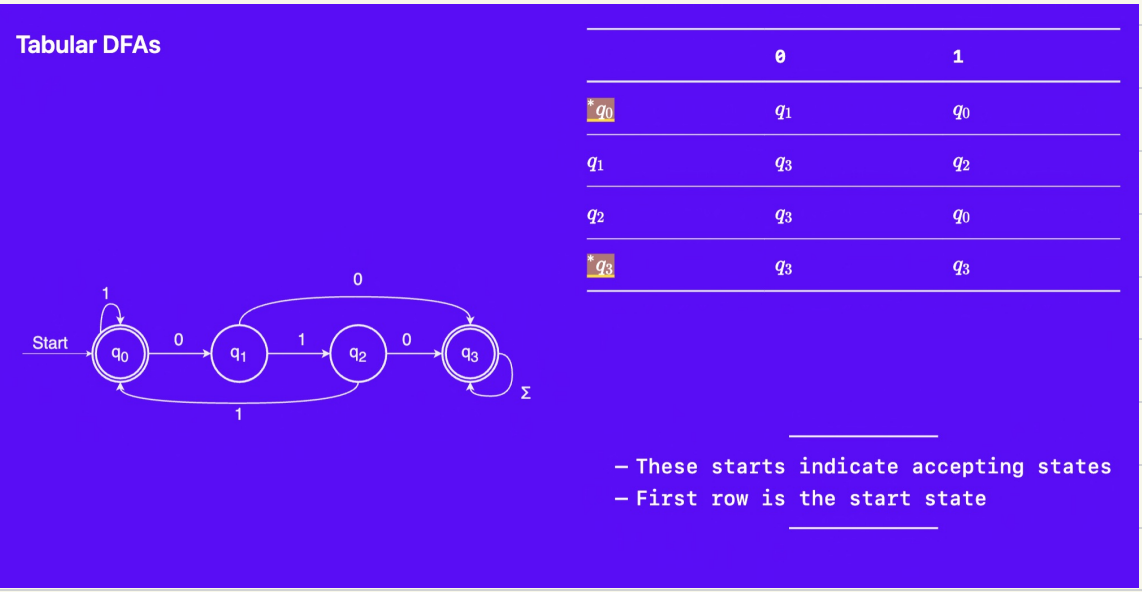


* Each string in a language is finite and therefore, we cannot have a string that is infinitely long.

DFA: Has exactly one starting state and from each state we have exactly one transition for each character in Σ

$D = (Q, \Sigma, \delta, q_0, F)$ — q_0: start state

↳ set of states; δ: transition function; F: Accepting states



	0	1
*q_0	q_1	q_0
q_1	q_3	q_2
q_2	q_3	q_0
*q_3	q_3	q_3

– These starts indicate accepting states
– First row is the start state

$\bar{L} = \Sigma^* - L$

Complement of a Regular language is also a Regular language

↳ closed under complementation

Any NFA can be transformed into a DFA

Any DFA is also considered a NFA

What is an Algorithm? An algorithm is a set of rules that precisely define a sequence of operations

What is a program? A program is an algorithm that has been spelled out in enough detail that a machine can carry it out

Programming is the craft of making programs

Flowcharts are graphical representations of programs

pseudocode is a text-based and closer to natural language

* A command-line is accessed from a terminal

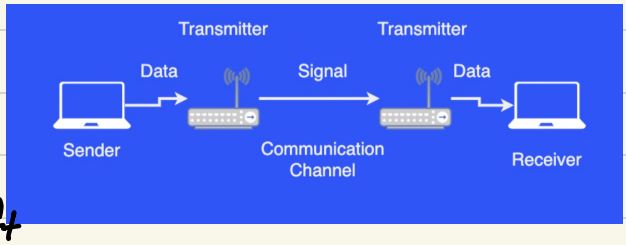
- The CLI is used by typing in commands
  - The CLI lets you know it is waiting for a command by displaying a cursor at a command prompt
- The command is passed to the operating system, which decides what to do with it
  - The biggest decision is security-related: can this user perform that command
- The result(s) of the command (if any) are then displayed under the command
- Each operating system has its own commands and formats for issuing them
  - Some operating systems have several different ones
- Despite these differences, simple commands have roughly the same format

