

# Peer-to-Peer (P2P) Architecture

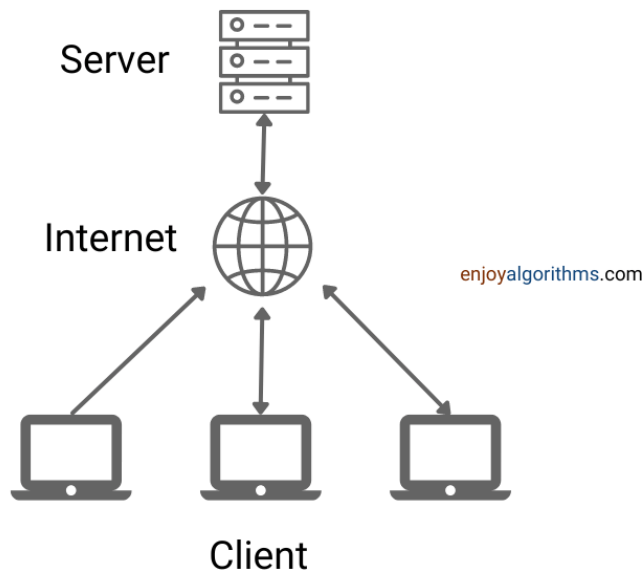
## What is Peer-to-Peer Architecture?

Peer-to-peer (P2P) architecture is a distributed system in which each node acts as both a client and a server so that nodes can share the workload or tasks among themselves.

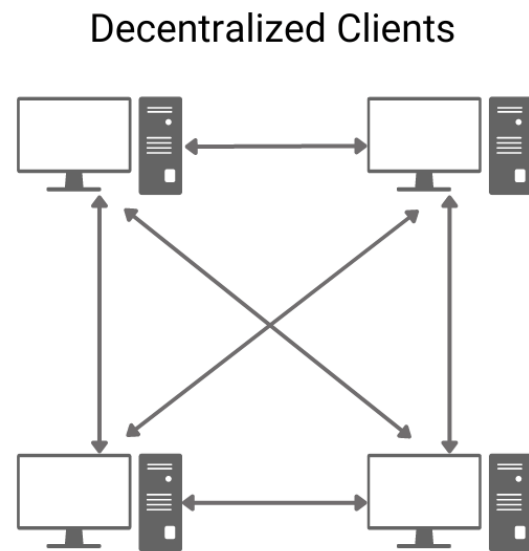
- Each node (or peer) has a similar functional capability and holds some parts of the system resources.
- Some peers may have more resources and may be able to contribute more to the network, while others may not contribute any resources and only consume from the network.
- Peers are visible to each other and can connect with as many other peers as they wish. They are free to join or leave the network at any time and can transmit and receive data from other peers simultaneously.
- Tasks are distributed among all the peers rather than being handled by a single server. This makes P2P architecture more resilient because there is no single point of failure. In some cases, one can use a central tracking server to manage the network and help the peers find each other, but it should not have control

[Courses](#) ▼[Blogs](#) ▼[Tags](#)[Reviews](#)[Stories](#)[Contact Us](#)

...essentially, due to the decentralized nature and the ability to share resources among multiple nodes.



Client Server Architecture



Peer to Peer Architecture

## How does P2P network work?

- Peer-to-peer networks do not have a dedicated server for user authentication. So each computer is responsible for its protection and may require a separate user account for each machine that a user wants to access. Users save files on their machines and are responsible for backing them properly.
- P2P architecture is most effective when there are many active peers in the network. If there are more active peers, new peers can easily find other peers to connect to and ensure that there are enough remaining peers to take up the slack if some peers leave the network. When there are only a few active peers, there are fewer resources available overall.
- In a P2P file-sharing, the more common a file is, the faster it can be downloaded because many peers are sharing it. To make P2P more efficient, one can divide the workload into small pieces that can be reassembled later. This will help many peers to work on the same task simultaneously and reduce the amount of work that each peer has to do.
- Security is a concern in P2P networks, as access to data and resources is managed on each machine. Each computer can allow or deny access to other computers, so users must be careful about what they share. For example, a user might share a folder containing sensitive payroll information. Because there is no central administrator, users must manage access to their files and resources. So

P2P networks are often used in small deployments or situations where security is not a concern.

