ETHEREUM AND SMART CONTRACTS

– INTRODUCTION –

2021

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

TRANSITION

Funded by the Erasmus+ Programme of the European Union

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

SOMMARIO

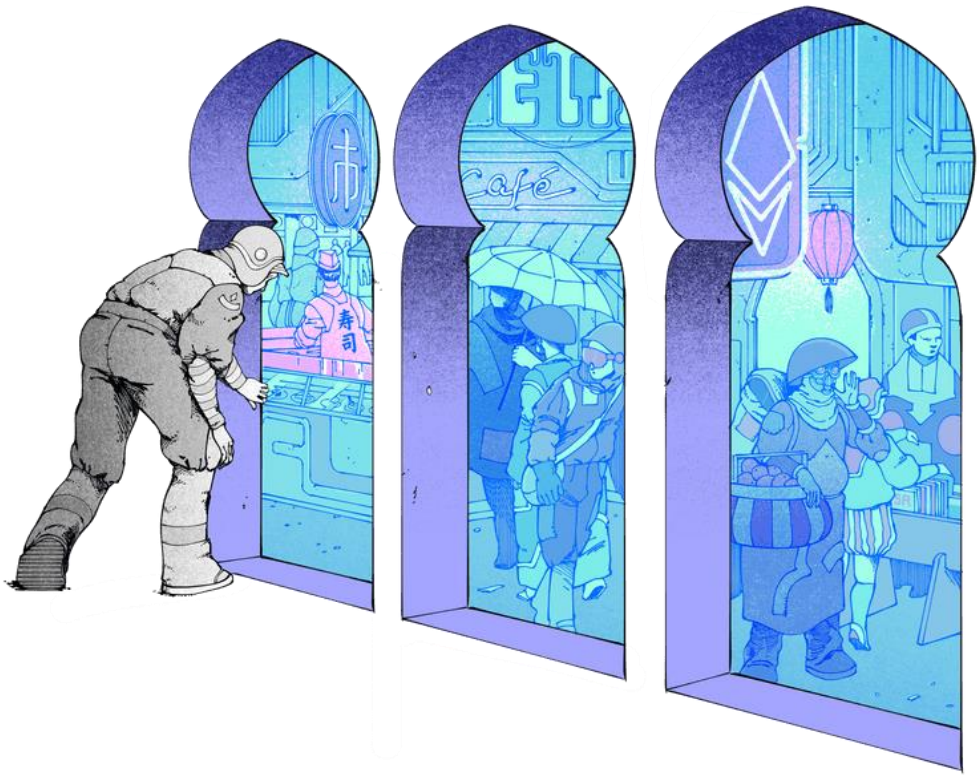
TRANSITION

Funded by the Erasmus+ Programme of the European Union

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

1. What is Ethereum?



[1]

1.1 Definition and General Overview

Ethereum has been first proposed through an introductory paper published in 2013 ("*Whitepaper*") by Vitalik Buterin – founder of Ethereum – before the project's launch in 2015. The latter started with the "*Yellowpaper*" by Dr. Gavin Wood and consists in a technical definition of the Ethereum protocol. But, what is Ethereum exactly?

Ethereum is a **peer-to-peer platform** (**P2P**) based on **Blockchain** technology. More precisely, it allows access to digital money, the coding of smart contracts and the creation of decentralised applications (Dapp). For the ease of comprehension, Ethereum may be deemed as a transaction-based state machine where the «**state**» is the set of what is representable by a computer at a given moment, such as an account balance or application data.

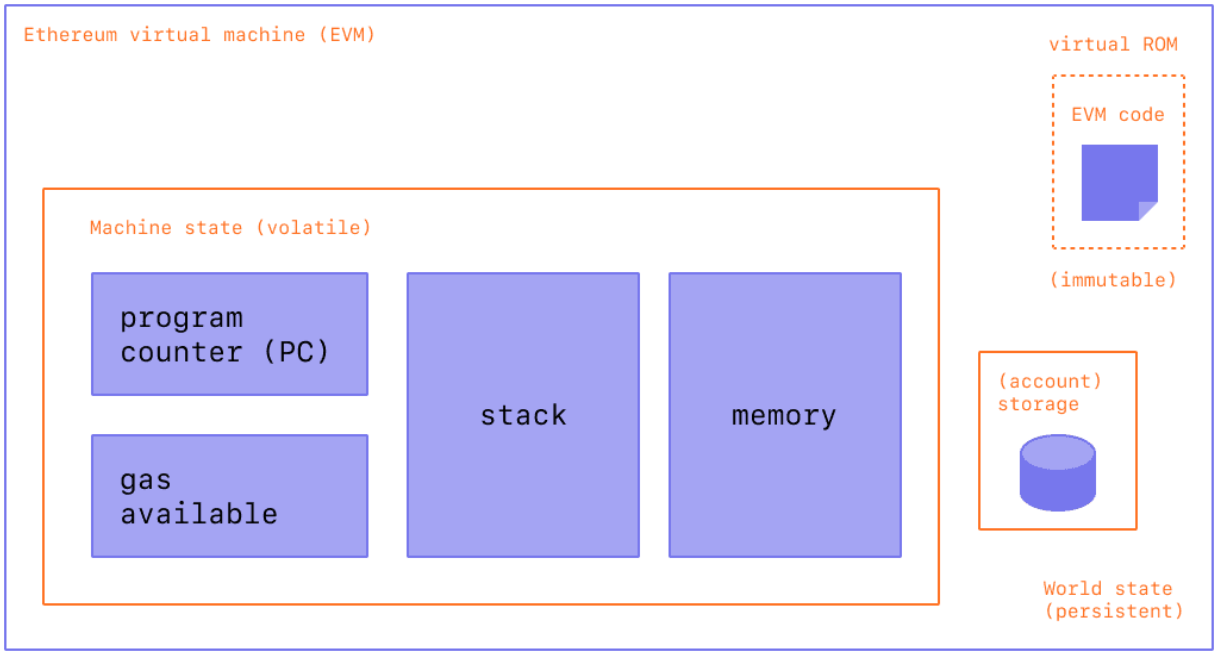
[1] https://ethereum.org/en/what-is-ethereum/

