**Why Blockchains are Here to Stay**

<https://medium.com/learning-machine-blog/why-blockchains-are-here-to-stay-df1d290c336>

Blockchains can’t do everything, but they do two things really well

1. return ownership of personal data to individuals.
2. verify claims about identity

How public blockchains work?

Public blockchains are called “public” because anyone can use them. Similarly, the internet is public because anyone can use it.

When it comes to the question of data security, in many ways public blockchains are much *more* secure than the internet.

This is because the data that needs to be verified is not recorded on the chain directly, but rather linked to the chain through various one-way methods. Some of those include

* hashing,
* pairwise decentralized identifiers (DIDs), and
* zero-knowledge proofs (ZKPs).

These technologies are all specifically designed to make off-chain data verifiable using public blockchains without putting the actual data on the chain.

This means that data can also be “de-linked” when necessary.

These methods make using public blockchains far more secure than a traditional database, because a public blockchain cannot be “hacked” to reveal sensitive information.

Without blockchains, individuals will continue to be dependent on platforms owned by large software providers to access and use their data.

Data ownership is also one of the big advantages of using *public* blockchains specifically.

If you’re using a private blockchain, the company or government running that chain still owns the network used to verify your data and prove your ownership.

But if you’re using a blockchain that no one owns, you can take your data with you wherever and verify it wherever without worrying about losing access or the ability to verify.

Hyperledger & Corda is not a Blockchain

Private blockchains, or DLTs, that require permissions to use can also have value. These are like traditional databases in that they generally record much more data on-chain and have an access control system that authorizes interactions with the DLT.

The more data you put on-chain, obviously the more you have to restrict who has permission to use that chain.

This is why public blockchains record very minimal data, while private blockchains often record lots of data.

Distributed ledgers, even though they are permissioned and more centralized than public blockchains, can still do things spreadsheets can’t — for example, cut down the time it takes to [verify the provenance of a mango or a diamond from 7 days to 2.2 seconds](https://techcrunch.com/2018/09/24/walmart-is-betting-on-the-blockchain-to-improve-food-safety/).

ERC20 Token Standard: A Beginners Guide

<https://medium.com/cryptoxtech/erc20-token-standard-a-beginners-guide-5b5f0c87e11f>

ERC20 which is basically a template, on the basis of which you can create your own token.

Think of MS Word or Google Docs. You can either create your own document or you can go with a template. Ethereum has the same ideology.

You can either create your own token from scratch or you can go with the ERC standard (template) to make changes according to your needs and customize your token the way you want.

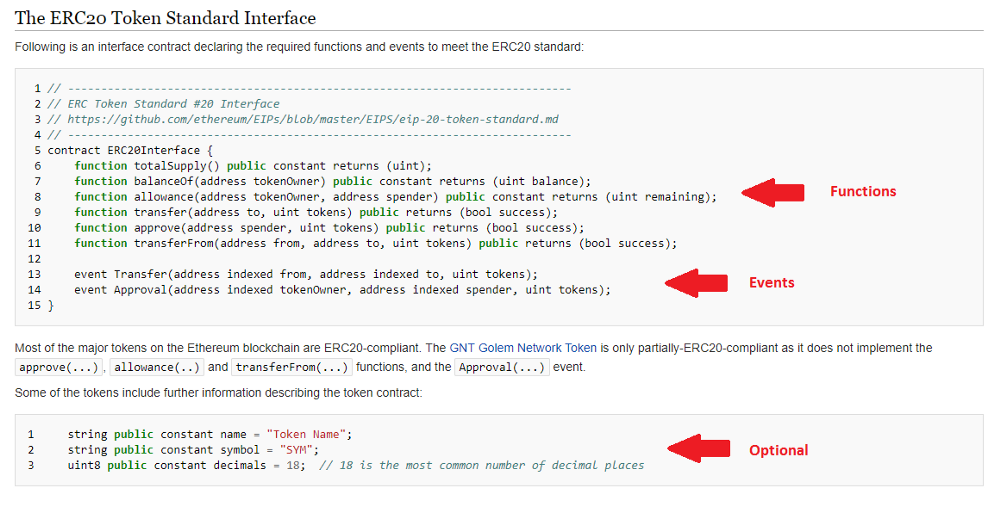
All tokens are essentially [Smart Contracts](https://blockgeeks.com/guides/smart-contracts/).

Smart contracts help you exchange money, property, shares, or anything of value in a transparent, conflict-free way while avoiding the services of a middleman.

The function of each smart contact is to create the token, handle the transactions and keep a check on the balances of the user.

Just like any contract contains certain terms and conditions, these smart contracts contain a set of functions, which define the behavior of the token, written in a programming language.

In Object Oriented Programming term, you can think of ERC as a simple interface, which has to be implemented by developers to make their own token.



But why do we need ERC20 Tokens? Why can’t just everyone make their Token Contracts in their own style?

Let us say you with your friends have to submit a report on a particular topic. Now if you all make your reports with a template provided by Docs, the person assessing your report will know exactly where to find the information and what is the structure of the report.

