Thijs Maas Follow
Aspiring crypto-lawyer—founder of www.lawandblockchain.eu
Oct 4 · 13 min read

Understanding Ethereum—a full guide

Ethereum is the subject of a lot of hype lately. It is praised by some as the new internet or the world's computer and criticised by others as a platform that enables widespread scams and ponzi schemes to thrive. I see badly informed articles about Ethereum, smart contracts, DApps, DAO's, ICO's and tokens on the daily so it is time to analyse the subject. I will present the argument that Ethereum might form the main protocol enabling the 'internet of value'.

It is time to take a closer look at what Ethereum is, what it enables and what role it could play in the future. But first, we have to backtrack a bit. Time for a small reminder about the value of distributed ledger technologies.

Decentralized and 'trustless'

To put it briefly, a blockchain is a database that is secured and maintained by a large number of nodes worldwide. Data is stored in blocks, which are time-stamped and linked to each other to form a

chain. The blockchain makes sure that all data stored in the chain is secure and immutable forever, which is what makes the blockchain so interesting. Through its consensus mechanism, the blockchain is 'trustless', which means that one does not have to rely on the trustworthiness of a counterparty.

It's trustless, so what?

For the first time, the blockchain enabled a safe way to send value on the internet without needing a trusted third party. This led to the rise of Bitcoin and other cryptocurrencies.

In essence, a Bitcoin transaction is just the execution of a piece of code stating that 'person A' sent X amount of Bitcoin to 'person B'. This transaction is self-enforcing. To execute, it first checks if there is enough Bitcoin in person A's digital wallet and then sends the money to B's wallet. In itself, this transaction is a piece of code that automatically executes if a certain variable is met. It is a digital contract. This (smart) contract was the first application of a piece of code that could be fully trusted to (a) act in the way it was designed and (b) be immutable, without the need of a controlling authority.

Computer code however, can be used to do much more than this.

Vitalik Buterin realized early on that if a cryptocurrency could potentially disrupt traditional finance by eliminating the need for trusted third parties, then smart contracts can disrupt many more industries in which intermediaries are needed to provide trust. He started playing around with Bitcoin's coding (scripting) language, but found that it was not flexible enough for the amount of applications he imagined. As such, he created Ethereum.

Ethereum

Ethereum is a programmable blockchain. Like any blockchain, Ethereum is based on a peer-to-peer network protocol consisting of many computers worldwide. Instead of providing users the ability to use a few pre-defined operations (e.g. bitcoin transactions), Ethereum allows its users to run pretty much any code they want. The code is stored on the blockchain for others to interact with and is often referred to as smart contracts.

The computers (nodes) in Bitcoin's network maintain and update the blockchain. In Ethereum, they also run the Ethereum Virtual Machine (EVM). The Ethereum Virtual Machine is a supercomputer that combines of all computing power of the nodes in the network combined. This computing power is used to run the user-submitted code (smart contracts) on the blockchain. To execute these, the EVM charges a very small transaction fee in exchange for the computational power used by the smart contract. This fee is called 'gas' and it is paid in Ether, which is why Ether should not really be seen as a cryptocurrency, but rather as the the oil to run the network.

> *Ethereum is a supercomputer, which lets users run pretty much any code they want.*

Smart Contracts

As mentioned, Ethereum allows people to deploy smart contracts on the blockchain. A smart contract is a self-executing piece of code, which defines and executes agreements between multiple parties. The idea of 'smart contracts' was first conceived by Nick Szabo in 1994. He reasoned that code is perfectly able to define a series of relations, parameters and actions.

In Ethereum, smart contracts can be written in Solidity, an Ethereum specific programming language. These smart contracts can then be uploaded to the blockchain and will continue existing there. Because the blockchain is secure and immutable, one can fully trust that a smart contract on Ethereum will execute like intended. Self execution results in that counterparty risks and moral hazards are essentially eliminated from the equation, as the contract enforces its own provisions.