  <command> <options> <target>
- There is always a command, but the number of options and targets can vary
  - And are often optional

CLIs are good to know because:

- They're powerful and they can perform tasks that are hard or even impossible with GUIs
- Using a CLI is faster than writing a program to do the same thing
- Its easier to string together a series of commands as opposed to button clicks

Shanon's communication model:



Network types:

- PAN (personal area networks)

connects devices within 30ft wirelessly

Serves as a single individual

Used for syncing data, wireless printing

the device is listed as a wifi network which you can connect to

- LAN (local area network)

connects PCs in a single building (Schools, work, ...)

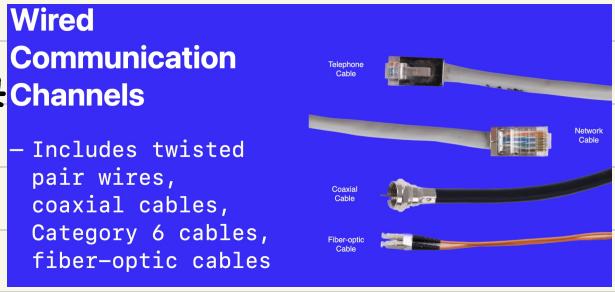
- WAN (wide area network)

covers large areas, consists of smaller networks (Internet, telephone systems, cable tv)

Communication channels:

Communication channel: medium for information transmission

wired channels: Use wires and cables / wireless channels



* wired connections are secure and difficult to tap without physical access or special equipment

pros → shielded, dependable, secure

Cons → Costly, limited mobility, easy to damage

* wireless Coms use Radio signals, micro waves and devices have transceivers for sending and recieving data — includes an antenna

Pros → Mobility, no cables, less power spikes

Cons → Speed, Range, Security, licensing

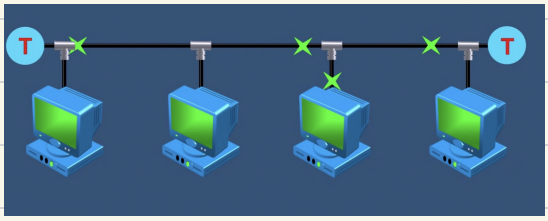
Bandwidth in Communication → Bandwidth → Transmission Capacity

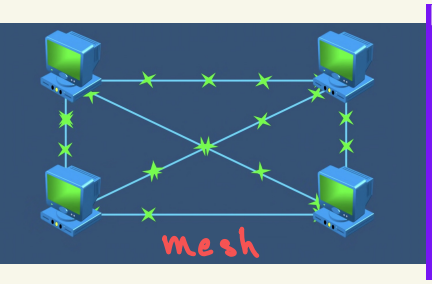
Broadband (At least 25 mbps) Narrowband (slower than 25 mbps)

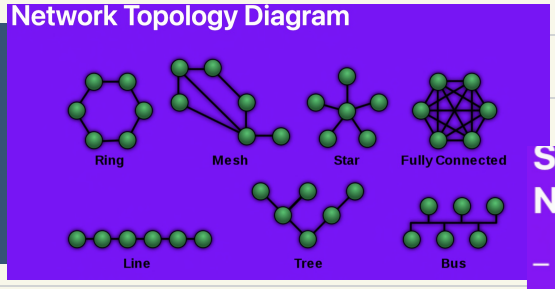
Network Topology:

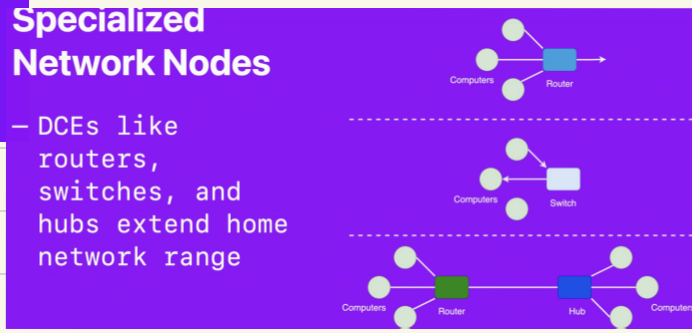
→ Topology is structure and layout of network components there are several types of network topologies:

