Day 1 - 16th May 2025

Your name: Vineeth Bharadwaj Y (bharady@)

Day 1 - bharady@ - Y Vineeth Bharadwaj

SDLC:

Task: 1

What is SDLC?

The Software Development Life Cycle (SDLC) is just a process that software development teams follow to maintain, test, create, and design software applications in detail. It involves several stages including planning, requirement analysis, system design, development, testing, deployment, and maintenance. SDLC develops software systematically, efficiently, also with minimal errors. It helps improve project management as well as meet customer expectations. It is also made possible to deliver software products of high quality within budget and on time. Teams may opt to follow different SDLC models such as Waterfall, Agile, or Spiral dependent on the project requirements and complexity, and each phase has its specific goals as well as deliverables.

Task 2:

Why is SDLC ?

Imagine trying to build a house without any plan then you’d end up with such a mess also waste money plus time. Software shares a similar fate. A roadmap is what is given by the SDLC. Teams can stay organized using it; they can also dodge errors. It also helps them build software that works in actuality and solves problems that are real. For success, thinking before doing is important.

Task 3:

What are the stages of SDLC ? write 2 lines about each.

There are 7 stages of SDLC

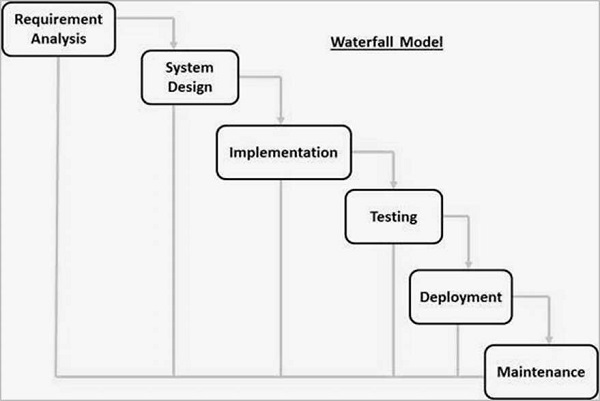
1. Planning Stage - this is an initial stage where we set out a plan on what we will build, set a goal, and form a team
2. Requirement Gathering Stage - We need to gather and understand what is expected from the software
3. Design: we need to create a blueprint on how sthe oftware will look
4. Development: Write code based on our design
5. Testing: We test our developed code, fix bugs if any
6. Deployment: Launch the software so users can start using it
7. Maintenance: Fix issues, update features, and keep the software running smoothly.

Task 4:

SDLC Models:

List them , description - 4 lines min and with a image

1. Waterfall Model:



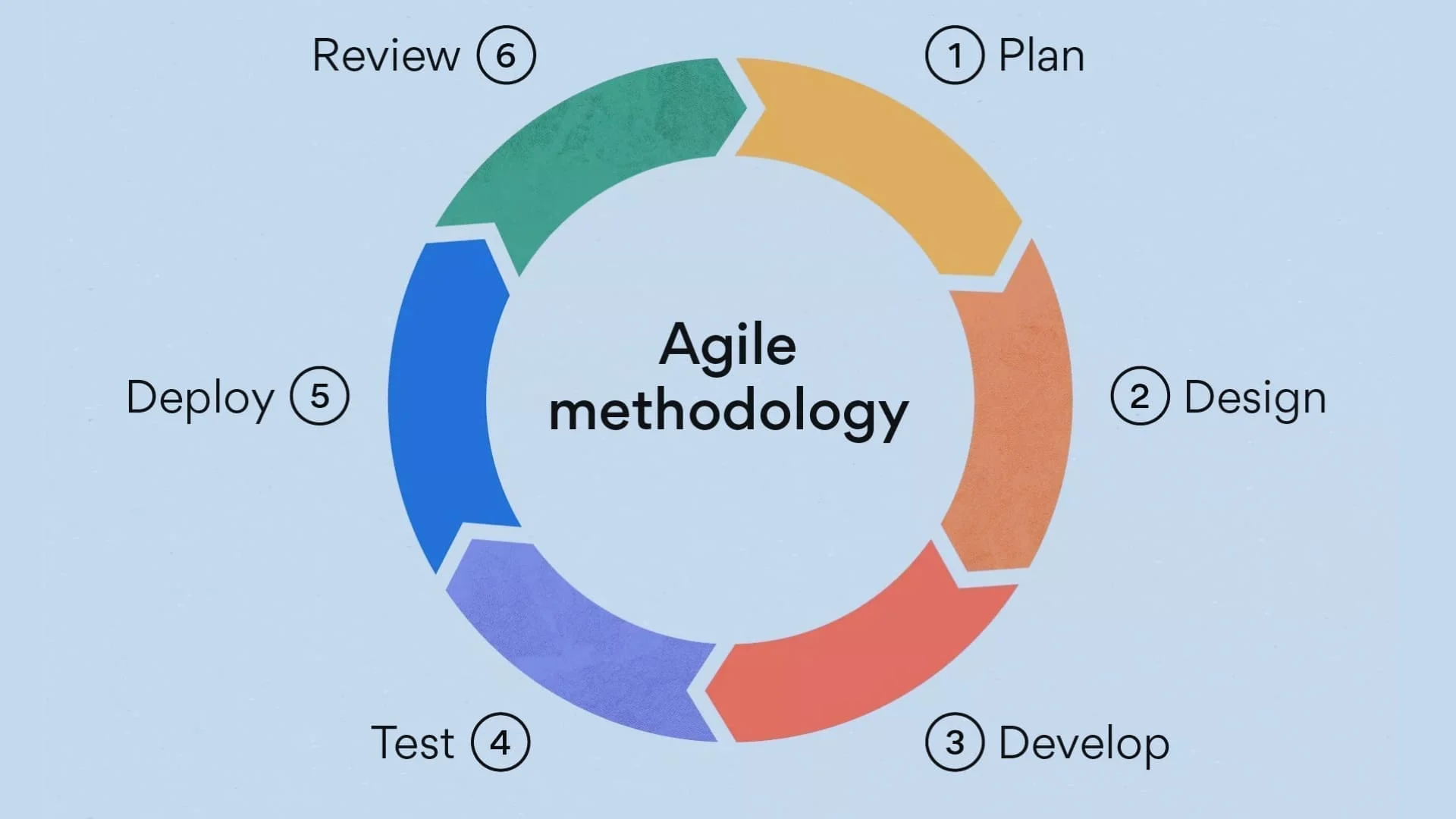
In this model where each stage happens step by step, like a waterfall. You finish one stage completely before moving to the next. It’s simple and easy to follow, but changes are hard to make once development begins.

Applications : Suitable for projects with clear, fixed requirements from the beginning.(Govt)

Advantage: Simple and easy to understand and manage due to its linear structure.

Disadvantage: Difficult to accommodate changes once the process has started.

1. Agile Model:

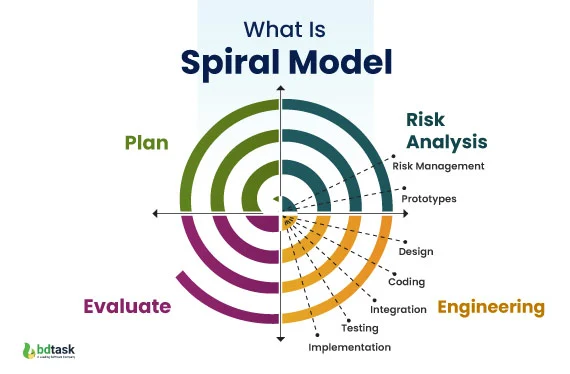


Agile is flexible and works in small cycles called sprints. Teams build software in parts, get quick feedback, and keep improving. It’s great for projects where needs keep changing. Communication and teamwork are key in Agile.

Applications : Ideal for projects with frequently changing requirements (startups)

Advantage: Highly flexible and adaptive to changing requirements through continuous feedback.

Disadvantage: Less predictable due to its iterative nature, which can complicate planning and budgeting.