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

Additionally, Ethereum works through the **Ethereum Virtual Machine** (**EVM**) which is a blockchain-based software platform (a decentralised machine) that executes the transactions by changing the state of Ethereum.

Lastly, Ethereum has also its own cryptocurrency, called **Ether (ETH)**, which can be used on the internet (it is similar to Bitcoin). Differently from "normal" currencies, Ether has no company or bank that can print more ETH or change the relevant terms of use.

1.2 Ethereum Virtual Machine



[2]

The Ethereum Virtual Machine (EVM) is a computational engine which acts like a decentralized computer. It acts as the virtual machine, milestone of the entire operating structure of Ethereum, and it runs execution and smart contract deployment (as for the contracts, they are written in the smart-contract coding (Solidity), however, a transformation is required since EVM can only read bytecode).

That said, the EVM works like a decentralized computer to complete several types of tasks within the blockchain. It is deemed as a **turing-complete virtual machine** that executes codes exactly as intended, being the runtime environment for Smart Contracts. In other words, it is able to solve any

[2] https://ethereum.org/en/developers/docs/evm/

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

computation problem and the code running on it has no access to other processes on your computer, making EVM fully isolated.

Every computer running software of the network (so-called "Ethereum **Node**") runs on the EVM in order to maintain consensus across the blockchain. As a consequence, there will be a running copy of EVM in each node of the system giving Ethereum the ability to execute the smart contracts code and the chance to act as a decentralised global computer.

Thanks to the blockchain technology, every *change of state* of such "global computer" is registered and every **transaction** or smart contract is processed. More precisely, a new block is broadcasted to the nodes in the network, checked and verified, thus updating the state of the blockchain for everyone.

Ethereum, as a blockchain, uses a consensus mechanism (process through which reaching an agreement about information on the network) which is called PoW (**Proof-of-Work**). It *allows the nodes to agree on the state of* information, therefore ensuring that the chain is difficult to attack or overwrite. Plus, PoW sets the difficulties and the rules for the works of miners (i.e. adding valid blocks to the chain).

It should also be noted that, each blockchain relies on having a single state as a source of truth and users always trust and choose the longest chain since it proves that a lot of work has been done on it. That being said, the scope of PoW is to extend the chain "by consensus", keeping also in mind that it is (usually) impossible to create new blocks that amend/erase transactions or create fake ones, as well as to build a second chain.

The applications able to "run" a node are known as "**Client**" and they can run different types of nodes:

- **FULL NODE**: they store the entire blockchain data, providing data to others on demand, they also participate in the validation of blocks.
- **LIGHT NODE**: they store only the header of the blocks (hash, nonce, timestap, etc.), not the transactions' data, and they do not check all the blocks. They are faster and lighter, as they store less data, therefore they can be used in devices that have a limited amount of memory.
- **ARCHIVE NODES**: they store the whole blockchain kept in the Full Node and build an archive of historical states. In this way, it is possible to demand, for example, the balance of each account in a given block.

For the sake of completeness, the Ethereum virtual machine implies three separate storage areas:

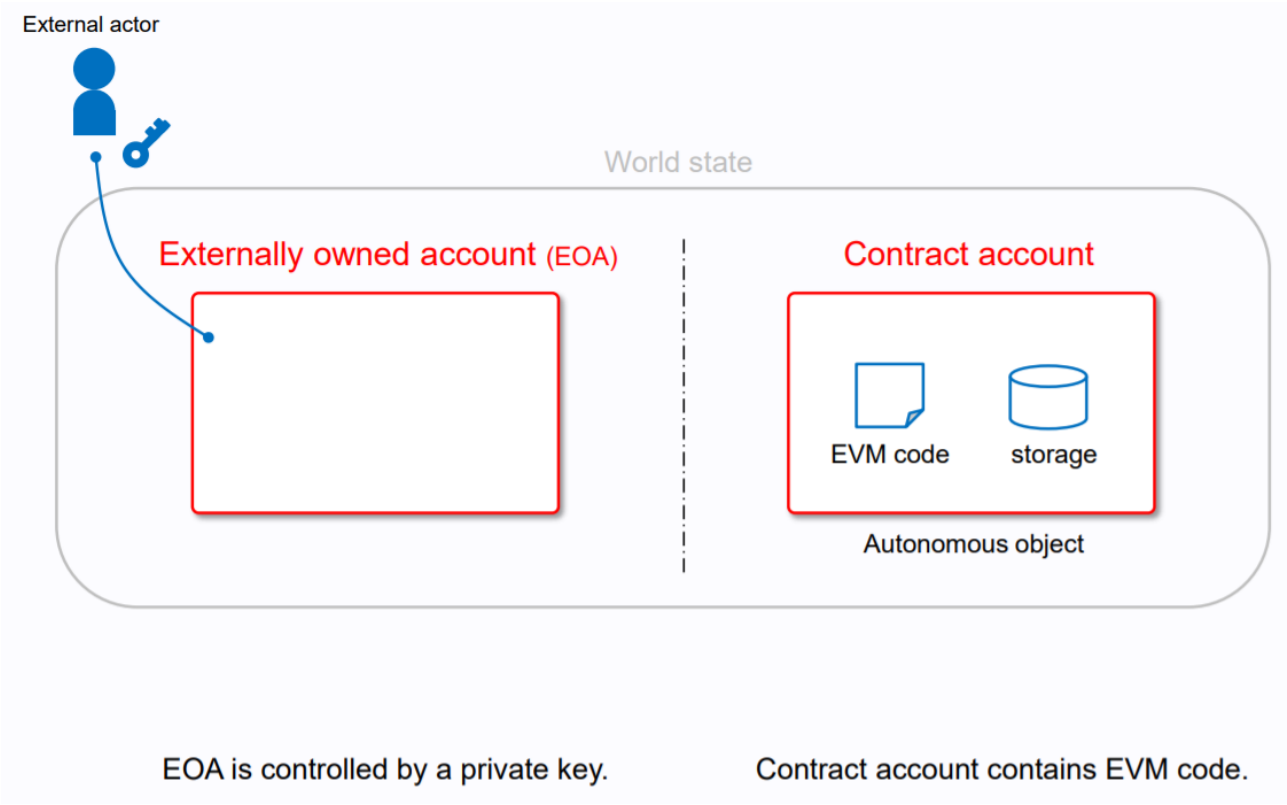
- **Storage:** it is assigned during the process of creating a contract, each account has one and it stores the contract state variables. Note that, none of the contracts can read the storage of or write into another contract.
- **Memory:** it is linear, holds temporary variables (existing in the calling function only) and gets erased between calls. The memory has a limit and a payment in gas is required to expand it

TRANSITION
Funded by the Erasmus+ Programme of the European Union
TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

(it scales quadratically, and the more it grows, the more it costs). However, it is still cheaper than storage.

- **The stack:** area where the computations happen, and it is the cheapest of all three data storing areas.

1.3 Accounts and Keys



[3]

An **Account** is an entity with an Ether balance that can send transactions on Ethereum. In this regard, Ethereum has two types of account which, despite the differences between them, are equally treated under the EVM:

- user-controlled accounts - *Externally Owned Accounts (EOAs)*
- accounts deployed as smart contracts - *Contracts Accounts*

Though both accounts are able to receive, hold and send ETH/tokens, as well as to interact with deployed smart contracts, they have different features that need to be highlighted.

EXTERNALLY-OWNED ACCOUNT	CONTRACT ACCOUNT
– Zero costs of creation – Owned by a real user	– Cost of creation

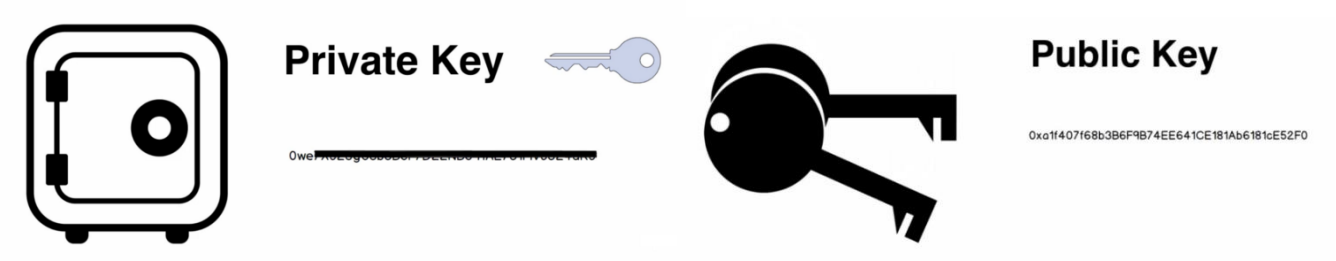
[3] https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

– Able to initiate transactions – Contains a user's private key, meaning that it has direct access over a user's money – Don't contain any code – Send money and change blockchain state	– Owned by the logic of the smart contract code itself – Not able to initiate transaction – Does not have a private key, therefore it cannot initiate a transaction – Has a Smart Contract code – Can send messages to other Contracts (in response to a transaction initiated by an EOA)

For the ease of understanding, an Ethereum account has an **Ethereum address** (like an email address) used to send and receive funds to/from another account. In other words, an Ethereum Address represents an EOA or contract that can receive or send transactions on the blockchain (i.e. the destination address and the source address).

