The Web3 development stack

Written byKingsley Arinze

Before you set out to build an application, you need to first decide on the technology stack to use. The term “technology stack” refers to the tools, libraries, and frameworks needed to build and run a software application. A simple application would consist of three layers:

1. The Data layer that handles information storage for easy retrieval and computation.
2. The Logic layer that handles the business logic. All the necessary computation happens here.
3. The Presentation Layer that is responsible for what the user sees. It presents everything happening under the hood in a user-friendly way.

To successfully build out each layer, you need to use specific tools, libraries, and frameworks. These tools, libraries, and frameworks are what forms your development stack for that software application.

In the traditional Web2 world, you hear references such as MEAN stack, LAMP stack, MAMP stack. These are all acronyms for some of the tools, libraries, and frameworks that power Web2 applications.

Web3 is an emergent iteration of the World Wide Web. As with every technology in its nascent stage, most people struggle to understand how the different parts come together. This blog defines the Web3 stack, but before we jump right in, let's review the evolution of the Web.

Web 1: Read-Only¶

Web1 refers to the World Wide Web era where users (mostly companies) created content in the form of static websites built withSSI orCGI and hosted on ISP servers.

Web1 was aread-only web and lasted between 1990 and 2004, with most users consuming content created by content creators.

The Web1 stack had two defining layers:

1. File System
2. for file storage (Data Layer)
3. HTML and CSS for presentation. (Presentation Layer)

Since Web1 was primarily static and presentational, there wasn't a layer that handled business logic.

Web 2: Read-Write¶

The Web2 is aread-write web where both companies and users can be content creators. It allowed everyday people to contribute to the web for the first time and was ushered in by the Facebooks and the Googles of the world -the tech giants .

While there wasn't any change in the nature and structure of Web1, there was a significant change in how people started to use the web. The introduction of Javascript libraries such as JQuery saw the web become more interactive and user-friendly.

Building on Web2 requires adding more sophisticated tools to the Web1 stack like dynamic programming languages (JavaScript, PHP, Java, Python) that could dynamically serve content from files and database systems.

The Web2 stack can be said to include the following:

1. Databases: Relational databases(Postgres, Mysql), NoSQL databases(MongoDB), and key-value databases(Redis), amongst others, form the data layer for Web2 applications.
2. Dynamic Programming Languages handle the business logic in Web2 applications. It saw the introduction of frameworks like Laravel (PHP), Django(Python), and libraries like Express(Node.js/Javascript) that streamlined the software development process.
3. HTML and CSS remain the go-to for building user interfaces and presenting data in Web2. However, with Javascript libraries like JQuery, React.js, and Angular, the data presentation layer in Web2 is more interactive and user-friendly than that of its predecessor.

Web3: Read-Write-Own¶

Thus far, we've experienced Web1 where companies created content and, as a result, earned money. We've also seen (and are still seeing) Web2 where users generate content but only companies and platform owners, through user-generated content, make money. This new web iteration known asWeb3 takes the web a step further by allowing content creators toown not only their content but their identity and assets.

Web3 is a decentralized, open, and censorship-resistant web, powered by blockchains like Ethereum, Solana, and hundreds of others. It is a trustless and self-governing web, where companies do not have as much power as they do in Web2 due to the openness and ownership of user data. Instead, it aims to return power to those responsible for keeping the network alive.

To explain what it means for users to own their data in Web3, let's examine the current situation on Web2 today. Assuming a user isn't impressed with the practices of a company they use regularly and decides to move to a different platform, this user would have to start fresh as all their existing data would live in a centralized database owned by a company like Google, Facebook, Amazon, and others. What makes Web3 different is that anyone with an Internet connection can access the blockchain which gives users more freedom and control over their data.

Web3 also introduced the concept of Decentralized Autonomous Organizations (DAOs), which is a new way of running companies through communities. In a traditional Web2 company, decisions regarding its direction are determined by investors and stakeholders, leaving out users directly affected by these decisions. With DAOs, users in the form of community members are responsible for deciding the direction of these companies through rules and regulations written in code and enforced by blockchain technologies. These programmable rules and regulations are known as smart contracts.

Smart contracts are computer programs that digitize agreements and automatically execute when the terms of said agreements are met. They are computer code that lives and runs on Ethereum and other blockchains.

The Web3 Stack¶

The Web3 stack is a collection of tools and technologies that are different from the Web2 stack. Web3 replaces centralized technologies like databases with open and decentralized technologies like blockchains. The only similarity between both stacks is how Javascript libraries remain the go-to for building user interfaces, and even at that, how we fetch data from the backend is different in that Web2 applications use libraries like Axios and Fetch to make HTTP calls to backend servers. In contrast, web3 applications use libraries like Ether.js and Web3.js to make RPC calls to blockchains.

Web3 also introduces new primitives like native payment support that completely removes the need for intermediaries and large banks.