1. Spiral Model:

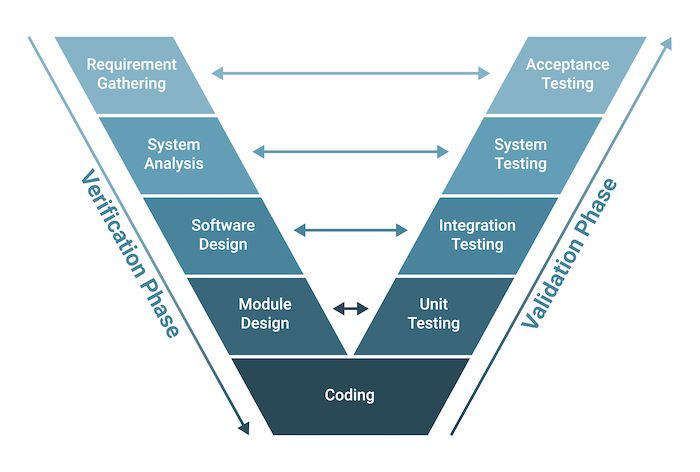
This model mixes design and prototyping in stages. It’s useful for big, risky projects. Each loop (or spiral) includes planning, risk checking, development, and testing. It helps find problems early and reduce risks.

Applications**:** Best for large, complex, and high-risk projects (aerospace, defense).

Advantage: Strong focus on risk assessment, which helps reduce project risks early.

Disadvantage: Can be complex and costly due to repeated iterations and risk analysis.

1. V-Model (Validation and Verification):



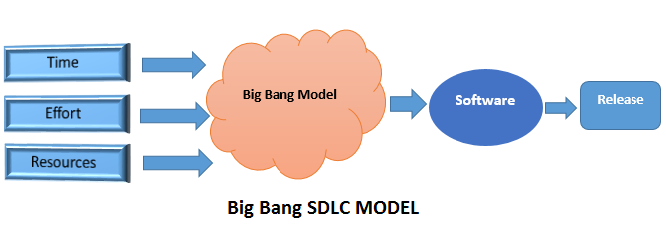
It’s like the Waterfall model but with a twist: every development stage has a matching testing stage. That means you plan tests early and test more. It helps improve quality but doesn’t handle changes well.

Applications: Suitable for projects that require high quality and reliability (medical software, critical systems).

Advantage: Early test planning leads to better defect detection and higher quality.

Disadvantage: Not suitable for projects where requirements are likely to change.

1. Big Bang Model :



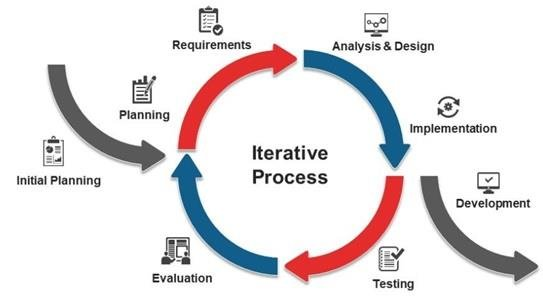
There’s no proper planning here. Developers start coding with little idea of what the final product should be. It’s simple but very risky and only suits small projects or experiments where requirements are unclear.

Applications: Best for small projects, academic experiments, or when requirements are not clear at the beginning.

Advantage: Simple and quick to start, with minimal initial planning.

Disadvantage: Very high risk of project failure due to lack of structure and planning.

1. Iterative Model :



You build a simple version of the software first, then keep improving it step by step. It’s good when you can’t define all requirements upfront. Feedback and updates help shape the final product over time.

Applications**:** Suitable for projects where requirements are unclear at the beginning and need regular feedback to evolve (product development, web applications)

Advantage: Allows gradual refinement through repeated feedback, improving the product over time.

Disadvantage: Can lead to scope creep and project delays if iterations are not well controlled.

What is Scrum ?

Scrum(Framework) is a way for teams to work together more effectively. Instead of trying to do everything at once, the team breaks the work into smaller parts, focuses on one part at a time, and regularly checks progress. This helps the team stay organized, adapt to changes, and deliver better results step by step.

What is Sprint ?

A Sprint is a short, focused period in which the team works on a specific set of tasks or goals. The goal of a Sprint is to complete a usable piece of work by the end of that period. After each Sprint, the team reviews what they’ve done, gathers feedback, and plans what to do next. This cycle keeps the team moving forward and continuously improving.

Do’s :

Focus on the Sprint Goal

Collaborate and communicate with the team

Deliver work that meets the Definition of Done

Dont’s :

Don’t add new tasks or change goals mid-sprint

Don’t skip daily stand-up meetings

Don’t hide blockers or delays

### What is Backlog ?

A backlog is a list of all the work to be done in a project. It includes features, fixes, and tasks prioritized by importance.