Vitalik Buterin explains smart contracts as following in Ethereum's white paper:

> *"Contracts in Ethereum should not be seen as something that should be "fulfilled" or "complied with"; rather, they are more like "autonomous agents" that live inside of the Ethereum execution environment (EVM), always executing a specific piece of code when "poked" by a message or transaction, and having direct control over their own ether balance and their own key/value store to keep track of persistent variables."*

Smart contracts can be used for a wide variety of different transactions. They are very suitable for simple transactions, in which obligations of both parties can be boiled down to easily verifiable set of requirements. Think for example of financial products (such as derivatives, futures, options etc.), simple consumer transactions, Internet of Things data exchange or automatic payments of musicians or content creators per minute of 'use' of their intellectual property.

As a further example, I could write a smart contract that acts as an insurance against drought for farmers. The basic parameters would be as follows:

weather derivative

- if, sensors from KNMI (Royal Dutch Meteorological Institute) indicate that less than 6 mm of rain has fallen in the past 3 months; and
- if, user has deposited a monthly insurance premium defined in the contract
- then, send amount X money to user.

In my example, the KNMI sensors act as an 'oracle' to get trustworthy up-to-date information for the contract to execute. An oracle is a trusted third-party, sensor or blockchain that can provide information needed for the contract.

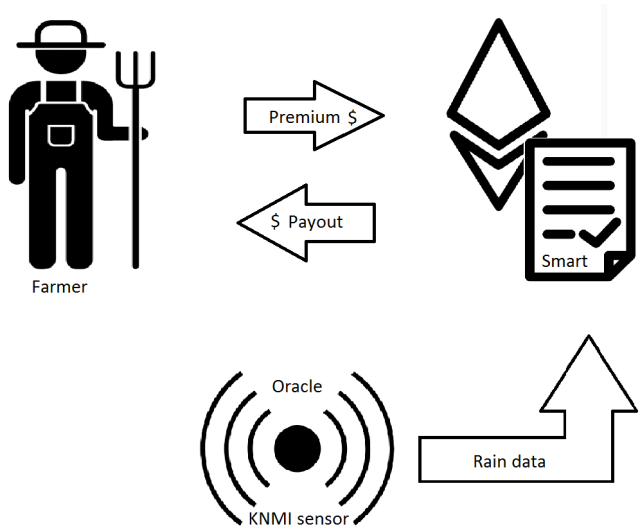

Illustration 1: Farmer Insurance visualised

Decentralized Applications

Now, let's say that, instead of just one insurance product, I want to start a whole insurance company on the blockchain. We'll call it BlockSurance. Firstly, I would want more than one product, and would code more than one smart contract. The different contracts would be coded for a variety of products and functions.

That smart contracts on Ethereum can interact with each other, is very helpful for BlockSurance. After a very dry year, I would not want the 'farmer-drought insurance' contract to run out of funds to compensate the farmers. As such, I could make my different insurance products (smart contracts) send the insurance premiums to one central smart contract that would act as the general reserve. If the farmer-drought insurance contract concludes that a farmer is entitled to X amount, a message will be sent to the general reserve contract to pay the farmer X amount, which will then automatically execute.

BlockSurance's smart contracts are stored and executed on Ethereum's blockchain, which means they are 'trustless': my customers don't have to trust me to know that they will receive a payout when they are entitled to one, as the contracts execute themselves. My customers are also ensured that these contracts cannot be hacked or changed because the blockchain is immutable. There is no counterparty risk as long as the code in the contracts is audited by the Ethereum community to confirm that they will work as intended/advertised.

Of course, BlockSurance is going to need customers—and I want their user experience to be as easy as possible. So, I will set up a website and create a great user interface. This website will be the front-end of my application, while Ethereum's blockchain and my smart contracts are the back-end.

If I have done all of the above, I have essentially created a decentralized application, or DApp. However, A DApp is increasingly understood to not only be an application running on a blockchain, but also to have its own cryptographic token.

Tokens

So what is a token?

A token is a new sort of digital property, the value of which is determined by supply and demand. A token grants a set of rights to its owner, in relation to the decentralized application it was created for. A token can for example be used for governance of the project, distribution of equity rights or access to the decentralized application it was created for. The widely varying characteristics of a token makes its legal nature a very interesting topic to examine.

For BlockSurance, let's imagine I create a token as well. This cryptographic token could for example be used for access to the DApp: to use my insurance, one would have to pay the insurance premium with the token.