## Applications of P2P architecture

- Used for file sharing: Users can share files directly with each other without the need for a central server.
- Used for instant messaging or real-time communication between users.
- Used for voice communication, such as through VoIP (Voice over Internet Protocol) technology.

## Use cases of P2P architecture

- **Napster:** Napster was a file-sharing service that was shut down in 2001 due to copyright issues. It used a central tracking server to facilitate file sharing among users.
- **BitTorrent:** BitTorrent is a P2P file-sharing protocol that allows users to share large files directly with each other without the need for a central server.
- **Skype:** Skype originally used a hybrid P2P protocol for voice and video communication, but after being acquired by Microsoft, it now uses a client-server model.
- **Bitcoin:** Bitcoin is a P2P cryptocurrency that operates without a central monetary authority. Transactions are verified and recorded on a decentralized network of computers, rather than being centrally managed.

## Types of peer-to-peer network

There are three different types of P2P networks.

- Unstructured P2P networks
- Structured P2P networks
- Hybrid P2P networks

### Unstructured (or Decentralized) P2P networks

In unstructured peer-to-peer networks, nodes are not arranged in any particular order i.e. node-to-node communication is random. Such type of P2P networks is well-suited for high-activity use cases like social platforms where users may regularly join or leave the network.

- No central control and nodes communicate directly.
- Resilience with no single point of failure.
- One of the use cases is file-sharing applications.

However unstructured P2P networks have a disadvantage. They require a significant amount of CPU and memory to function properly. The hardware must be able to support the maximum number of network transactions so that all nodes can communicate with each other at any given time. This can be a challenge, especially if the network is large or has a high volume of activity.

### **Structured (or Centralized) P2P networks**

Structured peer-to-peer (P2P) networks are the opposite of unstructured P2P networks i.e. nodes have a way to interact with each other in a more organized manner. So one can achieve this through a well-organized architecture that allows users to find and use files more efficiently rather than searching randomly.

While structured P2P networks are more effective, they have some centralization due to their organized architecture. This means that they may be more expensive to maintain and set up than unstructured P2P networks. However, structured P2P networks are more stable than unstructured P2P networks.

- Central server manages communication between nodes.
- Greater control and security.
- Efficient management of tasks.
- Can be used in online gaming, chat rooms, etc.

### **Hybrid P2P networks**

Hybrid peer-to-peer (P2P) networks combine structured and unstructured P2P networks.

- There can be some central servers with P2P capabilities like indexing servers, super peers, or trackers. Such centralized components can help us achieve higher performance and reliability by optimizing resource allocation, improving search performance, and coordinating communication.
- This can also use the core principle of P2P networking by decentralizing components for certain functions. For example, data storage and sharing.
- Can be more scalable than purely centralized systems.
- Depending on the specific requirements, one can use different degrees of centralization. So it provides flexibility by creating a balance between the benefits of centralized and decentralized architectures.

## **Advantages of peer-to-peer networks**

- P2P networks are resistant to single points of failure. Unless the network is very small, it is unlikely that the failure of a single peer will have a significant impact on the overall network. Additionally, changes in the number of peers in the network do not affect its stability.
- P2P networks are decentralized and distributed, which means that as more people interact with your data, more nodes are added to the network. This allows for scaling of bandwidth and storage.
- All nodes have equal privileges in a P2P network, ensuring minimal corporate control or intervention. This means that you have control over your data.
- P2P networks can be more cost-effective due to the absence of a central server that needs to be maintained and paid for (apart from monitoring servers).
- P2P networks can handle large simultaneous connections without difficulty due to their decentralized structure. This also makes them more resistant to attacks since no central server could be targeted.

## **Disadvantages of peer to peer networks**

- Peer-to-peer networks have some security risks. For example, if a peer becomes infected with a virus and uploads a file containing the virus, the virus can easily spread to other peers in the network.

- Another concern is that it can be difficult to ensure that all peers have the correct permissions to access the network, particularly if there are a large number of peers.
- Some users of P2P networks take advantage of resources shared by other nodes without contributing anything themselves. These users are known as "leachers." While the decentralized nature of P2P networks can make them difficult to shut down, this can also be a drawback if the network is used for illegal or unethical activities.
- If all the nodes in the P2P network are offline, the data becomes unavailable.