- Point-to-Point: Connects peripheral devices to a host
- Star: Connects devices to a central device
- Mesh: Connects devices to each other, full or partial (Internet)
- Bus: Connects devices in a linear sequence









Node: Any device on a network

DTE: Data Terminal Equipment stores or generates data → like your phone

DCE: Data Communication Equipment controls data speed, signal conversion, error checking, and routing → like a router

So the difference between DTE and DCE is clear

Router → controls dataflow and acts as a gateway

Modem → converts signals for communication channels

* A switch connects multiple devices to create a network, a router connects multiple switches to create an even bigger network

Internet: Network of networks, designed to be redundant (can reach a computer through multiple paths), hierarchical (email address like @gmail)

* nowadays, there are more nodes than people which shows how big it is
It started in universities

Client Server Network Model: – Server – Client

provides resources to clients

computing device that needs access to resources using servers

* Most servers tend to have a one-to-many relationship with clients

Client-Server Model

Loading a Website

- **Client:** Your computer
- **Server:** Another computer at the URL (e.g. bcs1110.ashish.nl)
- **Request:** Ask for a webpage (with a URL)
- **Process:**
  1. Client asks the server for the information (request)
  2. Request is sent to the server through a sequence of routers
  3. Server decodes the request, sends back the information (response)
  4. Client interprets the response

* A network of networks is the internet but, what we actually mean is:
a WAN in which all computers communicate using a standardized protocol called IP

Standardized Protocol: a protocol specifies how the communication is handled by establishing a standardized vocabulary it usually specifies two things:
– Hardware (frequency at which data is transmitted
– Software (representation of an address (name) in the network)

Accessing Information: URLs

Understanding URLs

- **URL:** Uniform Resource Locator
- **Protocol:** Rules for the information (e.g., http)
- **Domain Name:** Gets converted to an IP address via a Domain Name Server (DNS)
- **IP Address:** Computer-readable (e.g., 142.250.179.142 for google.com)
- **Hierarchy:** Each byte of the address gets more specific
- **Example:** Try 145.20.124.148 (`traceroute ou.nl`)

* TCP/IP is the protocol suite used for the internet

TCP/IP Protocol

- Rules for sending information between computers
- Developed by the US Department of Defense, used by everyone

Application Layer

Transport Layer

Internet Layer

Network Layer

1. **Network Layer:** Captures the physical aspects of data transmission such as the media used (wire/wireless) and the hardware related protocols.
2. **Internet Layer:** Looks after the logical transmission of data. We define the logical address (IP) of our devices connected to the Internet in this layer.
3. **Transport Layer:** This layer is responsible for end-to-end communication, specifically error-free transfer. Example: TCP and UDP protocols.
4. **Application Layer:** This is where your server needs to define its networking preferences such as using SSL etc.

Network layer

Internet layer

Transport layer

Application layer

Application layer: Facilitate data conversion and interpretation for specific applications, define rules for data presentation (HTML/CSS), encryption and session management

Application layer protocols → HTTP (Hyper Text Transfer Protocol)
→ SMTP (Simple Mail Transfer protocol)

Transport layer: TCP → ensures reliable coms
UDP → fast but less reliable
↳ netflix

TCP → separates information into chunks
sends and reassembles packets
ensures information is complete and correct

Transport Layer Protocols
- Responsible for end-to-end communication and data integrity
- Handle data segmentation, sequencing, error correction, and flow control
- **TCP (Transmission Control Protocol):** Ensures reliable communication, confirms data receipt, retransmits lost data
- **UDP (User Datagram Protocol):** Faster but less reliable, suitable for real-time applications like video streaming

