

A decentralized application (dapp) is an application built on a decentralized network that combines a [smart contract](#) and a frontend user interface. On Ethereum, smart contracts are accessible and transparent – like open APIs – so your dapp can even include a smart contract that someone else has written.

Prerequisites {#prerequisites}

Before learning about dapps, you should cover the [blockchain basics](#) and read about the Ethereum network and how it's decentralized.

Definition of a dapp {#definition-of-a-dapp}

A dapp has its backend code running on a decentralized peer-to-peer network. Contrast this with an app where the backend code is running on centralized servers.

A dapp can have frontend code and user interfaces written in any language (just like an app) to make calls to its backend. Furthermore, its frontend can get hosted on decentralized storage such as [IPFS](#).

- **Decentralized** - dapps operate on Ethereum, an open public decentralized platform where no one person or group has control
- **Deterministic** - dapps perform the same function irrespective of the environment in which they get executed
- **Turing complete** - dapps can perform any action given the required resources
- **Isolated** - dapps are executed in a virtual environment known as Ethereum Virtual Machine so that if the smart contract has a bug, it won't hamper the normal functioning of the blockchain network

On smart contracts {#on-smart-contracts}

To introduce dapps, we need to introduce smart contracts – a dapp's backend for lack of a better term. For a detailed overview, head to our section on [smart contracts](#).

A smart contract is code that lives on the Ethereum blockchain and runs exactly as programmed. Once smart contracts are deployed on the network you can't change them. Dapps can be decentralized because they are controlled by the logic written into the contract, not an individual or company. This also means you need to design your contracts very carefully and test them thoroughly.

Benefits of dapp development {#benefits-of-dapp-development}

- **Zero downtime** – Once the smart contract is deployed on the blockchain, the network as a whole will always be able to serve clients looking to interact with the contract. Malicious actors, therefore, cannot launch denial-of-service attacks targeted towards individual dapps.
- **Privacy** – You don't need to provide real-world identity to deploy or interact with a dapp.
- **Resistance to censorship** – No single entity on the network can block users from submitting transactions, deploying dapps, or reading data from the blockchain.
- **Complete data integrity** – Data stored on the blockchain is immutable and indisputable, thanks to cryptographic primitives. Malicious actors cannot forge transactions or other data that has already been made public.
- **Trustless computation/verifiable behavior** – Smart contracts can be analyzed and are guaranteed to execute in predictable ways, without the need to trust a central authority. This is not true in traditional models; for example, when we use online banking systems, we must trust that financial institutions will not misuse our financial data, tamper with records, or get hacked.

Drawbacks of dapp development {#drawbacks-of-dapp-development}

- **Maintenance** – Dapps can be harder to maintain because the code and data published to the blockchain are harder to

modify. It's hard for developers to make updates to their dapps (or the underlying data stored by a dapp) once they are deployed, even if bugs or security risks are identified in an old version.

- **Performance overhead** – There is a huge performance overhead, and scaling is really hard. To achieve the level of security, integrity, transparency, and reliability that Ethereum aspires to, every node runs and stores every transaction. On top of this, proof-of-stake consensus takes time as well.
- **Network congestion** – When one dapp uses too many computational resources, the entire network gets backed up. Currently, the network can only process about 10-15 transactions per second; if transactions are being sent in faster than this, the pool of unconfirmed transactions can quickly balloon.
- **User experience** – It may be harder to engineer user-friendly experiences because the average end-user might find it too difficult to set up a tool stack necessary to interact with the blockchain in a truly secure fashion.
- **Centralization** – User-friendly and developer-friendly solutions built on top of the base layer of Ethereum might end up looking like centralized services anyways. For example, such services may store keys or other sensitive information server-side, serve a frontend using a centralized server, or run important business logic on a centralized server before writing to the blockchain. Centralization eliminates many (if not all) of the advantages of blockchain over the traditional model.

More of a visual learner? {#visual-learner}

Tools for creating dapps {#dapp-tools}

Scaffold-ETH - *Quickly experiment with Solidity using a frontend that adapts to your smart contract.*

- [GitHub](#)
- [Example dapp](#)

Create Eth App - *Create Ethereum-powered apps with one command.*

- [GitHub](#)

One Click Dapp - *FOSS tool for generating dapp frontends from an [ABI](#).*

- [oneclickdapp.com](#)
- [GitHub](#)

Etherflow - *FOSS tool for Ethereum developers to test their node, and compose & debug RPC calls from the browser.*

- [etherflow.quiknode.io](#)
- [GitHub](#)

thirdweb - *SDKs in every language, smart contracts, tools, and infrastructure for web3 development.*

- [Homepage](#)
- [Documentation](#)
- [GitHub](#)

Further reading {#further-reading}

- [Explore dapps](#)
- [The Architecture of a Web 3.0 application](#) - Preethi Kasireddy
- [A 2021 guide to decentralized applications](#) - LimeChain
- [What Are Decentralized Apps?](#) - Gemini
- [Popular dapps](#) - Alchemy

Know of a community resource that helped you? Edit this page and add it!

Related Topics {#related-topics}

- [Introduction to the Ethereum stack](#)
- [Development frameworks](#)