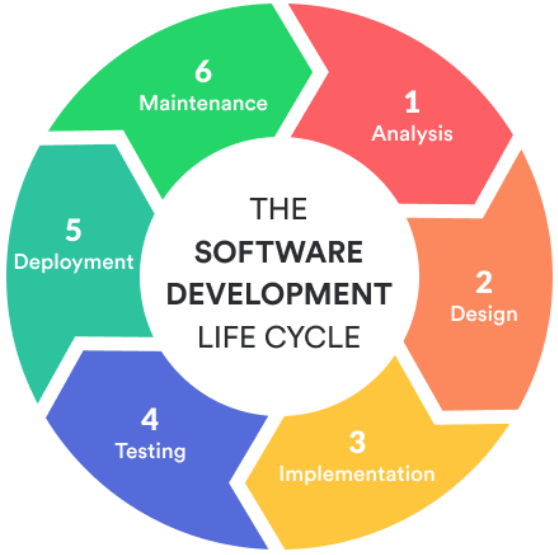
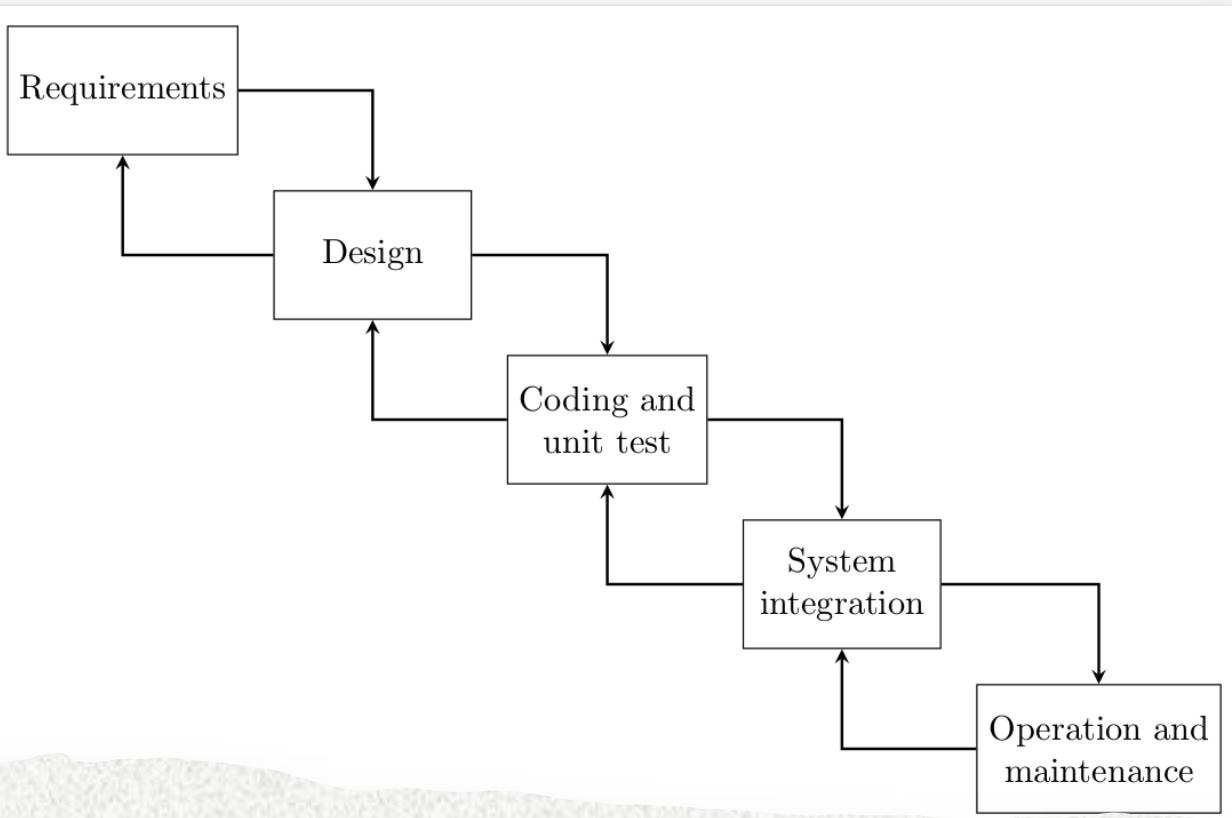
Software Development and Design Basics