What is Story ?

A user story is a short description of a feature from the user’s point of view. It explains what the user wants and why.

What is Artifacts ?

Artifacts are key documents in Scrum that help manage work

### What is Product Backlog ?

A list of all features, fixes, and improvements needed in the product.

### What is Sprint Backlog ?

A list of selected tasks the team will complete during the current Sprint.

### What is Increment ?

The working product or feature completed during the Sprint that is ready to use or show.

What is Burn-Down Chart ?

A chart that shows how much work is left in the Sprint.

The line goes down each day as tasks are completed.

What are Ports and Protocols?

Protocols are rules for how data is sent and received over a network.

Ports are like doors that help send the data to the right service on a device.

What are the types of servers ?

Servers are computers that provide services or data to other devices (clients).

Common types include Web Server, Database Server, Mail Server, and File Server

What do you know about DNS? Domain Name System

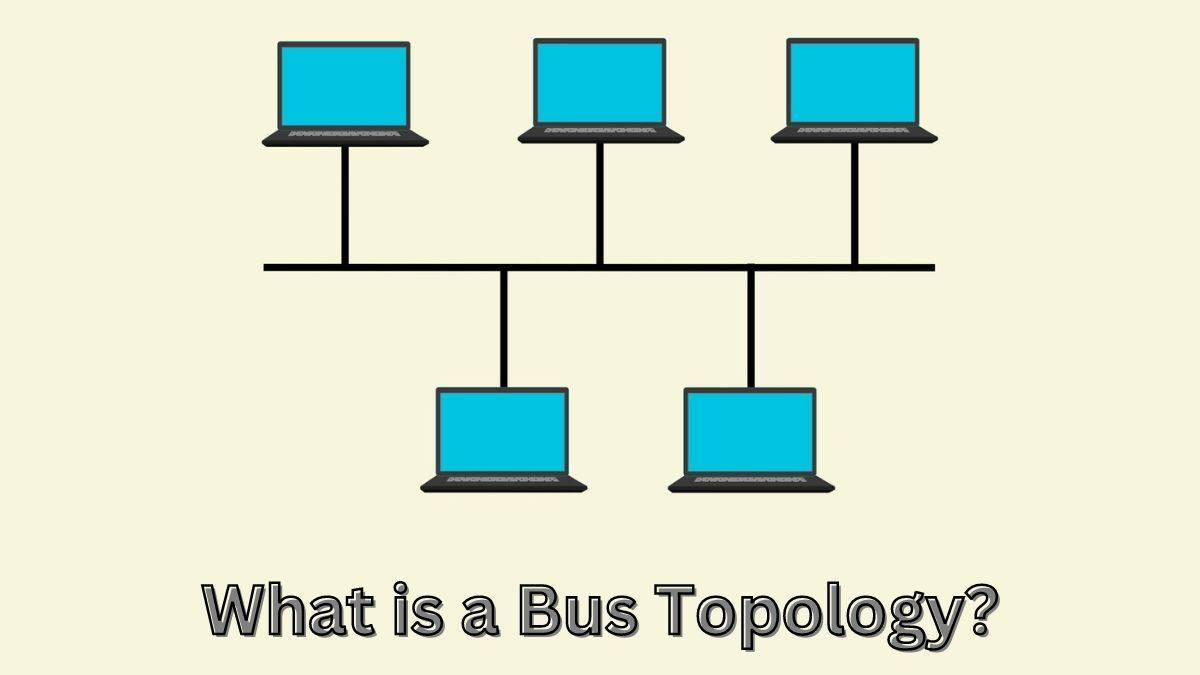
DNS translates website names (like amazon.com) into IP addresses that computers use.It helps users reach websites without remembering number-based addresses.