* packet is a data parcel sent across a network

Internet layer: Every device that is connected to the internet is assigned a unique 4 byte IP Address

* if we pass the IP Address to our router, it knows how to find the computer with that address

each is a hex code (0-f) ← xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx

IPV4 → 32 bit → 4 bytes → xxx.xxx.xxx.xxx

IPV6 → 128 bits → 32 bytes → xxx*

Static and Dynamic IP: You get your Ip address through the ISP using the internet. Most devices are usually not connected to the internet → we don't assign a unique ip address to each

Static: Ip address doesn't change (useful for web servers)
Dynamic: Ip addresses change frequently (phone, laptop,..)

private IP: Only valid within a local network (household, campus)
public IP: Accessible from anywhere on the Internet

At times it does not make sense to get an IP address from your ISP. For e.g., if you want to access a server (e.g., Maastricht University's Gitlab, Canvas) from the local area network of UM.

* If you google your Ip address, it gives you your pubic address

Domain Names and DNS:

- Domain name: easy to remember Ip addresses
- DNS: Tracks domain names and Ip addresses

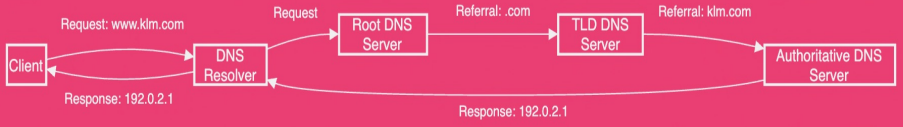
Top level domains ⟶ .edu, .org, ...

- DNs spoofing: Un authorized changes

Internet Addresses: Summary

- Controlled by IP, static and dynamic addresses
- IPv4 and IPv6 standards
- Private IP addresses and public IP routing

DNS Spoofing



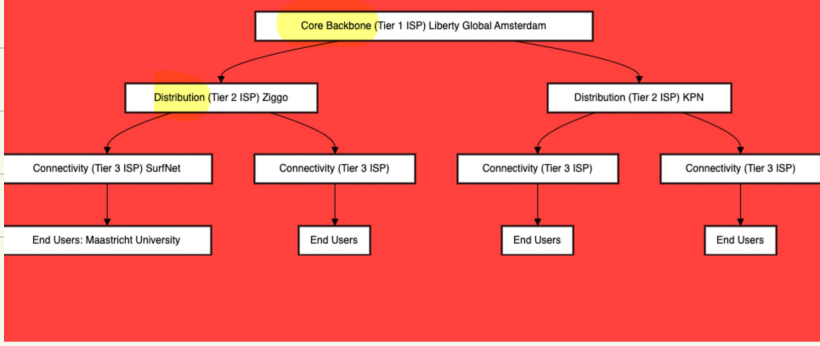
Internet governance:

- no single entity runs the internet
- governed by non profit organizations

Network Layer

- Define the physical connection between devices
- Specify cable types, signal standards, and data rates
- **Ethernet**: Standard for wired connections, using twisted-pair or coaxial cables
- **Wi-Fi**: Standard for wireless connections, using radio frequency (RF) technology
  - **Communication**: Handles communication to/from router

Routers: They help us forward packets to the clients and servers. Also, they help us handle the allocation/deallocation of private Ip addresses.