Moreover, an account consists of a cryptographic pair of **keys** (**public** and **private**) which helps to prove that a transaction was actually signed by the sender, simultaneously preventing forgeries.



4

A ***private key*** is a secret number that allows users to prove their ownership of an account or contracts, by producing a digital signature. Plus, a key is what a user uses to sign a transaction, granting the custody over the funds associated with the relevant account. That being said, the user never really holds a cryptocurrency, he/she holds a private key, and the funds are always on Ethereum's ledger.

On the contrary, a ***public key*** is a number, derived from a private key transaction, which can be shared and used by anyone to verify the sender by checking the digital signature made with the corresponding private key.

Private and public key together establish a mechanism that ensures the constant possibility to verify the sender of a transaction, preventing third parties from broadcasting fake transactions.

[4] https://medium.com/@markmuskardin/mastering-the-fundamentals-of-ethereum-for-new-blockchain-devs-part-iii-wallets-keys-and-4cd3175b535b

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

1.4 Wallet

Wallets on Ethereum are applications that let users interact with their **Ethereum account** and therefore manage what the user owns: they give access to funds and Ethereum applications. Obviously, having a wallet is necessary in order to send funds and manage ETH.

A wallet may be deemed as an internet banking app that lets you read your balance, send transactions and connect to applications. Note that wallets don't have custody of funds, the user does.

There are four different types of wallets:

- **Hardware Wallet**, material hardware wallets that let the user keep the crypto offline
- **Mobile Wallet,** mobile application that allows the users access their funds from anywhere
- **Web Wallet,** web wallet allows the interaction with the account through a web browser
- **Desktop Wallet,** desktop application for the management of funds via MacOS, Windows or Linux.

1.5 Token

The Ethereum blockchain allows users to own, besides Ether, also the "**Tokens**" which are used to represent assets (digital or material) or a right to an asset (ownership to an asset or the access to a service). They can be generated by a Dapp or by a Smart Contract and they can be exchanged within Ethereum blockchain.

The Ethereum platform has two different *ERC* standard ("*Ethereum Request for Comment*") to represent tokens:

- *Fungible* Token (ERC-20), it has no features of its own and can be exchanged without distinction with other fungible tokens;
- *Non-Fungible* Token (ERC-721), it has its own features and cannot be exchanged with other tokens that do not have the same characteristics.

TRANSITION

Funded by the Erasmus+ Programme of the European Union

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

2. Smart contract

"A smart contract is a self-executing contract

with the terms of the agreement between buyer and seller

being directly written into lines of code."[5]

2.1 Definition

Smart Contracts are like contracts in the real world. The difference is that they are completely digital: a Smart Contract consists of a computer program which is inside a blockchain. They facilitate, execute, and enforce an agreement between untrustworthy parties without needing to involve a trusted third-party. It can also be defined as a common agreement between two or more parties, which stores information, processes inputs, and writes outputs thanks to its pre-defined functions.

"Smart Contracts" can regulate the exchange of money, properties, shares and any asset with a value. Legally speaking, a contract is deemed as a source of obligations that binds two or more parties. However, a Smart Contract is conceptually different: a smart contract is a self-executing computer program with the terms of the buyer's and seller's agreement directly embedded into lines of code; it implies a connection that takes place thanks to a **cryptographic code**. It's a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain.

Smart contracts can also be defined as transaction protocol. They have ability to verify and execute a contract without any help from third parties.

Smart contracts reduce:

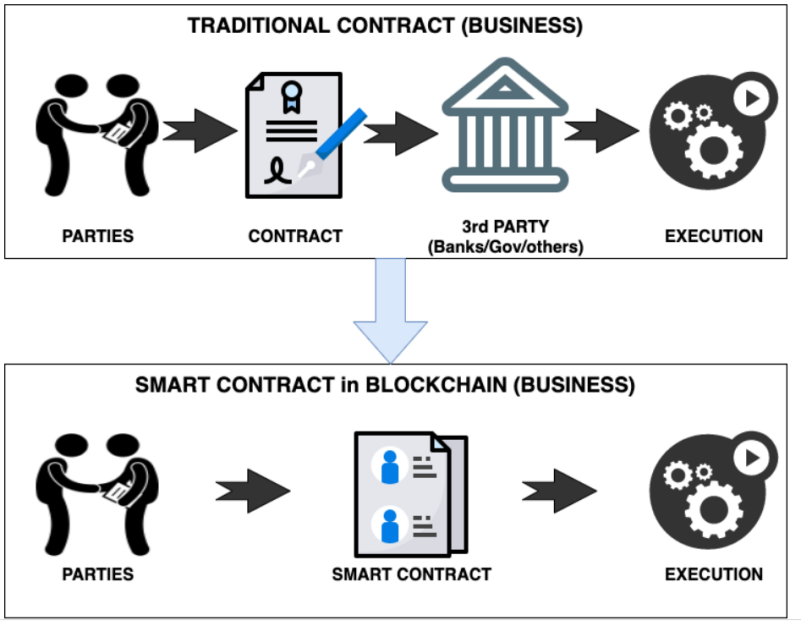
- Time: they can be verified in minutes
- Money: there is no need for trustworthy intermediaries
- Mistakes: there is no need to fill in documents by hands

Smart contracts use the logic called If-Then: the results depend on a certain condition. In his writings, Nick Szabo compared it to using a vending machine: you can only get your purchase after you insert money.