By Khaja

Software Development Life Cycle





Waterfall Model

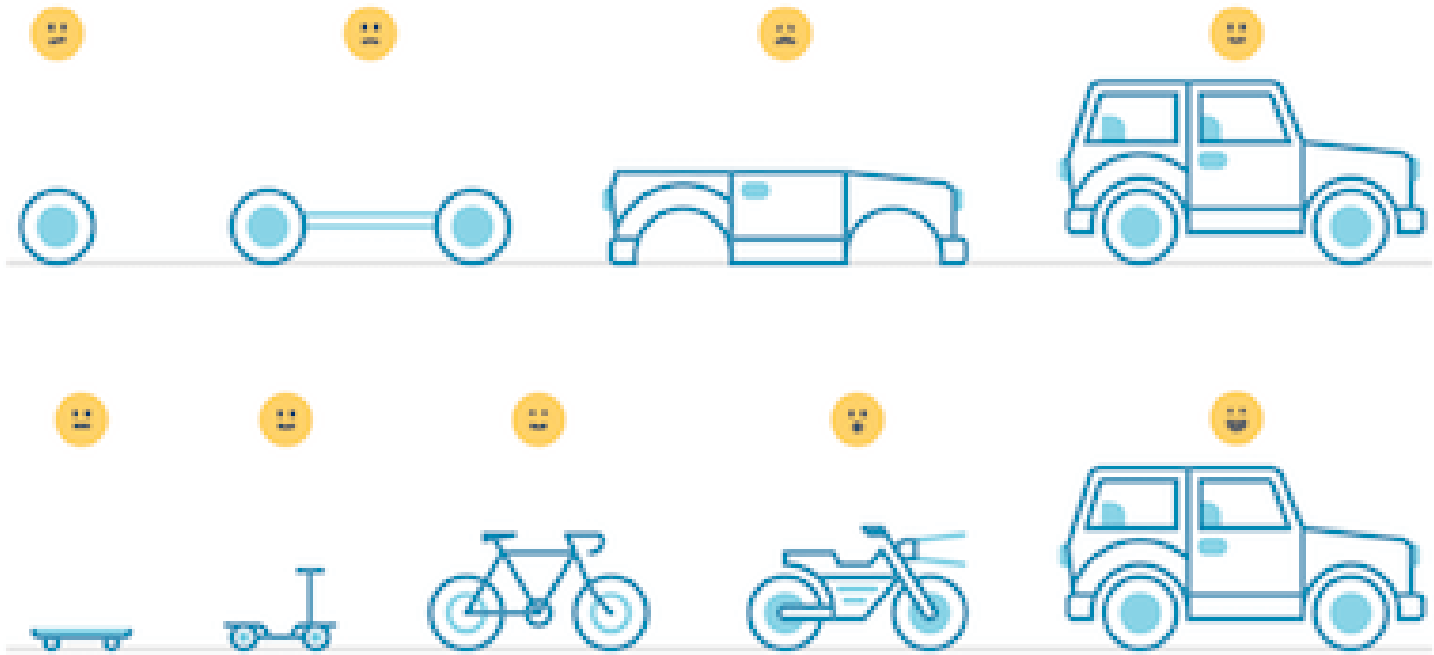
Agile Principles

- Individuals and Interactions over Processes and Tools
- Working Software over Comprehensive Documentation
- Customer Collaboration over Contract Negotiation
- Responding to Change over Following a Plan

Agile Based Software Development

- Agile Software Development is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and customer satisfaction
- Scrum is a popular Agile framework that structures work in fixed-length iterations called sprints. Sprints are usually two to four weeks long. Scrum includes defined roles (Product Owner, Scrum Master, and Development Team), events (Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective), and artifacts (Product Backlog, Sprint Backlog, and Increment).

