

## CHAPTER 1

# Decentralization and Web3

In this chapter, we are going to examine the evolution of Web 3.0 starting from Web 1.0 and 2.0.

## 1.1 Web 1.0

In Web 1.0, websites delivered static content (rather than dynamic content) written in hypertext markup language (HTML). The data and content were sent by means of a static file system as opposed to a database, and the web pages had only a minimal amount of interactive information.

The following is a list of the principal technologies that were included in Web 1.0:

- HTML (HyperText Markup Language)
- URL encoded in the HyperText Transfer Protocol, or HTTP (Uniform Resource Locator)

## 1.2 Web 2.0

Since in Web 1.0 most content was static, it gave rise to Web 2.0.

The vast majority of us have only experienced the World Wide Web in its most recent iteration, sometimes referred to as Web 2.0, the interactive read-write web, and the social web. Participating in the creative process of Web 2.0 does not require you to have any prior experience as a developer. A great number of apps are designed in such a way that anyone can make their own version of the program.

You are capable of coming up with ideas and communicating those ideas to people around the globe. You are able to post a video to Web 2.0, where it will be available for millions of users to watch, interact with, and comment on. Web 2.0 applications consist of social media sites, such as YouTube, Facebook, Flickr, Instagram, and Twitter, among others. Think about how popular these sites were back when they first launched, and then compare that to how popular they are now.

Web 2.0 allowed applications to scale, but also gave rise to centralized platforms that took control of all user data. This gave them a chance and opportunity to do data mining, which can be put to both positive and negative uses.

Since users were increasingly losing control to these centralized platforms, it led to the emergence of Web 3.0

## **1.3 Web 3.0**

Web 2.0 design inherently depended on centralized systems. Those systems might have been distributed but were not decentralized. This meant that control of those systems remained with individuals or a group of individuals. This led to huge issues as privacy and data usage concerns started to emerge.

This led to the evolution of Web 3.0, which keeps decentralization as its first requirement. This means applications for Web3 will be deployed in a decentralized way and data storage for its apps will also be decentralized. Therefore, it's important to have an idea of what decentralization is before understanding the Web3 landscape.

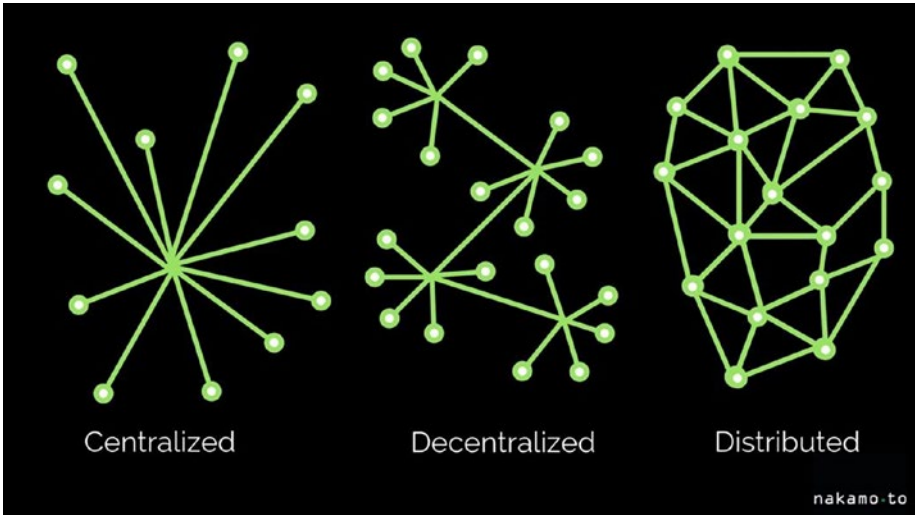
## 1.3.1 Introduction to Decentralization

“Unthinking respect for the authority is the greatest enemy of truth.” This famous quote by Albert Einstein goes a long way in making us think about whether there should be systems that are controlled by an authority, thereby asking us to have complete faith and trust in the authority, or there should be systems that are decentralized, with consensus as the root of decision making. Central systems, as their name suggests, can be controlled by the whims of a central authority, whereas a decentralized system will favor more inclusive decision making, thereby reducing the chance of corruption within that system.

In a decentralized setup, there is no owner or master. There are only peers participating in the network. These nodes exchange messages between them to form consensus as and when needed. The protocols governing these networks make sure that the overall system of nodes is in a consistent state. Every node in the network has an up-to-date copy of all the data that was recorded. Decentralized networks can also distribute data to validate particular private information without having to hand over that information to a third party. This validation is made possible by the fact that data does not have to be transferred. The validation of data is accomplished by the utilization of a consensus-based technique, wherein the nodes on the network agree to a certain state of the overall system at a certain point in time.

Each participant node in a decentralized network functions independently of the other nodes in the network. Decentralized nodes interact with one another through the use of common standards, but they keep their independence and are in charge of their own privacy management, rather than adhering to the directives of a centralized authority. This not only helps to maintain the network’s safety but also ensures that it is governed in a democratic manner.

Figure 1-1 shows how nodes connect and communicate in centralized, decentralized, and distributed networks.



**Figure 1-1.** *Difference between centralized, decentralized, and distributed networks*

The terms “centralization” and “decentralization” both refer to levels of control. Control is exercised by just one organization or person in a centralized system (a person or an enterprise, for example). There is not a single entity that controls the operation of a decentralized system. Instead, the control is divided among a number of different autonomous entities.

Differences in location are meant to be understood as distribution.

In a system that is not distributed, also known as being co-located, all of the components of the system are situated in the same physical location. A distributed system is one in which different components can be found in different physical locations.

