



Peer-to-Peer Architecture – Part 2

This lesson contains the second part of the peer-to-peer architecture discussion. We will continue where we left off in the previous lesson.

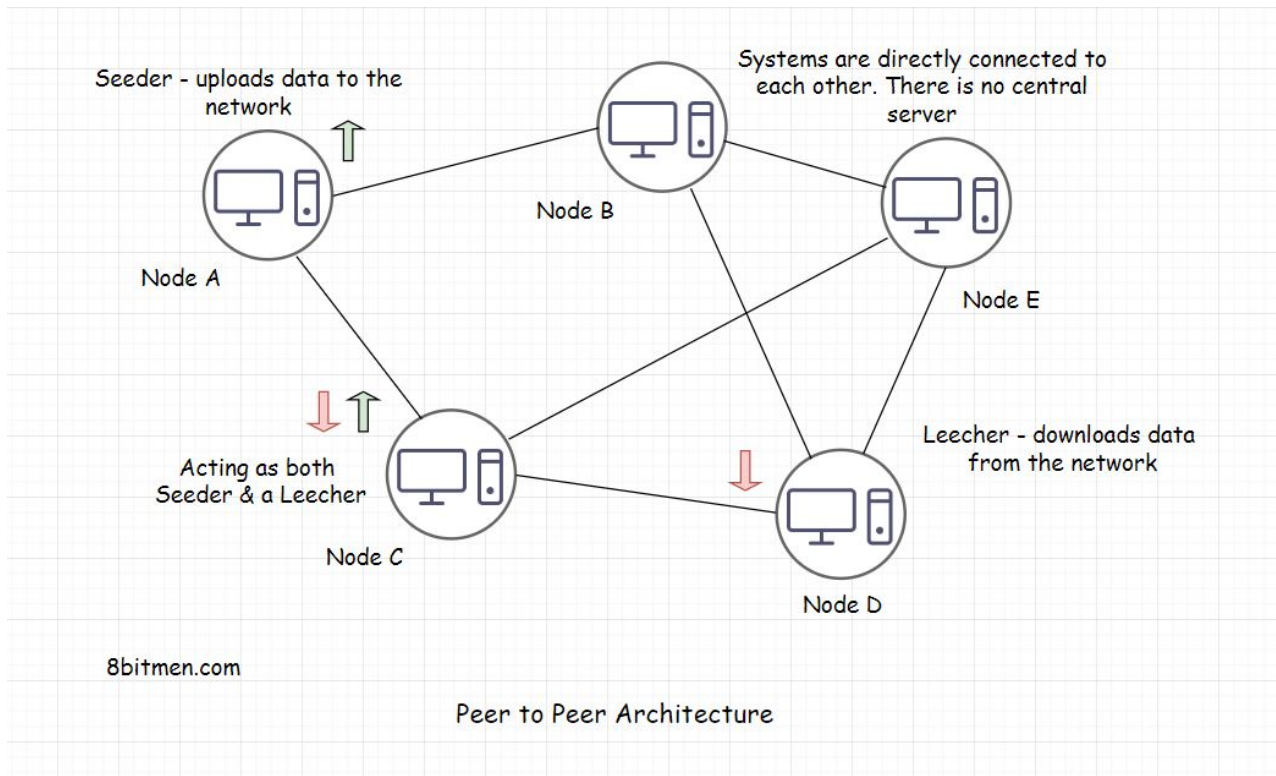
We'll cover the following



- What is a peer-to-peer architecture? How does it work?
- Types of P2P networks
 - Unstructured network
 - Structured network
 - Hybrid model

What is a peer-to-peer architecture? How does it work?#

A *P2P* architecture is designed around several *nodes* in the network taking equal turns acting as both the *client* and the *server*.



The data is exchanged over *TCP IP* just like it happens over the *HTTP* protocol in a *client-server* model. The *P2P* design has an overlay network over *TCP IP*, which enables the users to connect directly. It takes care of all the complexities and the heavy lifting. *Nodes/peers* are indexed and discoverable in this overlay network.

A large file is transferred between the nodes by being divided into chunks of equal size in a non-sequential order.

Say a system hosts a large file of 75 *gigabytes*. Other *nodes* in the network in need of the file locate the system containing the file. Then, they download the file in chunks, re-hosting the downloaded chunk simultaneously, making it more available to the other users. This approach is known as a *segmented P2P file transfer*.

Based on how these peers are linked with each other in the network, the networks are classified into a *structured*, *unstructured*, or a *hybrid* model.

Types of P2P networks#



Unstructured network#

In an *unstructured* network *nodes/peers* keep connecting with each other randomly. So, there is no structure, no rule. Just simply connect and grow the network.

In this architectural design, there is no indexing of the nodes. To search the data, we have to scan through each and every node in the network. This is $O(n)$ in complexity where n is the number of nodes in the network. This is very *resource-intensive*.

Think of it in this way. There are a billion systems connected in the network. Then, there is a file stored in just one system in the network. In an *unstructured network*, we have to run a search through each system in the network to find the file.

So, let's say the search for a file in a system needs 1 second. The search through the entire network would require one billion seconds.

Some of the unstructured network's protocols are *Gossip*, *Kazaa*, and *Gnutella*.

Structured network#

In contrast to an *unstructured network*, a *structured P2P network* holds the proper node indexing, or the topology. This makes it easier to search for specific data.

This kind of network implements a *distributed hash table* to index the nodes. This index is just like the index of a book where we check to find a piece of particular information in the book rather than searching through every page.

BitTorrent is an example of this type of network.



Hybrid model#

The majority of the blockchain startups have a *hybrid model*. A *hybrid model* means cherry-picking the good stuff from all the models like *P2P*, *client-server* etc. It is a network, involving both a *peer to peer* and a *client-server* model.

As we know, in a *P2P* network, one single entity doesn't have all the control. So, to establish control, we need to set up our own server. For that, we need a *client-server* model.

A *P2P network* offers more availability. To take down a blockchain network you have to literally take down all the network's nodes across the globe. A *P2P* application can scale to the moon without putting the load on a single entity or the node. In an ideal environment, all the *nodes* in the network equally share the bandwidth and the storage space. The system scales automatically as new users use the app.

Nodes get added as more and more people interact with your data. There are zero-data storage and bandwidth costs, and you don't have to shell out money to buy third-party servers to store your data. There is no third-party intervention, so data is secure. Share stuff only with friends you intend to share with.

The cult of the *decentralized web* is gaining ground in the present times. I can't deny that this is a disruptive tech with immense potential.

Blockchain, or *Cryptocurrency*, is one example of this. It has taken the financial sector, in particular by storm.

There are numerous *P2P* applications available on the web for instance:

1. *Tradepal*

2. *Peer-to-peer digital cryptocurrencies like Bitcoin and Deercoin*

2. Peer-to-peer digital cryptocurrencies like *Bitcoin* and *Ethereum*.



3. *GitTorrent* (a decentralized *GitHub* which uses *BitTorrent* and *Bitcoin*).

4. *Twister* (a decentralized microblogging service, which uses *WebTorrent* for media attachments).

5. *Diaspora* (a decentralized social network implementing the *federated architecture*).

Federated architecture is an extension of the decentralized architecture, used in *decentralized social networks*. We are going to discuss this next.

[← Back](#)[Next →](#)[Peer-to-Peer Architecture – Part 1](#)[Decentralized Social Networks](#)[Mark as Completed](#)[Report an Issue](#)