Waterfall vs Agile

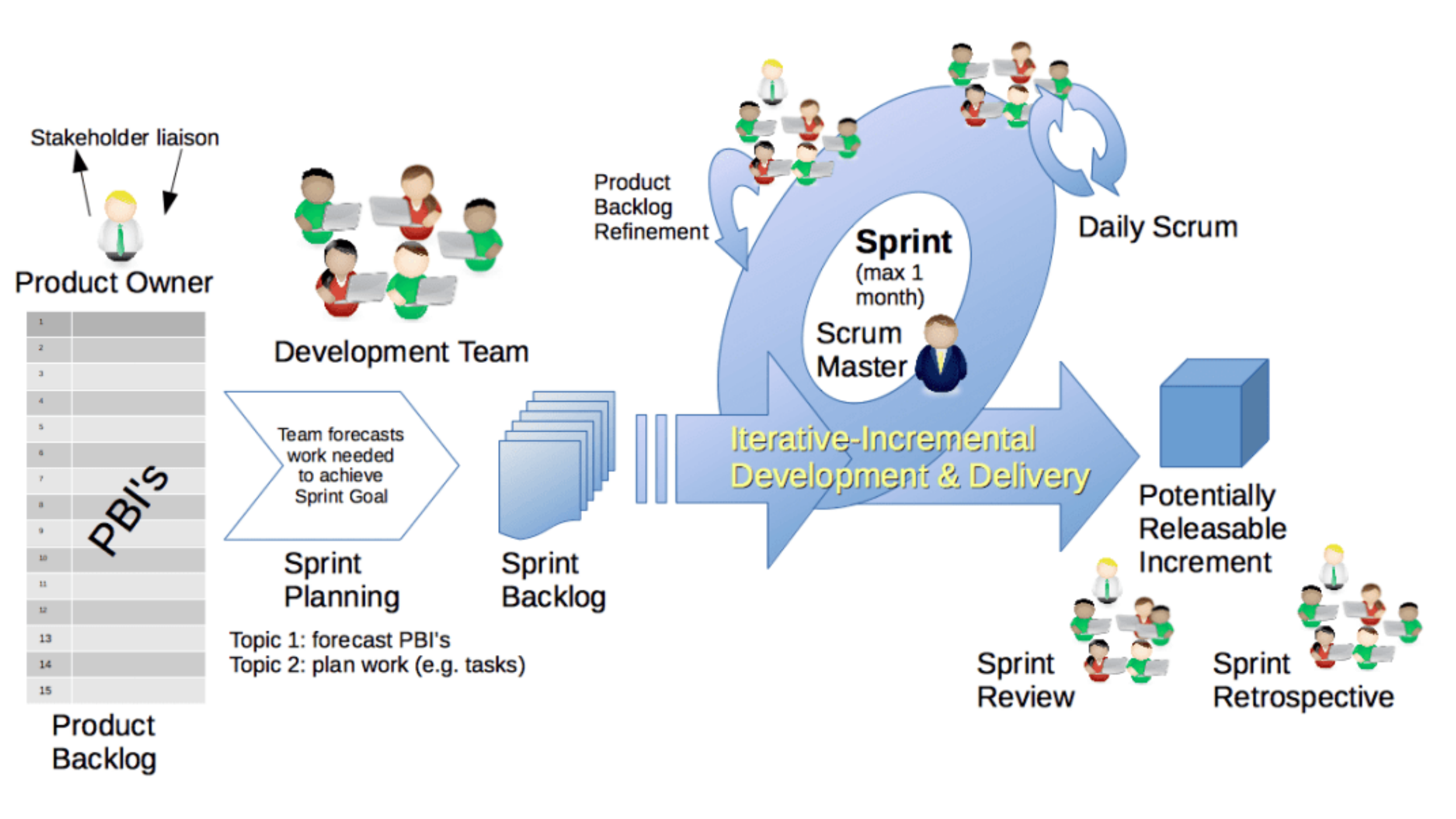


Roles in Scrum

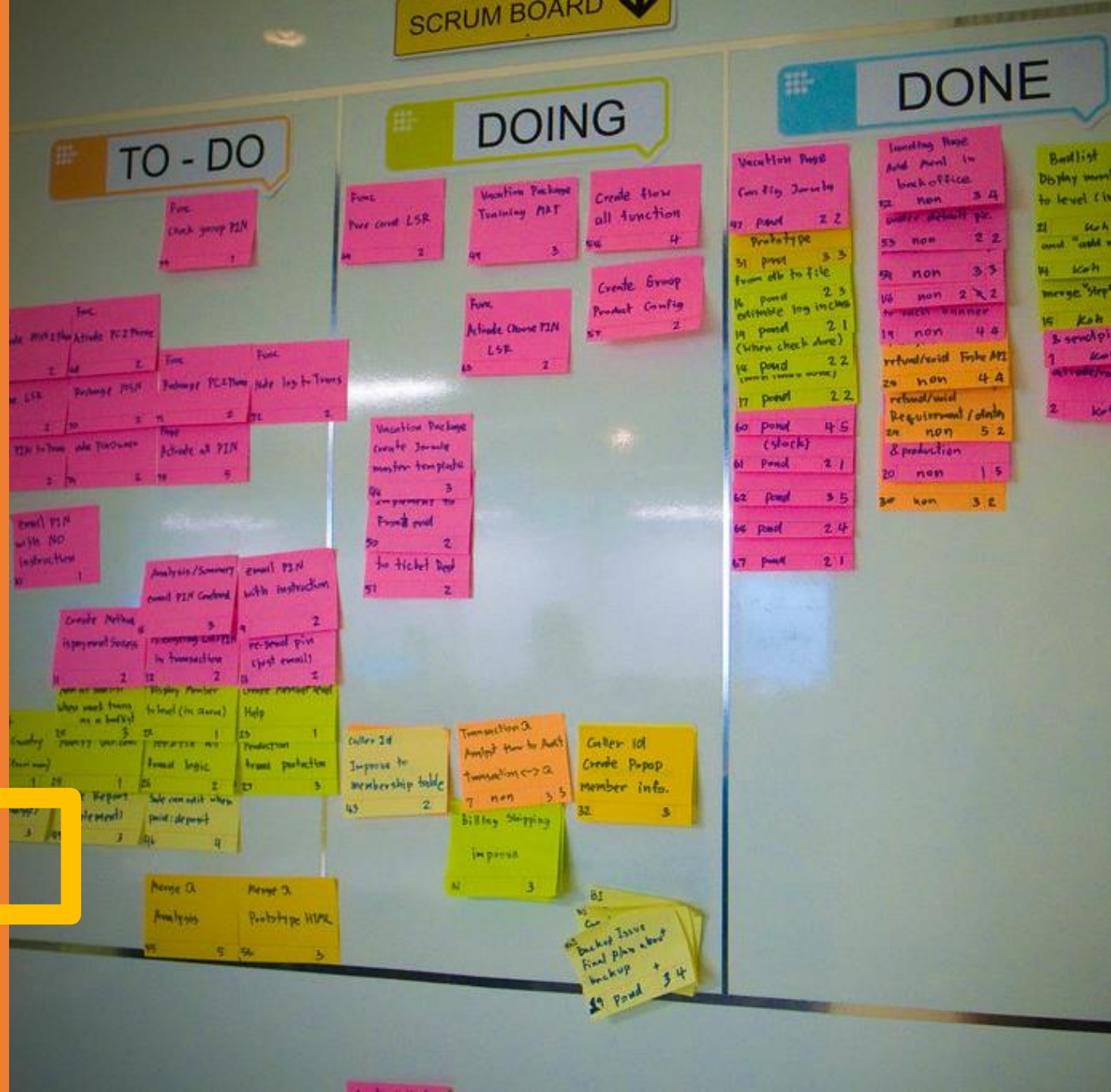
- Product Owner:
  - Represents the stakeholders and is responsible for defining and prioritizing the product backlog
  - Ensures that the team is working on the most valuable features and user stories
  - Makes decisions about the product and accepts or rejects work results
- Scrum Master:
  - Facilitates and ensures the Scrum framework is understood, implemented, and followed
  - Acts as a servant-leader to the team, removing impediments and helping the team improve
  - Facilitates Scrum events (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective)