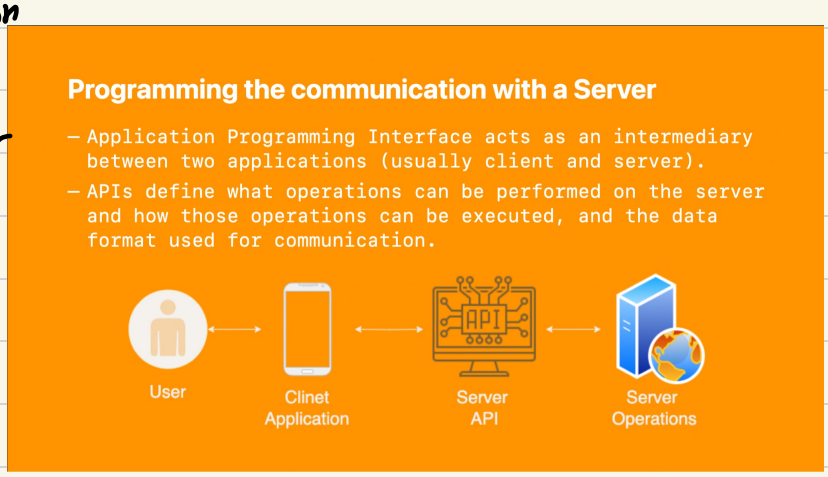
Problem → memory and CPU are expensive

Idea → linking cheap computers into a "giant computer" so that each computer does a different task

Challenges → Computer failure, storing information across multiple computers, waiting for all of them to finish calculations

solution → using AWS AWS services → managing servers, Amazon S3, easy website hosting

APIs: Its used for abstraction It has a list of commands for data access (many companies offer APIs)



Serializing and Deserializing Data: when we want to use data in a different machine we use serialization → most of the time we use JSON (Javascript Object Notation) → data must be well-organized

---

Passwords are like door keys, we don't want other people to have access to our house.

Encryption puts the data in a secret code (cipher-text) and Decryption converts the cipher-text to plain-text using a Key

Caesar cipher is a basic encryption technique that shifts all the letters in a text by some number and the number you shifted by is the Key.

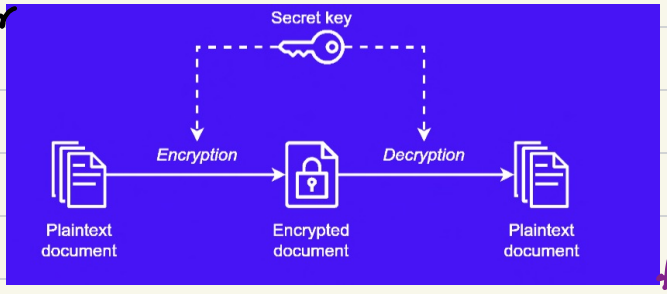
* A more complicated encryption is one-time pad

In cryptography we want to keep the scheme open and public so that other people can find its possible flaws and only keep the keys secret

Classical cryptography (Symmetric): Encrypts and decrypts with the same key (kept secret) → provides secrecy and authenticity
widely used and reliable

*encryption is reliable if the attacker must use Brute Force

Any encryption can be attacked using Brute Force



A^x → double the length ⇒ A^{2x}

→ squares the possibilities

flaw
*you have to share the secret code

HTTPS is an application of encryption in which the browser and the server establish a secure connection with keys only known to them
→ Asymmetric encryption (SSL)

Encryption Application: Disk encryption you can encrypt your disk data so when it gets stolen, they cannot see your files

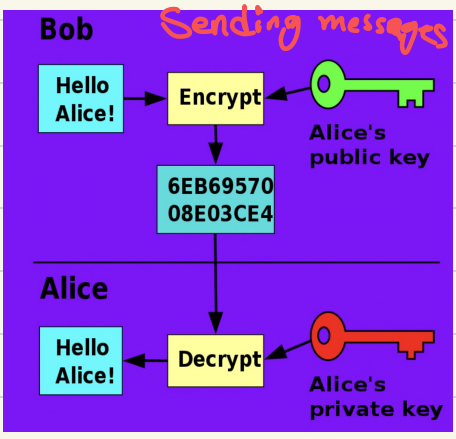
Modern Cryptography (Asymmetric): significant jump compared to symmetric crypto, instead of a single key, there are two keys public and private
- private
- public