On the other hand, if all of you prepare your reports with different structures, then it will be relatively difficult to assess and evaluate them.

Similarly, while making your own Token Contract in your style is possible, it can have**certain disadvantages**.

If the code structure of all the tokens is different, then

* exchanges,
* wallets,
* smart contracts and
* other developers

will have to write **custom code** to communicate with each type of token.

With ERC20, as soon as you launch your token, third parties can already support the token as they have support for Ether as well as other ERC20 tokens.

This means your token can **instantly** be listed on exchanges and users can trade them with their existing wallets. Thus, it establishes interoperability.

The Dark Side of ERC20

The companies wanting to launch their own tokens do not have to go through the long process of making their own blockchain, thus, saving time, effort and money.

The two main bugs (errors) here are

1. **batchOverflow** bug
2. **Unexpected Money Losses**

The batchOverflow Bug

This bug helped attackers to possess a lot of tokens out of nowhere.

Unexpected Money Losses

All tokens are smart contracts and these contracts define the behavior of the token. These contracts are designed to work with only specific types of tokens.

So if you send ether to a smart contract which is not built to work with ether, it should return your funds back to you and the transaction would not be completed, right?

Wrong! This is where the tragedy happens with ERC20 tokens. It is not designed to work in this way.

When you send a ‘non-compliant’ coin to the smart contract, the transfer function in that contract produces weird results like permanent loss of funds of the sender.

Around $3 million have been [lost](https://github.com/Dexaran/ERC223-token-standard) due to this bug as of December 2017.

The Solution

The [ERC 223](https://github.com/Dexaran/ERC223-token-standard) standard improves upon the ERC20 standard and is backward compatible too.

This means that all the contracts and services interacting with ERC20 tokens will be able to interact with ERC223 tokens as well without any changes.

One of the main advantages of ERC223 over ERC20 is that it allows the developers to handle incoming transactions and provides a possibility to avoid losses arising due to the inability of a contract to work with the incoming token, thus, solving the occurrence of unexpected losses with ERC20.

ERC20 Token Standard is a huge catalyst in the growth of ICOs and the number of projects on the Ethereum Blockchain.

Various other standards are such as ERC223, ERC777 and ERC1155.

What is the difference between a Coin (Ether), a Token, and a Smart Contract on Ethereum?

<https://www.quora.com/What-is-the-difference-between-a-Coin-Ether-a-Token-and-a-Smart-Contract-on-Ethereum>

Smart Contract

Smart contracts are a series of instructions, written using the programming language “solidity”, which work on the basis of the IFTTT logic aka the IF-THIS-THEN-THAT logic.

Ether

Every single step in a smart contract is a transaction or a complex computation and would have a cost that is measured in “gas”. The price of this gas is paid by the requester in “Ether”.

Ethereum Token

Think of Ethereum like the internet and all the [DAPPS](https://blockgeeks.com/guides/dapps-the-decentralized-future/)as websites that run in it. There is something really interesting about these DAPPS, they are all decentralized and not owned by an individual, they are owned by people. The way that happens is usually by a crowd-sale called the “ICO”

Basically, you buy certain tokens of that DAPP in exchange of your ether.

These tokens are usually of 2 varieties:

* Usage Tokens.
* Work Tokens.

Usage Tokens:

These are the tokens that act like native currency in their respective DAPPS.

While these tokens have monetary value they won’t give you any particular rights or privilege within the network itself.

Work Token:

These are the tokens that identify you as a sort of shareholder in the DAPP.

A perfect example of this is the DAO tokens. If you were a DAO token holder, then you had the right to vote on whether a particular DAPP could get funding from the DAO or not.

Why Do We Need Tokens?

Right now you must be wondering, if all these [DAPPS](https://blockgeeks.com/guides/dapps-the-decentralized-future/)are made in the Ethereum Network, then why don’t we simply use Ether to pay for every transaction within those DAPPS?

Remember that time you went to the water park? Remember how they took your money and tied a band around your wrist which you used to gain access to all the rides in the park and to buy food as well? In this example the water park is the DAPP, your money is ether and the band is the token.

By using tokens to execute certain functions in the smart contract of the DAPPS you make the process much more simple and seamless. Plus, tokens are also great for the overall value of ether as well.

A coin(ether) = fuel to run the network/transactions

A token = is more of a utility tool used to perform an action

Smart contract = protocol that self-executes in a way determined by the code written.

Journey to Scale on Ethereum — State Channels, Plasma and Plasma Cash

<https://blog.ethlend.io/journey-to-scale-on-ethereum-state-channels-plasma-and-plasma-cash-4fe7914e06a>

The Difference Between Blockchains & Distributed Ledger Technology

<https://towardsdatascience.com/the-difference-between-blockchains-distributed-ledger-technology-42715a0fa92>

Every blockchain is a distributed ledger, but not every distributed ledger is a blockchain. Each of these concepts requires decentralization and consensus among nodes.

**However, the blockchain organizes data in blocks, and updates the entries using an append-only structure.**