What is Network Topology ? Explain types of Topologies with diagrams

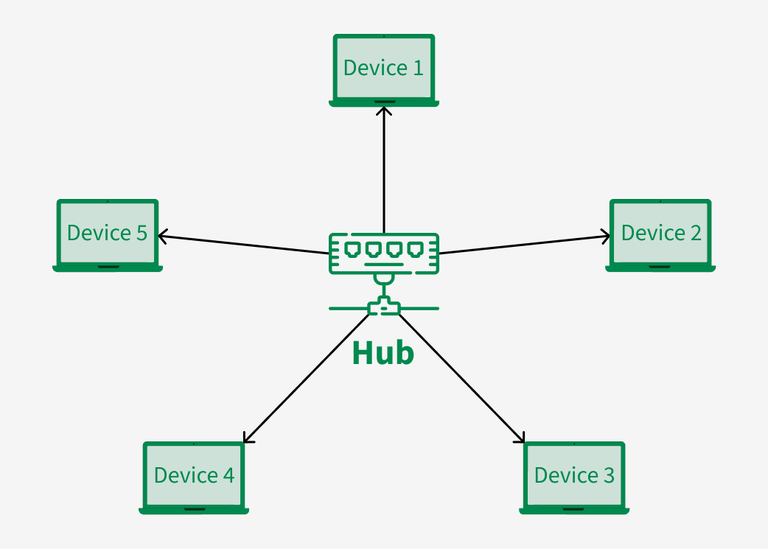
Network topology is the way computers and devices are connected in a network.

Common types include Bus, Star, Ring, Mesh, and Tree.

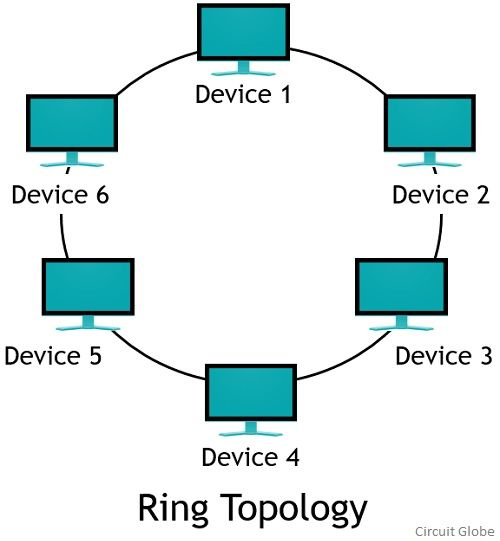
Bus Topology

  
 All devices share a single communication line — like old-school LANs.

Star Topology

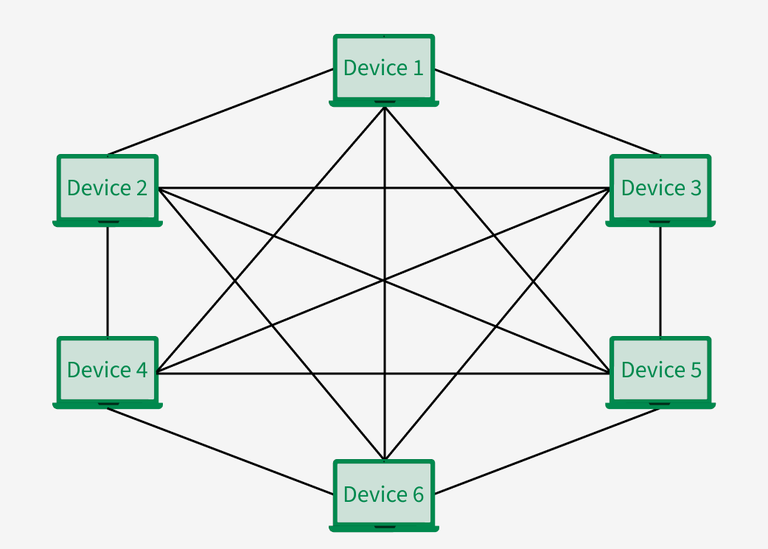
  
 All devices connect to a central hub — like home Wi-Fi routers.

Ring Topology



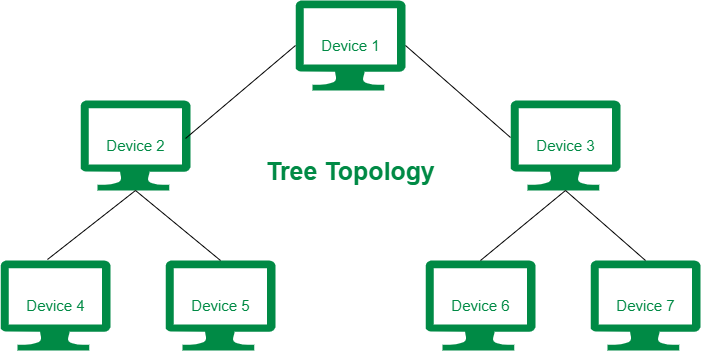
Each device connects to two others in a circle — used in some office networks.