Lastly Smart contracts are written in Solidity which is a programming language.

[5] [1] https://www.investopedia.com/terms/s/smart-contracts.asp

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices



[6]

2.2 Historical Background

Nick Szabo is thought to have first used the term smart contract in one of his articles in 1994 : *"A smart contract is a computerized transaction protocol that executes the terms of a contract. The general objectives of smart-contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimise exceptions both malicious and accidental, and minimise the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitration and enforcement costs, and other transaction costs."* [7]

His idea was basically to convert a traditional contract into code which is embedded into hard- or software needed to enforce the terms of the contract by eliminating the need of a third party. In addition, Szabo believed that smart contracts needed to be made more valuable to society. Therefore, they would have to be verifiable, observable, and enforceable. Hence, they would cause legal barriers to turn lower, transactions costs and execution time to decrease. An additional result would be the creation of new types of business.

Only when in 2008, Nakamoto introduced the Blockchain, it was possible to provide trust within an untrustworthy environment which enabled various technologies to develop. Smart contracts were popularized in 2015, when the Ethereum platform was realized and it can be underlined that Szabo's predictions were correct: today, as smart contracts develop and replace some traditional contracts, they are reducing costs and speed up execution. They have become a trend thanks to the blockchain technology.

- 1994 - Nick Szabo introduced smart contracts

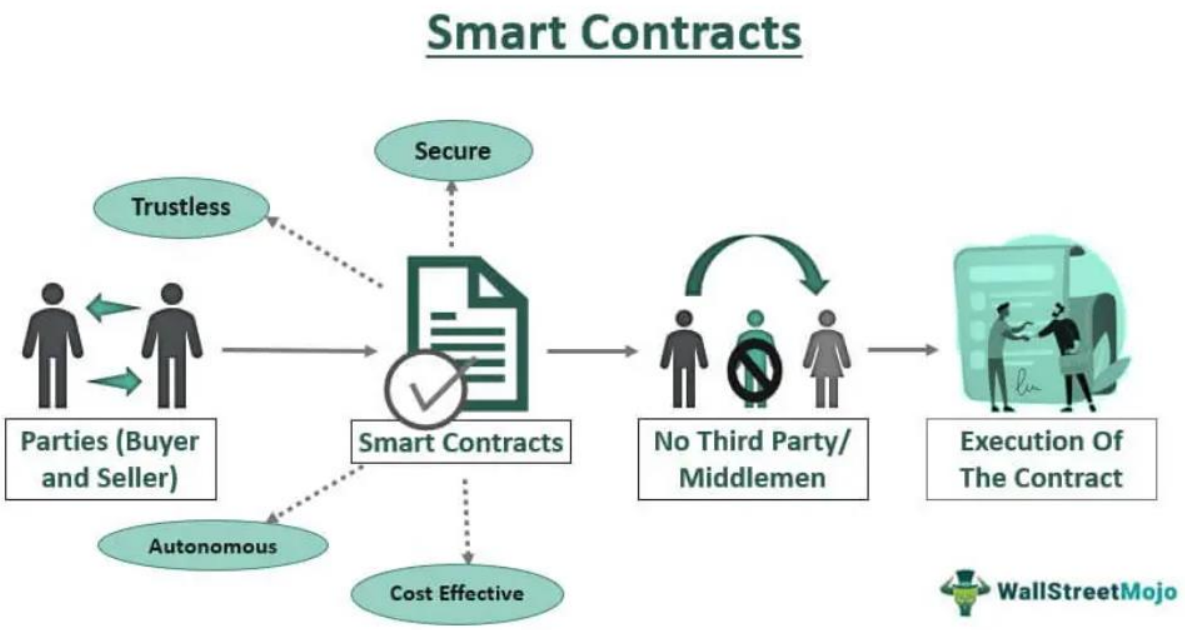
[6] https://medium.com/@markmuskardin/mastering-the-fundamentals-of-ethereum-for-new-blockchain-devs-part-iii-wallets-keys-and-4cd3175b535b

[7] What are Smart Contracts in Blockchain, & how do they work? (blogsaays.com)

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

- 2008 - Nakamoto introduced the blockchain
- 2012 - Rise of the cryptocurrency
- 2015 - Etherum popularized the smart contract concept

2.3 Features of Smart Contracts



[8]

The following are the most significant features of Smart Contracts:

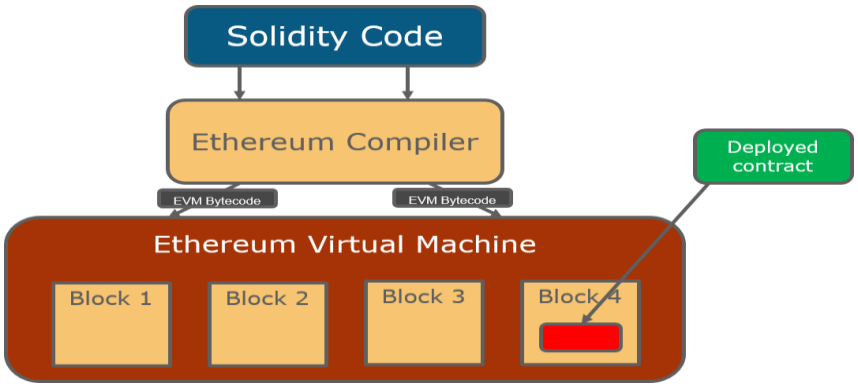
- **Accuracy =** Smart Contracts record all terms and conditions in explicit detail.
- **Transparency =** The terms and conditions of Smart Contracts are fully visible and accessible to all relevant parties.
- **Clear Communication** =Due to the accuracy of the Smart Contract miscommunication, misinterpretation and misunderstanding can be avoided.
- **Speed =** Smart Contracts run on software code on the internet and can therefore execute transactions very quickly.

[8] Smart Contract - Definition, Explanation, Examples & Types (wallstreetmojo.com)

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

- **Security =** Smart Contracts use the same standard of data encryption that modern crypto-currencies use.
- **Efficiency =** Their speed and their accuracy guarantee the efficiency with which they work.
- **Paper Free =** There is no need for paper, therefore they are suitable for those business who would like to reduce their negative impact on the environment.
- **Storage & Backup =** All the essential details of each transaction are being stored. Therefore, in case of data loss they can be easily retrievable.
- **Savings =** There is no need for third parties: lawyers, witnesses, banks etc..
- **Trust =** Smart contracts are transparent, autonomous, and secure and guarantee the impossibility of manipulation, bias, or error. Once solemnized, the contract is executed automatically by the network.
- **Guaranteed Outcomes =** Smart Contracts reduce the need for litigation and courts.
- **Immutable =** Once a smart contract is created, it can never be changed again. Nobody can change they code of the contract
- **Distributed =** Everyone of the network validates a deployed contract. Nobody can force the contract and release funds because all the other network members will become aware of this.

2.4 Creation of an Ethereum Smart Contract



[9]

The main programming language used for writing Smart Contracts is Solidity. It is an object-oriented, high-level and curly-bracket language which has been influenced by **C++** and which is statically typed (the type of a variable is known at a compile time). After having written a Smart Contract using Solidity it needs to be compiled to low-level machine instructions (called **opcodes**) since Solidity

[9] Ethereum Tutorial For Beginners - Ethereum Architecture | Edureka

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

cannot be executed by EVM directly. These opcodes need then to be encoded into **bytecodes** so that the contract can be stored on EVM.

This process is being carried out by using the Solidity Compiler "Solc" which is the main product of the project Solidity.

In order to make a smart contract available to users of an Etherum Network it needs to be deployed by sending an Ethereum transaction containing the code without specifying any recipient. Deploying a contract costs Ether. After having deployed a Smart Contract, it is no longer amendable.

A smart contract can be removed from the blockchain only if the self-destruct operation has been programmed into its code. However, all the transactions will remain part of the blockchains history.

Example of a Smart Contract

```
pragma solidity ^0.4.13;

contract Ownable {
  address public owner = msg.sender;
  /// @notice check if the caller is the owner of the contract

  modifier onlyOwner {
    require (msg.sender == owner) ;
    _;
  }
  address[] pharmas;
  function Add_pharmas(address[] pharmas_) public
  onlyOwner
  {
      for (uint i = 0; i < pharmas_.length; i++) {
        pharmas.push(pharmas_[i]);
      }
  }

  mapping (address => uint) perms;
  function set_permission() public{
        for (uint i=0;i<subjects.length;i++)
        {
            perms[subjects[i]]=3;
        }
        for (i=0;i<pharmas.length;i++)
        {
            perms[pharmas[i]]=2;
        }
        perms[owner]=1;
        //1 is highest, 2 is high, 3 is low
  }
```

10

[10] A Smart Contract example demonstrating ownership and permission levels... | Download Scientific Diagram (researchgate.net)

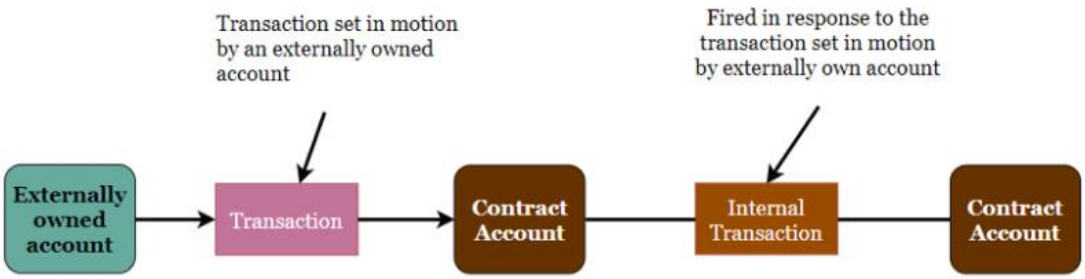
TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