Moreover, if the insurance turns out to make more profit than needed, I could have my token holders vote on what should be done with this profit. I could do so by setting a threshold in my general reserve contract, which, once every year, if crossed, allows token holders to vote for:

- a profit distribution among token holders
- a reduction of the insurance premium
- an increase of insurance payouts
- the deployment of funds to developers

For every voting option I just deploy a smart contract which allows users to send tokens to it during the voting period and returns them afterwards. The contracts communicate with each other at the end of the period to see which received the most votes. Whichever contract received the most will automatically execute.

To make a my own cryptographic token, I used to be required to make my own blockchain. Building a whole blockchain is a time-consuming process as I would have to design its protocol, consensus mechanism and set up nodes. If I don't have enough nodes it could easily be subject to a sibyl or a 51% attack. As a hypothetical developer, I don't want to have to worry about security or designing a whole blockchain.

Ethereum solved this problem by not only allowing the deployment of smart contracts, but also the creation of tokens on top of its blockchain. But how these tokens distributed?

Initial Coin Offerings

Tokens are often distributed for the first time through an Initial Coin Offering, or ICO. In an ICO, investors and potential users of the platform can buy tokens by exchanging them for Bitcoin or Ethereum. It is a form of crowdfunding which lets developers raise money while early adopters and opportunistic investors speculate on the future value of the project's tokens. The ICO is also known also known as a token distribution. This year, more than $2.2 billion has been raised with an ICO so far.

By doing an ICO for BlockSurance, I can raise money to hire developers to improve the front and back-end of my decentralized application. The ICO also gives my insurance company a ton of exposure and helps kick start my business. Optionally, I could also put a part of the funds collected in the 'general reserve contract' or pay back my seed investors. I am also going to pocket a part of the tokens for myself, resulting in a distribution like this.

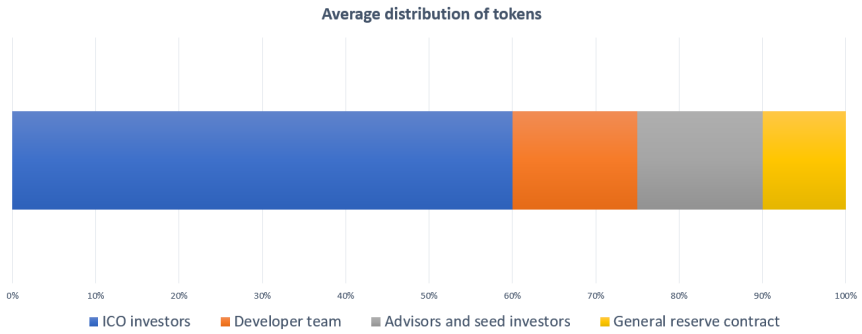

Illustration 2: Distribution of tokens

Most ICO distributions look something like this. Of course, the exact percentages can vary, but almost always developers give themselves a share of the tokens. The funds raised from ICO investors go towards the actual development of the project.

Decentralized Autonomous Organisations

A Decentralized Autonomous Organisation (DAO) is an organisation that is completely governed and managed by the rules encoded in its underlying bundle of smart contracts. A DAO is essentially a DApp in which the developers do not play a special role anymore once it is launched. The difference between a DApp and a DAO is one of

governance and autonomy. A DApp's source code and future road map is governed by a group of developers or an organisation, while a DAO is truly autonomous. What the organisation does and what it will do is *all* previously determined in its code. In fact, the code is the organisation itself.

In other words, in a DAO, the future development (if any) is fully determined by in-built governance mechanisms that let tokens holders vote for the deployment of the raised funds. Voters could for example vote for the DAO to invest in other projects. Another option is allowing any developer to submit proposals for changes or additions to the DAO, which can then be voted on. If successful, a developer will get permission to implement his proposal and get paid from the raised funds. For this to work, the smart contracts to allow for such voting-based implementations have to be built-in the DAO from day one.