Thanks to Navtosh Kumar for his contribution in creating the first version of this content. If you have any queries/doubts/feedback, please write us at [contact@enjoyalgorithms.com](mailto:contact@enjoyalgorithms.com). Enjoy system design, Enjoy algorithms!

[Prev Chapter](#)[Next Chapter](#)

Author: [Shubham Gautam](#)

Reviewer: [Keshav Nandan](#)

Share on: [!\[\]\(5a132f13505a6571904d622757b7a8f0\_img.jpg\)](#) [!\[\]\(0f17417dd77a61b2fdbff69a33adf9f2\_img.jpg\)](#) [!\[\]\(36c143dff828c7ad385930a18d411514\_img.jpg\)](#) [!\[\]\(531448a1ec87799b9d407c1deac59d92\_img.jpg\)](#)

Find us on: [!\[\]\(10f8862fc183b400327470ea85afe9ae\_img.jpg\)](#) [!\[\]\(4ba8d838a2aa5445d51c9dee78fcb0cc\_img.jpg\)](#) [!\[\]\(4fe307d00a844a23eb14d503e73187bd\_img.jpg\)](#) [!\[\]\(eee55ba6d819a850185cead073659564\_img.jpg\)](#)

[system-design-interview](#)[system-design-concepts](#)

## Share Your Insights

## More from EnjoyAlgorithms

## Difference Between SQL and NoSQL Databases

SQL and NoSQL databases have their own advantages and disadvantages. SQL databases are the best choice for consistency, complex query, and ability to handle structured data, while NoSQL databases are the best choice for their scalability, flexibility, and ability to handle unstructured data. So the decision between SQL and NoSQL will depend on the project requirements.

[Read More](#)

## Pastebin System Design

Pastebin is a online content hosting service where users can store and share content in the form of text or images over the internet by generating a unique URL so that anyone can access the content via that URL. Users can also update the content if they are logged in. This blog will focus on pastebin system design and discussion around various components.

[Read More](#)

## Design Twitter: System Design Interview Question

Twitter is a social media platform where users can post and interact with tweets. Users can also subscribe to feeds of other users and receive tweet notifications from those they follow. Tweets are almost 140–280 characters communication. This blog will focus on system design of Twitter and discussion around various components.

[Read More](#)

## WebSockets in System Design

Web Socket is a communications protocol which provides full-duplex communication channels over a single TCP connection. This is also a vital component behind multiplayer games or applications that rely on real-time data transfer. In this blog, we have explained: 1) What is WebSocket in system design? 2) How does it work?

[Read More](#)

## Introduction to Hadoop Framework for Big Data

Hadoop is an open-source framework that addresses the analytical and operational needs of Big Data by overcoming the limitations of traditional data analysis methods. With support for highly scalable and fault-tolerant distributed file systems, it allows for parallel processing. It comprises four main components - HDFS, YARN, MapReduce, and Hadoop Common.

[Read More](#)

## Database Partitioning in System Design

Data partitioning is a technique of dividing data into independent components. It is a way of splitting data into smaller pieces so that data can be efficiently accessed and managed. Database partitioning is the backbone of modern system design, which helps to improve scalability, manageability, and availability.

[Read More](#)

## Self-paced Courses and Blogs

[Coding Interview](#)[Machine Learning](#)[System Design](#)[OOP Concepts](#)

[DSA Course](#)[ML Course](#)[SD Course](#)[OOP Course](#)[DSA Blogs](#)[ML Blogs](#)[SD Blogs](#)[OOP Blogs](#)

## Our Newsletter

Subscribe to get well designed content on data structure and algorithms, machine learning, system design, object oriented programming and math.

[✉ Subscribe](#)[Courses](#)[Latest Blogs](#)[Shubham Blogs](#)[Ravish Blogs](#)[Popular Tags](#)[EnjoyMathematics](#)[About Us](#)[Contact Us](#)[Terms and  
Conditions](#)[Refund Policy](#)[Privacy Policy](#)[Cookie Policy](#)

©2023 Code Algorithms Pvt. Ltd.  
All rights reserved.