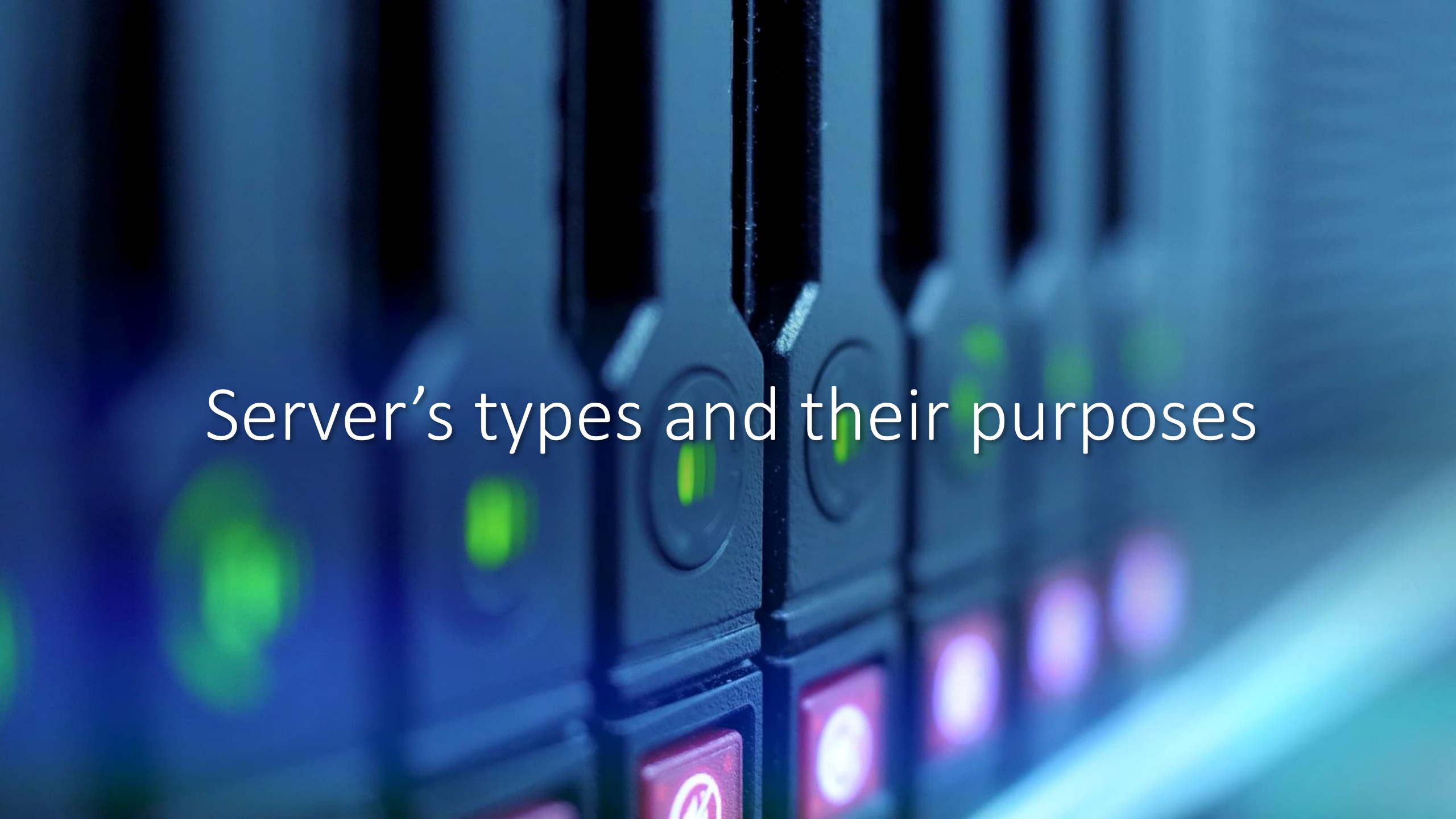
Roles in Scrum (Contd)