TRANSITION

Funded by the Erasmus+ Programme of the European Union

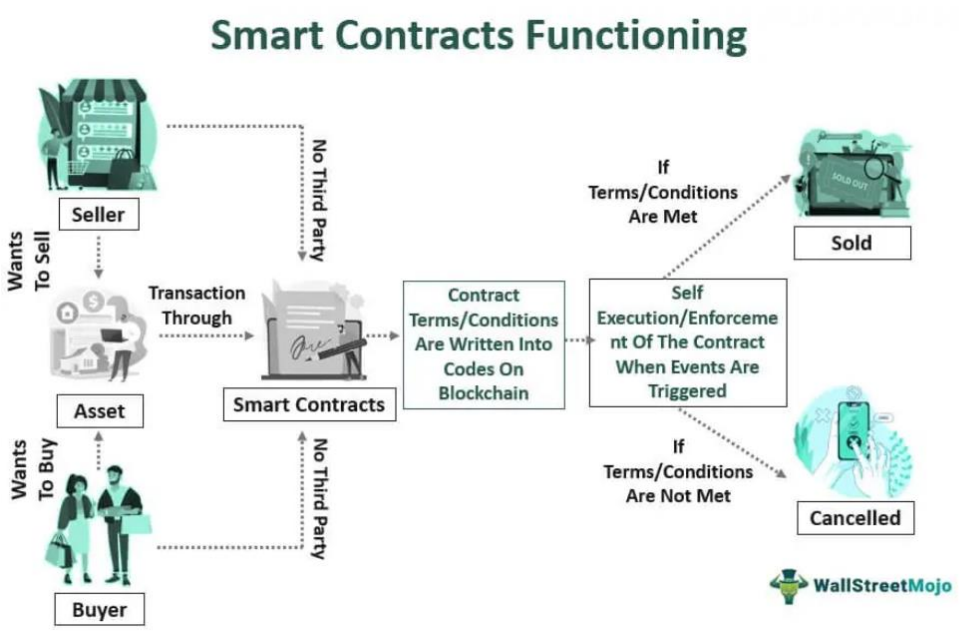
2.5 How does a Smart Contract work?

A smart contract is a type of Ethereum account and therefore also called Smart Contract account. Smart contact accounts do not have private keys but only public keys. They cannot be controlled by users but run as programmed.



11

A Smart Contract can send messages to other Smart Contracts on Ethereum in response to a transaction initiated by an EOA, but it cannot start any type of transaction due to the fact that it does not have a private key. Transactions can only be generated by EOAs; they are "signed messages" because they include what's known as a **"digital signature"** in cryptography. A digital signature proves the identity of the person, who initiated the transaction.



12

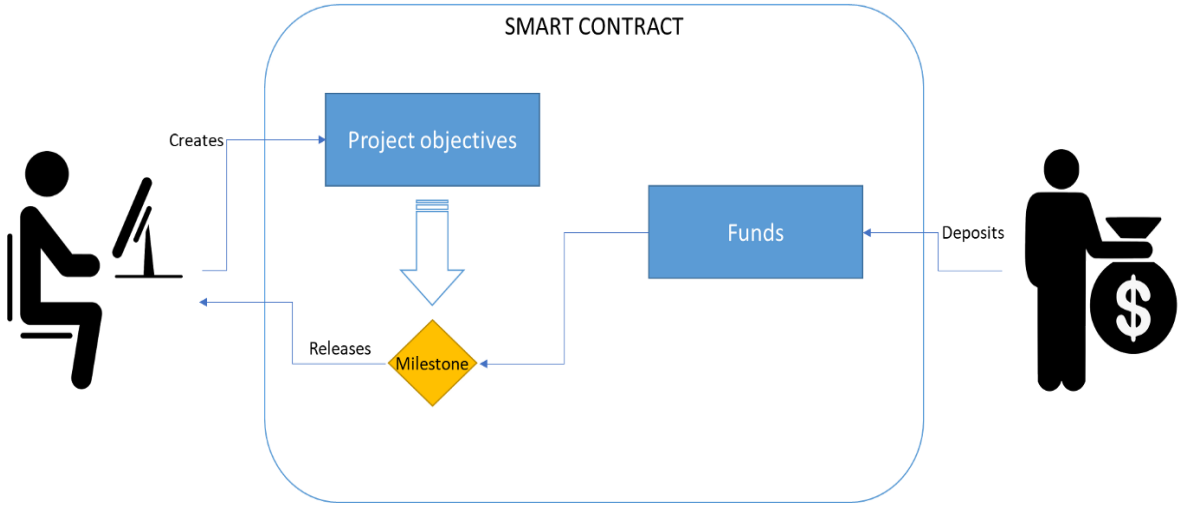
[11] Ethereum account. This post is a reference to "How does… | by 胡家維 Hu Kenneth | Coinmonks | Medium

[12] Smart Contract - Definition, Explanation, Examples & Types (wallstreetmojo.com)

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

TRANSITION

Funded by the Erasmus+ Programme of the European Union



Example 1:

Example for Fundraising: *"A smart contract can be programmed so that it holds all the raised money until the goals of the project are met. The donors can transfer the money to the smart contact. Only if the project is fully funded the money will be transferred to the creator of the project. Otherwise, the money will go back to the donors."*[13]

Example 2:

"Let's imagine that John wants to buy Mike's house. This agreement is formed on the Ethereum blockchain using a smart contract. This smart contract contains an agreement between John and Mike. In the simplest terms, the agreement will look like this: "WHEN John pays Mike 300 Ether, THEN John will receive ownership of the house". Once this smart contract agreement has been put into place, it cannot be changed — meaning John can feel safe to pay Mike 300 Ether for the house. Without the use of a smart contract in this scenario, Mike and John would have to pay lots of fees to third-party companies. Including the bank, a lawyer and a house broker." [14]

2.7 Smart Contract and Law

There are two types of Smart Contracts:

- Smart legal contracts, which are smart contracts on a blockchain that represent - or that would like to represent - a legal contract, along with the issues that involves.