Mesh Topology



Every device connects to every other — used in military or mission-critical networks.

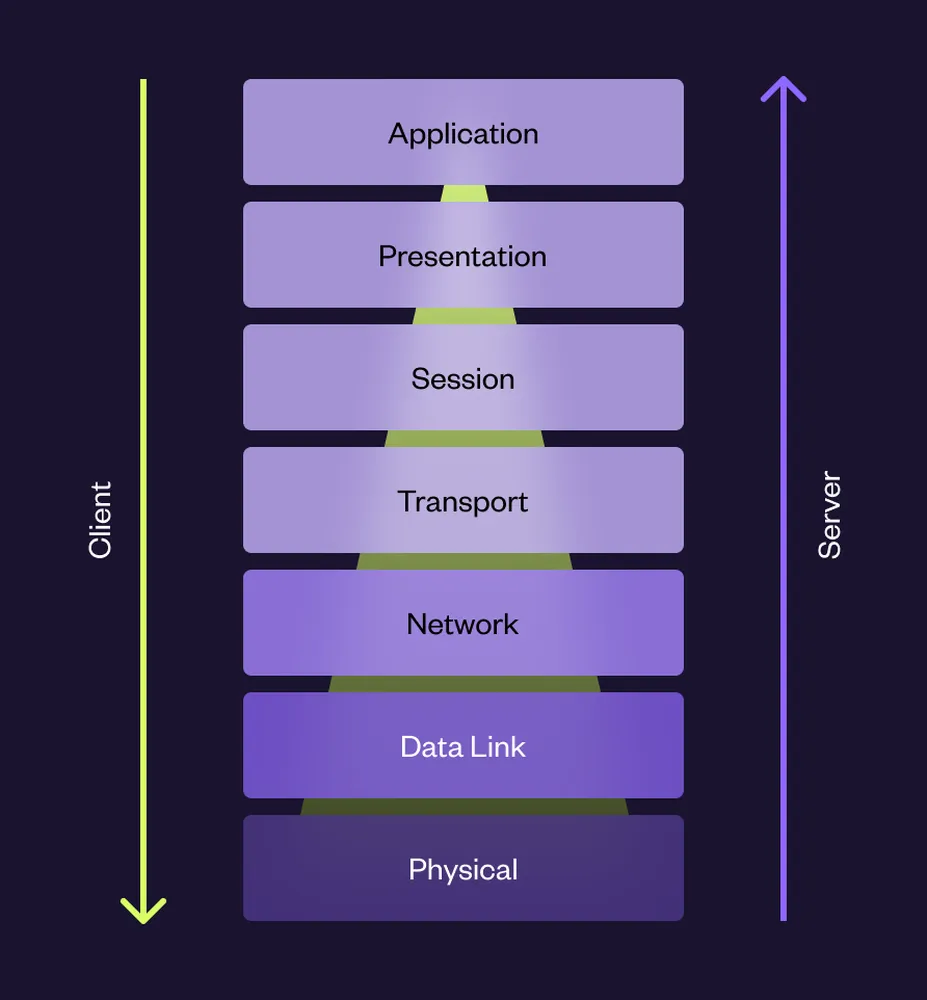
Tree Topology



A mix of star and bus — common in large corporate networks.

### What is OSI Model?

The OSI (Open Systems Interconnection) model is a framework that shows how data moves through a network in 7 layers — from one device to another.



### 

### 

### 7 Layers of the OSI Model :

1. Application Layer – What the user sees (e.g., web browsers, email apps).
2. Presentation Layer – Formats data (e.g., encryption, compression).
3. Session Layer – Manages connections (start, maintain, end sessions).
4. Transport Layer – Breaks data into smaller pieces (e.g., TCP/UDP).
5. Network Layer – Finds the best path (e.g., IP addressing, routing).
6. Data Link Layer – Moves data between nearby devices (MAC addresses).
7. Physical Layer – Transfers raw bits through cables, Wi-Fi, etc.