- Team Member:
  - Cross-functional: Comprising members with all the skills necessary to deliver a potentially shippable product increment.
  - Self-organizing: Determines how to accomplish the work without external interference
  - Delivers a potentially shippable product increment at the end of each sprint
  - Collaborates with the Product Owner to understand and deliver the highest-priority product backlog items


Stakeholder liaison
Product Owner
Development Team
Product Backlog Refinement
Sprint
(max 1 month)
Scrum Master
Daily Scrum
PBI's
Team forecasts work needed to achieve Sprint Goal
Sprint Planning
Sprint Backlog
Iterative-Incremental Development & Delivery
Potentially Releasable Increment
Sprint Review
Sprint Retrospective
Topic 1: forecast PBI's
Topic 2: plan work (e.g. tasks)
Product Backlog

SCRUM BOARD





Server’s types and their purposes

Web Servers

- Purpose: Web servers are designed to host and serve web applications and content. They handle HTTP requests from clients (browsers) and respond with the requested web pages.
- Examples: Apache, Nginx, Microsoft Internet Information Services (IIS)

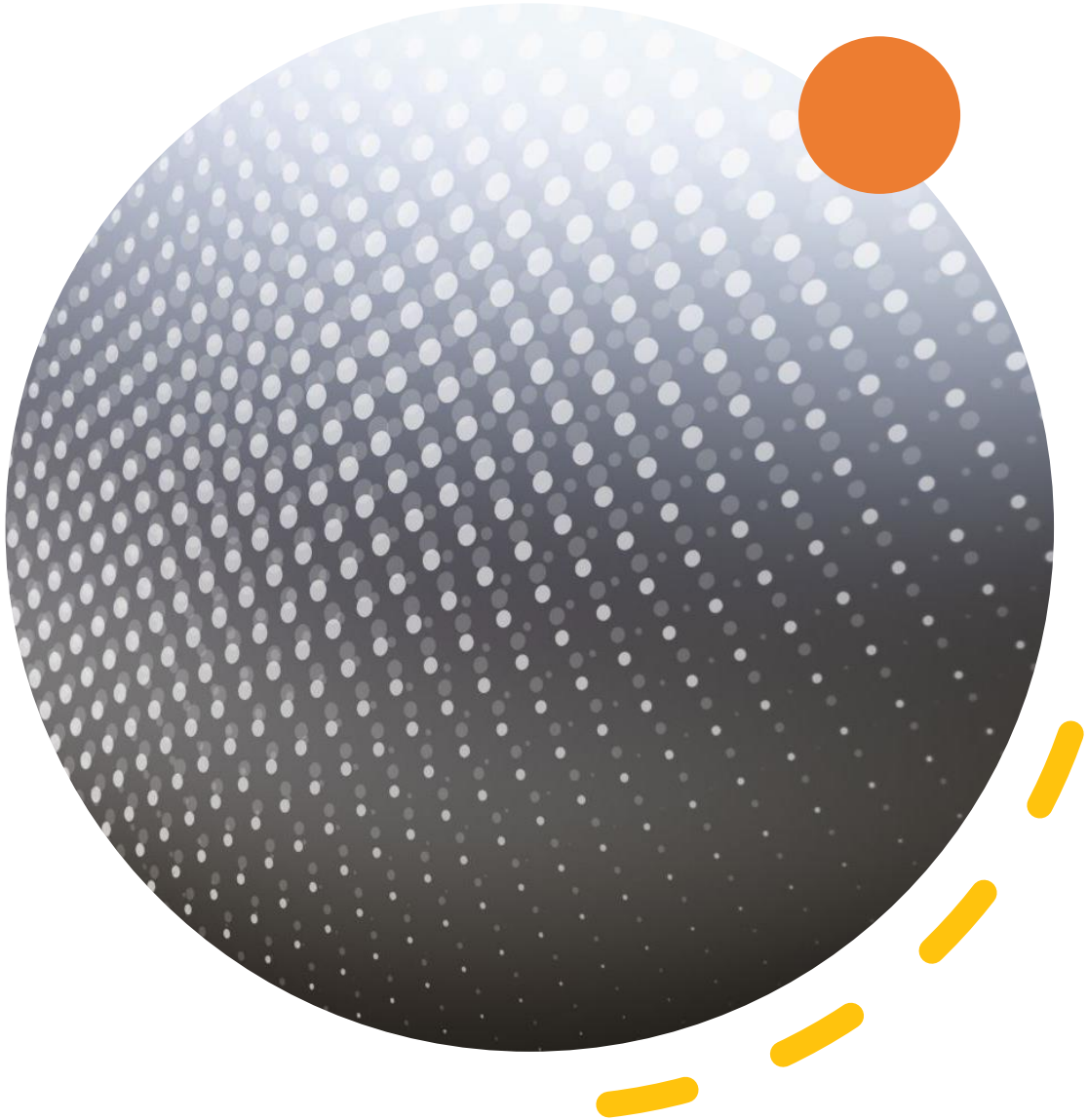
Application Servers

- Purpose: Application servers are responsible for running the business logic of applications and managing the communication between the application and database servers.
- Tomcat, JBoss, WildFly, Microsoft .NET, Node.js.

Database Servers

Purpose: Database servers store and manage data. They handle database queries, updates, and provide a secure and efficient way to manage and retrieve data.

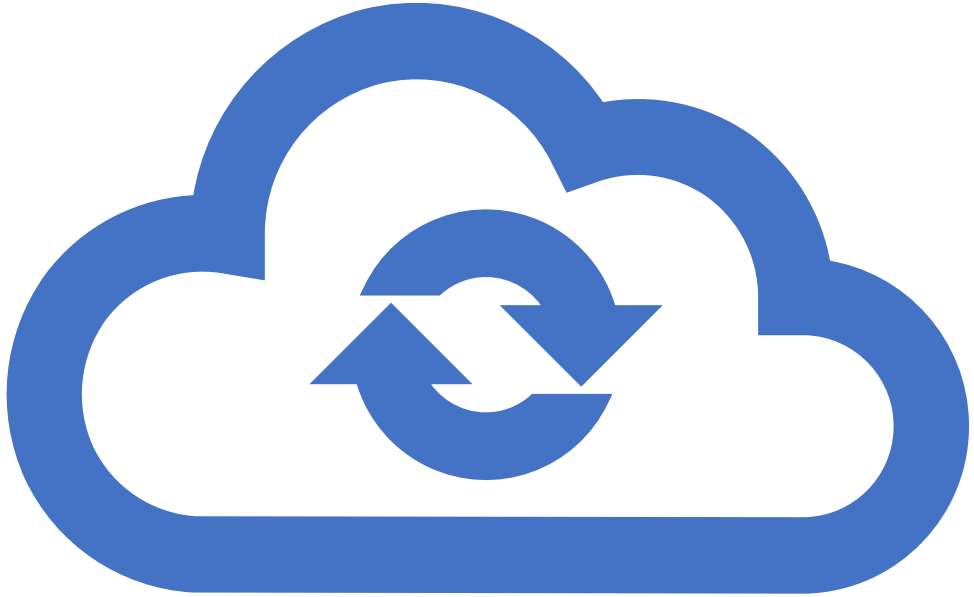
Examples: MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server.



File Servers

- Purpose: File servers store and manage files, allowing users to access and share files across a network.
- Windows File Server, Samba

FTP Servers



- Purpose: FTP (File Transfer Protocol) servers enable the transfer of files between systems over a network.
- Examples: vsftpd, ProFTPD, FileZilla Server

