

Web 3.0 Economics

With technological decentralization also came economical decentralization. It stands on 3 pillars - Non-Fungible tokens (NFTs), Fungible tokens (FTs) and Decentralized Finance (DeFi).

Non-Fungible Tokens

At the heart of the new Web 3 economy lies [Non-Fungible token](#) (NFT). In a nutshell, it's a way to represent digital ownership in a decentralized way. From a technical perspective, it's just a piece of data on a blockchain. The simplest case of such data is just a (token_id, account_id) tuple, where token_id uniquely identifies an asset, and account_id identifies an owner. A smart contract that owns this data defines a set of allowed operations - like creation of a new token (minting) or transfer of a token to another account. An exact set of allowed operations is defined in an NFT standard. Different blockchains have different standards, NEAR NFT Standard is available [here](#).

Because NFTs are tied to a specific contract, they mostly make sense only in scope of this contract, and subsequently they are tied to a specific dApp. It's possible to implement transfer of NFTs between contracts, but there's no standard way to do this.

What digital asset is hiding behind a token_id is up to the smart contract to decide. There are few common ways how to handle this:

- Store an asset itself in a smart contract alongside the ownership information. This is the most straightforward way, but often is not feasible since storage cost is quite high and many types of digital assets, especially media, are quite big.
- Store token data off-chain. Such an approach solves storage cost problems, but requires some level of trust to guarantee that data in the off-chain storage won't be changed or removed.
- Store asset's metadata and hash on chain, and an asset itself on some off-chain storage. Storing an asset's hash on a chain guarantees data integrity and immutability. On-chain metadata usually includes basic token information, like title, description and media url. It's required to quickly identify an asset without downloading it from the storage. This is the most popular approach to handle NFT's since it combines the best of 2 previous approaches - token is immutable and storage cost is cheap (exact cost depends on the storage solution, but it usually several orders of magnitude cheaper than an on-chain storage)

Choosing the right off-chain storage also can be a challenge, in general they can be divided into 2 buckets:

- Centralized storages - traditional Web 2 storage solutions, like relational databases or blob storages. While suitable for some applications, this means NFTs can be destroyed if a central server goes offline, so they aren't the most popular in the Web 3 world.
- Decentralized storages. As we already mentioned, BitTorrent protocol is one of the first examples of such decentralized storage solutions, but in recent years more advanced solutions have appeared - like [IPFS](#), [FileCoin](#) and [Arweave](#).
- Such solutions are a preferred method to store digital assets, since they are cheap and decentralized, so no-one can destroy or alter NFT assets.

In addition to the NFT standard, NEAR also provides [its implementation](#), which can be used by Smart Contract developers to implement NFTs in their smart contract. Implementation itself doesn't dictate assets storage model, so it's up to a developer to decide how and where it will be stored.

Fungible Tokens

NFTs changed digital assets ownership model, but by itself they are not enough to build a full digital economy. In the simplest model, NFTs can be sold and bought using main blockchain currency (e.g. NEAR tokens), but this is quite limiting since circulation and price of such tokens is dictated by the blockchain itself. What if, instead of relying on blockchain currency, applications could create their own? For exactly this reason, Fungible Tokens (FT) have been created.

Similarly to NFTs, fungible tokens are also just a piece of data stored in a smart contract, but instead of storing unique token ids, an amount of tokens held by an account is stored.

Smart Contracts can define allowed operations - like transfer or payment using this token [NEAR defines a standard](#) for fungible tokens and provides [a default implementation](#).

Since an application is fully in control over emission and circulation of such tokens, a full fledged application economy can be created. For example, users can earn FTs for performing actions, and spend them to buy or mint new NFTs. Another exciting option is creation of [Decentralized Autonomous Organizations](#) (DAOs), in which FTs can be used as a membership (or governance) tool. In such scenarios, tokens are awarded to members and can be used to vote on decisions or participate in community events.

But we can push our tokens even further, by tying them to other cryptocurrencies and giving them a real-life monetary value. That's where [Decentralized Finance](#) (DeFi), and especially [Decentralized Exchanges](#) (DEX) come into play. We won't go into details here, but at the core a [liquidity pool](#) for a Fungible Token can be created on DEX, which allows trades of this token for other tokens or [stablecoins](#) . This opens the door for a new gaming model [Play-to-Earn](#) , where players can earn real-life money just by playing a game. [Edit this page](#) Last updated on Apr 25, 2023 by An Vo Was this page helpful? Yes No

[Previous NEAR Protocol](#) [Next NFTs for Web 2.0 Apps](#)