We can breakdown the Web3 stack into the following layers:

1. The data layer powered by decentralized blockchain technologies
2. File storage layer powered by distributed peer-2-peer networks
3. API/Data Access Layer
4. Presentation Layer
5. Identity Layer

The data layer¶

This is where information is stored and is the major difference between the Web2 and the Web3 stack.

Whereas Web2 applications store data on central databases, owned and maintained by companies who restrict access, Web3 applications rely on open, decentralized, and permissionlessstate machines for data storage, where anyone with the requisite skills and enough curiosity can access data.

Blockchain technologies primarily power this layer and Ethereum is at the forefront with the majority of Web3 developers choosing to build their dapps there. Ethereum has a virtual environment known as theEthereum Virtual Machine or EVMthat stores user data (like accounts and accounts balance) and computes the next state of the network. This virtual environment is also responsible for storing and executingsmart contract code mostly written in the Solidity and Viper programming languages.

Due to the popularity of the EVM and the Ethereum blockchain, we've seen the emergence of other blockchains, sidechains, and Layer 2 solutions fully compatible with the EVM, allowing applications built on Ethereum to run on these platforms without needing any modification. Examples of these platforms are:

- The Arbitrium layer two solution
- The Optimism layer two solution
- The Polygon sidechain
- The Avalanche blockchain
- The Fantom blockchain

There are other blockchains that operate differently from Ethereum and are not EVM-compatible but are home to decentralized applications too likeSolana , NEAR, Avalanche, and others Although to build dapps on Solana and NEAR, you must understand the Rust programming language.

File storage Layer¶

Since users' data storage happens on decentralized blockchains, it makes sense that files like images, videos, HTML, and CSS files are also stored in decentralized networks so no one organization has full access and control over them. Unfortunately, storing data like the ones mentioned above on blockchains like Ethereum (and other EVM compatible ones) or Solana can be costly.

When choosing a data storage mechanism for a dapps, one has to consider the following:

- Persistence mechanism and incentive structure
- Data retention enforcement
- Decentrality
- Consensus

Fortunately, there are several options to choose from likeArweave who offer permanent file storage by incentivizing node operators and miners who in turn maintain storage of these files for extended periods.

Contract-based platforms likeFilecoin andSkynet do not store files forever but instead, keep files for a period specified by a contract.InterPlanetary File System or IPFS , another distributed system for storing and accessing files, doesn't have any built-in incentive mechanism for miners but can be used together with other contract-based solutions.

API/Data Access Layer¶

Another essential layer of the Web3 stack is the API or Data access layer. This comprises platforms that enable easy access to data that lives on the blockchain and files on distributed file storage networks.

Due to the current state of popular blockchain networks like Ethereum, running a self-hosted node to access blockchain data can be time-consuming. Therefore, maintaining these nodes does not contribute directly to productive hours, especially during development.

This is why platforms likeInfura exist. Infura exposes a set of APIs that allow Web3 developers to connect their dapps to the Ethereum network with zero stress, letting them focus on building out application features rather than maintaining nodes. It also provides an API for connecting to IPFS for distributed files storage and retrieval.

Alchemy is another example of API platforms supporting popular blockchains like Ethereum, Flow, and PolygonQuicknode , on the other hand, provides Dapps developers with access to blockchain data via APIs and dedicated node instances.

Presentation Layer¶

The presentation layer for a dapp is not too different from that of a Web2 app. We use the same set of libraries to create user interfaces. For example, due to its popularity and thriving ecosystem, theReact.js library remains the preferred choice for building out the frontend of Web3 applications. Other libraries such as JQuery, Angular and VueJS can also be used.

The major difference between the Web3. frontend stack and that of Web2. is how we fetch data from external APIs. By comparison, a typical Web2 application uses HTTP libraries like Axios and Fetch to query API endpoints or clients like Apollo for GraphQL endpoints; in Web3, we use libraries likeEther.js orWeb3.js to make Remote Procedure Calls (RPC) to blockchain nodes.

Identity Layer¶

Unlike in Web2, where internet users are identified by their unique username/email and password, Web3 implements an entirely different strategy for identity management powered by blockchain wallet likeMetaMask .

In Web3, the typical user experience involves connecting your wallet to a dapp you want to use and granting this wallet the power to sign transactions on your behalf. There are two popular ways to connect a wallet to a dapp:

1. By accessing the provider object exposed by these wallets on the browser window. This primarily applies to wallets in the form of browser extensions.
2. By scanning a barcode exposed by the dapp using tools likeWalletConnect
3. (for Ethereum). Wallets in the form of mobile applications mostly use this method.

Libraries likeWeb3Modal andWeb3React exists to help developers connect to multiple wallet providers with the same code.

Conclusion¶

Web3 enforces the concept of decentralization by replacing centralized technologies with blockchains and protocols. It also introduces a new way of giving users ownership of their data and identity through private-key cryptography. What will define Web3 moving forward is the ease at which dapps can be built. This is why defining the stack is critical. At Truffle we aim to be the number one choice for development.