Proxy Servers

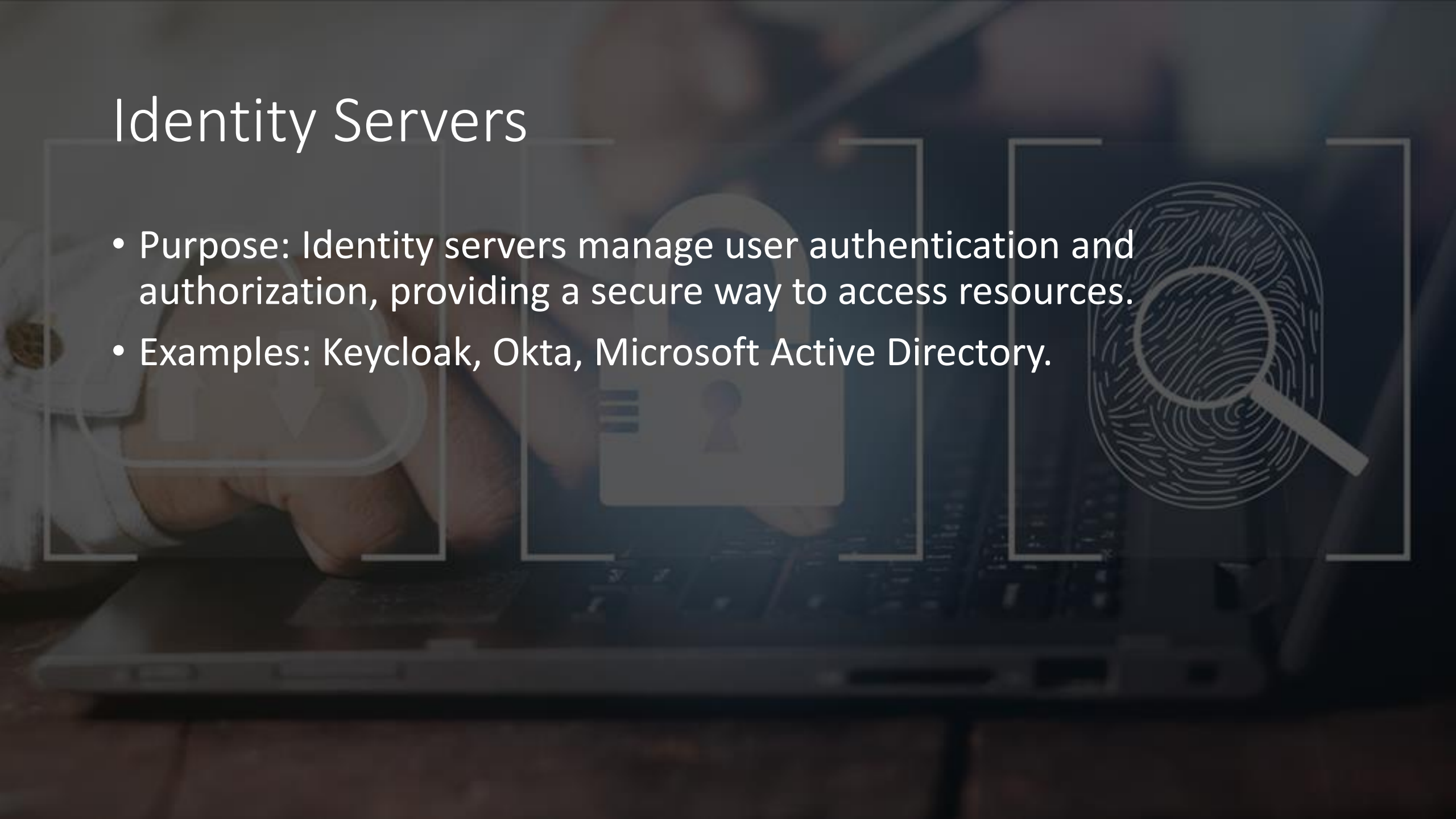
- Purpose: Proxy servers act as intermediaries between clients and other servers. They can be used for security, load balancing, content caching, and anonymizing internet access.
- Examples: Squid, Nginx (as a reverse proxy).

DNS Servers

- Purpose: DNS (Domain Name System) servers resolve domain names to IP addresses, facilitating the translation of human-readable domain names to machine-readable IP addresses.
- Examples: BIND, Microsoft DNS, Unbound.

Identity Servers

- Purpose: Identity servers manage user authentication and authorization, providing a secure way to access resources.
- Examples: Keycloak, Okta, Microsoft Active Directory.



Collaboration Servers

- Purpose: Collaboration servers facilitate communication and collaboration among users, offering services like chat, video conferencing, and document sharing.
- Examples: Microsoft Teams, Slack, Mattermost