At the time of writing there are 266 DApps & DAO's. Most of these are deployed on Ethereum. Some notable examples are:

Dapps:

- Augur—A prediction market
- Golem—A market place for idle computing power
- Basic Attention Token—A digital advertising exchange and browser with an 'attention-based' revenue model for content creators
- Status: A decentralized browser, messenger with in-built access to DApps

DAO's:

- The DAO: A crowd-sourced investment fund (also known as the biggest failure in the short history of smart contracts)
- MakerDAO: A DAO that aims to minimize price volatility of its own token
- DigixDAO: A marketplace that allows for trade in a gold-backed token on the blockchain

Use cases and the 'internet of value'

What Bitcoin did was allow individuals to exchange cash without involving any middlemen, like banks, payment processors or the government. For Bitcoin's use case, the trust these institutions provide is not necessary.

Ethereum's impact may be more far-reaching. As almost anything can be coded and secured on its blockchain, other intermediary actors that provide trust could potentially be replaced. Depending on how much value you believe these intermediaries bring aside from trust, these actors could include notaries, bankers, the insurance sectors, real estate agencies, online marketplaces for goods and services like Uber and Ebay etc. The argument goes that almost anything of value for which intermediaries are necessary to provide trust can be done more efficiently on the blockchain.

Although this argument has its flaws, it is clear that blockchain technologies empower individuals from around the world to deal with each other by eliminating counterparty risk from the equation. Inevitably, this means that, to some extend, trusted intermediaries lose their importance in our economic system. We will see a shift away from these parties towards peer-to-peer services and the sharing economy. A shift away from a centralized economy to a distributed one.

The internet distributed information in a manner that was not conceivable before the rise of the technology. The blockchain allows for the trustless exchange of value over the internet, pushing us to challenge how we have structured society, defined value and rewarded participation. This is why blockchain technology is referred to as the driving force behind the 'internet of value' or the Internet 3.0. Ethereum is already the biggest protocol that allows the creation of smart contract, DAO's and DApps, and many believe it might just become the backbone of this new internet.

The Enterprise Ethereum Alliance and permissioned chains

The Enterprise Ethereum Alliance (EEA) is a network for academics, companies and startups to share knowledge and 'to define enterprise-grade software capable of handling the most complex, highly

demanding applications at the speed of business'. This network is quite extensive, with notable members such as JP Morgan, Microsoft, Master Card, Intel, Deloitte and BP. Considering the possible use cases for blockchain technology, it is logical that a wide variety of companies want to experiment with their own implementations to increase efficiency within their respective sectors.

Illustration 3: non-exhaustive visualisation of Enterprise Ethereum Alliance Members

It is often argued that the EEA is a driving factor in Ethereum's valuation, but this is not (or better said, should not) be the case. The companies are experimenting with the deployment of private (for use within one company) and consortium chains (in which multiple parties with existing relations can benefit from having a shared ledger for a specific use case). These 'permissioned' blockchains are private 'forks' or private versions of Ethereum. A private fork of Ethereum allows companies to restrict user access to certain parties. The nodes to secure the network are supplied by the companies involved. Because there is few parties to the network, there is more flexibility in changing the rules of the blockchain, reverting transactions, modifying balances etc. There are also less transaction costs (because there are less nodes) and common risks in public blockchains such as 51% attacks are not possible, as all validators are known.

On a personal note, I would like to add that I believe most people overestimate the potential of permissionless chains and underestimate the potential of permissioned chains. I am not questioning the efficiencies that well-structured blockchains can bring to many sectors.

However, as said before, permissioned chains are more efficient than their permissionless counterparts due to the differences in the difficulty of reaching consensus. I am not saying that transparency, censorship resistance and security are not great in and of itself, but for a decentralized project to succeed, much more is needed than just those attributes (think user experience, competitive costs etc.).

Conclusion

Blockchains give the potential to provide efficient, fast, secure, reliable and auditable transacting of value. Like the internet before it, the blockchain promises to upend business models and disrupt industries. Blockchain technologies are pushing us to challenge how we have structured society, defined value and rewarded participation. This is why some refer to blockchains as the enabling technology for the internet 3.0, or the internet of value.

As almost anything can be coded and deployed on Ethereum's blockchain, Ethereum might just form the base of this new internet. However, high potential comes hand in hand with hype, and talk of the internet 3.0 brings problems resembling the ones surrounding the dot-com crisis.

Edit: This piece has focused solely on the positive aspects of Ethereum. There is however, a lot of valid criticism on the project, which I decided to summarize in a new article because this one became too long.