### 1.3.2 Different Topologies for Networks

Networks can be configured in different topological settings. A few of them are explained in the following sections.

### 1.3.2.1 Centralized and Non-distributed

In the event that you are putting together a stand-alone application that will run on your Windows computer, this configuration is not distributed and also functions as a centralized system. Your application is under the control of one party (the application vendor), and an additional party, such as Microsoft, is in charge of your operating system (centralized). The application and the operating system are both stored locally on your personal computer, making them both non-distributable components.

### 1.3.2.2 Centralized but Distributed

Imagine that you are working on a cloud application that is hosted on Amazon Web Services and runs on virtual machines hosted by Amazon Web Services. This configuration is both centralized and distributed at the same time. Your application and operating system are both under the control of AWS (centralized). Cloud storage serves as the host for both the application and the operating system, which may be partitioned across multiple locations.

## 1.3.3 Decentralized Systems

By its very nature, a decentralized system will be distributed. Peer-to-peer software can be thought of as an example of a decentralized system, which is defined as “a system that requires multiple parties to make their own independent decisions.”

It is impossible to have a single, centralized authority in a decentralized system that is responsible for making decisions on behalf of all the different parties. Instead, each party, also known as a peer, is responsible for making locally autonomous decisions in the direction of achieving its own individual goals, which may or may not be in conflict with the goals pursued by other peers. Peers engage in one-on-one communication with

one another, during which they may exchange information or perform a service for other peers. When it comes to decentralized systems, an open system is one in which the participation of peers is not restricted in any way. At any point, a peer may join or leave the system at their discretion.

Before information is added to a blockchain (which is an example of a decentralized system), the nodes participating in the network must use something called a consensus mechanism to reach a decision on whether or not it is accurate. Once the block has been updated with the data, it is sent to all of the nodes in the network. This makes it extremely difficult to alter information that has already been added to the ledger. Mutation of state on a decentralized network requires a majority of nodes to reflect the updated state, which is a very difficult thing to achieve, and therefore the decentralized network of nodes is immutable. This is in contrast to changing the data in a single database that is centralized. If rule of immutability is broken, the validating nodes will disregard such information.

The process of dispersing functions and power away from a central location or authority is known as decentralization. It is difficult, if not impossible, to identify a specific center in a decentralized architecture. The World Wide Web was created as a decentralized platform. Decentralized architectures and systems are demonstrated by blockchain technologies such as Bitcoin and Ethereum.

As the technological landscape changes, decentralized structures emerge. Decentralization has the power to change a lot of things, from governance and industry to justice systems.

Now, with this basic knowledge of decentralization, we can start to understand the Web 3.0 landscape.

Web 3.0 apps are built on blockchains, which are decentralized networks of numerous peer-to-peer nodes. Blockchains are used to record transactions that take place between users. In the context of the Web 3.0 ecosystem, these applications are referred to as decentralized apps (DApps), which is a term that is frequently used. The participants

in the network, known as developers, are paid for their efforts to provide services of the highest possible quality in order to maintain a decentralized network that is both robust and safe.

At a high level, the Web 3.0 architecture comprises four major parts, as follows:

1. **Blockchain** – A peer-to-peer network of nodes is responsible for the maintenance of these global state machines. Access to the state machine, as well as writing to it, is open to anybody on the globe. In essence, rather than being owned by a single company, it is held collectively by all of the participants in the network. Users are able to upload new data to the blockchain, but they are unable to edit the data that has already been added. There are many types of blockchain available, like Ethereum, Solana, Polkadot, etc.
2. **Smart Contracts** – The Ethereum blockchain can host something called a smart contract, which is a computer program that runs on the blockchain. These are written by app developers using high-level programming languages such as Solidity or Vyper to specify the logic that lies beneath the state transitions. We will discuss Solidity in Chapter 3.
3. **Ethereum Virtual Machine (EVM)** – It's a virtual machine that is tasked with putting into action the logic that is outlined in smart contracts. It is in charge of managing the state machine's transitions between states. It generally operates on the bytecode, which is generated by taking the Solidity source code and compiling it via the Solidity compiler.

4. Front End/UI – The user interface (UI) logic is defined by the front end, just like it is in any other application. However, it does communicate with smart contracts, which are programs that describe how an application works.

Before we can develop a simple Web3 app, we need to be conversant with certain technologies and environments. We will start with the Remix integrated development environment (IDE), which we will use as an environment in this chapter for the development and deployment of our Web3 app based on smart contracts.

### 1.3.4 Web3 Case Study

Every one of us as professionals has heard about LinkedIn, which is a social network for professionals. It is controlled by LinkedIn, and its bosses can decide the rules of the platform. They can also choose whom to censor if they want to. But we don't need to worry. There is a Web3 alternative to LinkedIn that goes by the name of Entre. It can be accessed at <https://joinentre.com/>.

It's a decentralized professional network built using Web3 technologies. It currently has around 50,000 members and is growing. It provides posting for jobs, calendar for events, and many other features. Since this is a decentralized platform, the rules are hard to change by a central authority.

Apart from this, there are other Web3 platforms, like Deso as an alternate to Twitter and Odysee as an alternate to YouTube. I can say that we are in very exciting times as far as the shape of the technology is concerned.



## 1.4 Summary

In this chapter, we briefly touched upon the notion of Web 3.0 and decentralization and its forms. In next chapter, we will introduce the reader to the blockchain ecosystem and its various types. We will also study the functioning of some of the existing blockchain networks, like Bitcoin and Ethereum.