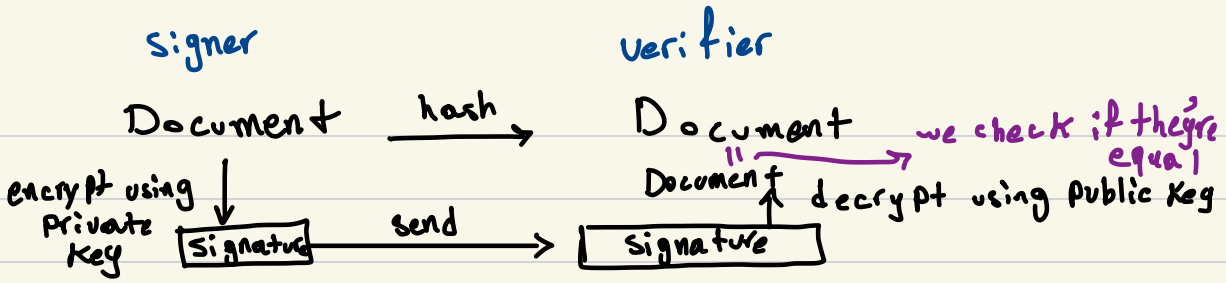
} → they are mathematically linked

Public key encryption: A website has a private key and gives out its public key
public key lets you check that the site is who they say they are
Public key is issued by a certificate authority to limit impersonation
*private keys are used to generate a shared key for the actual encryption

Public Key Application: Digital Signature

- Generate a pub/priv pair on a machine
- Publish the pub key
- Priv key never leaves the machine
- Digital signature: use the priv key to "sign" a PDF, creating a signature
  - 1 Anyone can use the pub key to verify signature
  - 2 Only the holder of the priv key can make the signature





---

*Facebook messenger and Gmail send/recieve data in plain-text
End-to-end encryption means the message is encrypted when it leaves your phone and decrypted only on your friend's end. The ISP sees gibberish.

-Incognito and private browsing are the same as normal browsing but they don't keep the browsing history.

-Don't use the same password on every website since some websites store passwords as plain text and if your password gets compromised, hackers could extract your data on other websites

Hash: Mathematical function that turns plain text into random looking info
Idea: Instead of storing passwords, websites store hashed passwords and when you enter your password, it will hash it and compare the hash with the stored hash

Multi-Factor Authentication: Knowledge → something you know; Possession → something you have; Bio-metric → something you are
you need to have access to at-least 2 of these factors → 2FA

Attacks to get passwords — Phishing, Dictionary Attack

Phishing → bad guy tricking you to enter your password in their website
↳ Redirects into a fake website, impersonates someone else
* always check the URL Fake ATMs

HTTPS is more secure than HTTP: – Domain verification → prevents phishing – Data encryption → safe gaurds transmitted data

*Users must verify URLs to prevent phishing

Dictionary Attack

- Try every known password
  - as if they are trying all the words in a dictionary
- Try common passwords as guesses ("password" or "password1")
- 1 guess/second = 31 million guesses per year

> fails mostly, but works some percentage of the time

Typical Attacks - Bulk

- Typically the bad guys are not crafting some attack just for you
- They send out millions of generic attacks, just snaring who falls for it
- If you avoid the most common errors, you will probably be fine

cracking: if hacker already have encrypted / hashed passwords from a site they can do billions of guesses per seconds most passwords can be guessed within days

Weak Passwords

- How would a hacker guess password?
  - 1 Dictionary of words
  - 2 List of commonly used passwords from other sites
  - 3 Heuristic changes
    - batman
    - generate automatic variations: batman1, batman2, batman3,bat.man,iheartbatman

> Avoid anything from common list

- Second thing to log in
  - Typically something you
    - SMS
    - OTP generator App (Microsoft Authenticator, Apple Passwords)
    - U2F (Apple Passkey)

What to do

- **Counter: programmers can build in a short delay so it takes longer to try passwords**
- **Counter: programmers can limit number of attempted logins**
- **Counter: make your password unique AND long**
- **Counter: Two-Factor Authentication (2FA)**

Future of 2FA → U2F (device acts as the 2nd factor)

Issues with SMS based 2FA

- 3 weaknesses
  - 1 Bad guys could trick your mobile provider: FTC's lead Technologist gets hacked
  - 2 Phishing
  - 3 Malware on phone (more on this in a bit)