[13] https://stefano-tempesta.medium.com/smart-contracts-simply-explained-5f5140a24c95
[14] https://www.bitdegree.org/crypto/tutorials/what-is-a-smart-contract

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

TRANSITION

Funded by the Erasmus+ Programme of the European Union

- Smart contracts with legal implications, which are artefacts/constructs based on smart technology that clearly have legal implications.

Smart Contracts are changing many areas of private law transactions. It is not possible to tell whether the current system of private law can readily cope with these new forms of 'self-executing' agreements. These contracts also have the potential to create increasing uncertainty in the area of jurisdiction and choice of law.

An evolution of smart contracts towards smart legal contracts has been undertaken by some states in the U.S. which have already issued legislation providing a framework for the commercial and legal application of blockchain technology and the applicability of smart contracts.

In the European Union, legislators are still silent on blockchain technology or, as in Italy, merely give a definition.

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

3. Decentralised Applications (DAPP)

Dapp is the acronym di "Decentralized Application" and it refers to a type of applications whose functioning does not derive from central servers (like any "traditional" application, such as Facebook, Youtube, etc.), they are in fact based on a decentralised network and no decision is taken on central server under the control of an authority. For completeness, they use the Ethereum blockchain for data storage and smart contracts for their app logic. Plus, once on the Ethereum network, nobody can change them.

Dapps, as any traditional application, have three basic structures, and namely:

- ***Frontend***, interface that users use to interact with the App. A Dapp can have frontend code and user interfaces, be written in any language (just like a traditional app), that can make calls to its backend.

  A Dapp combines smart contract and frontend user interface. Moreover, since smart-contracts are accessible and transparent, the Dapp may even include a smart contract written by someone else.

- ***Backend***, it refers to the principal logic of Dapp. As for the Ethereum Dapp, the backend is related to smart contracts that are executed within the blockchain. In this regard, since the smart contract are visible and public, high-level of both transparency and security is granted.

  A Dapp has its backend code running on a decentralized peer-to-peer network, whereas the backend code of an app runs on centralized servers.

- ***Data storage level***, with regard to Ethereum Dapp, the storage is fully decentralized, whereas the storage related to traditional application is centralized.

  The above happens in a safe cryptographic way, preventing non-authorised access from third parties and guarantying the integrity of data and the relevant accessibility.

To sum up the Dapps features:

- **Decentralised**, as they are independent, and no one can control them as a group;
- **Deterministic**, since they perform the same function irrespective of the environment they are executed;
- **Turing complete,** in fact the Dapp can perform any action;
- **Isolated** because they are executed in a virtual environment (EVM), isolated from the blockchain network.

TRANSITION

Funded by the Erasmus+ Programme of the European Union

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

4. Gas & Fees

Each Ethereum transaction requires computational resources in order to be executed, meaning that each operation requires a fee. In this regard, the **gas** is the fee needed by Ethereum to work properly. More precisely, gas represents and measures the computational effort to successfully perform the operation on the network. Note that, gas fees guarantee a secure network, preventing actors from spamming it.

For completeness, these gas fees are paid in ETH and the gas prices are denoted in gwei (a gwei is equal to 0.000000001 ETH (10^{-9} ETH)). Furthermore, for the purpose of avoiding computational wastage in code (such as loops), a limit to how many computational steps of code execution it can use is set.

Example:

- A has to pay B 1 ETH
- The gas limit in this transaction is 21,000 units
- The gas price is 200 gwei

Total fee to be paid: Gas units limit * Gas price per unit = 21,000 * 200 = 4,200,000 gwei → 0.0042 ETH

- A sends the money, 1.0042 ETH will be deducted from A's account
- B receives 1 ETH
- Miner gets 0.0042 ETH

For the ease of understanding, the gas limit represents the maximum amount of gas the user is willing to consume for the transaction. Obviously, complicated operations (such as those involving Smart Contract) require significant computational work that leads to a higher gas limit. Note that a standard ETH transfer requires a gas limit of 21,000 units of gas.

Whenever a wrong gas limit is set, the EVM will either not complete the transaction or return the unused gas to the user. By way of example, for a simple ETH transfer requiring a limit of 21,000 units:

a) If the limit set is 30,000, the EVM will consume 21,000 units and give back the remaining 9,000 to the user.

b) If the limit set is 18,000, the EVM will consume 18,000 units while attempting to complete the transaction, but won't succeed. Therefore, the EVM will revert the changes, however, the 18,000 gas units has already been consumed and won't be given back.

TRANSITION

Funded by the Erasmus+ Programme of the European Union

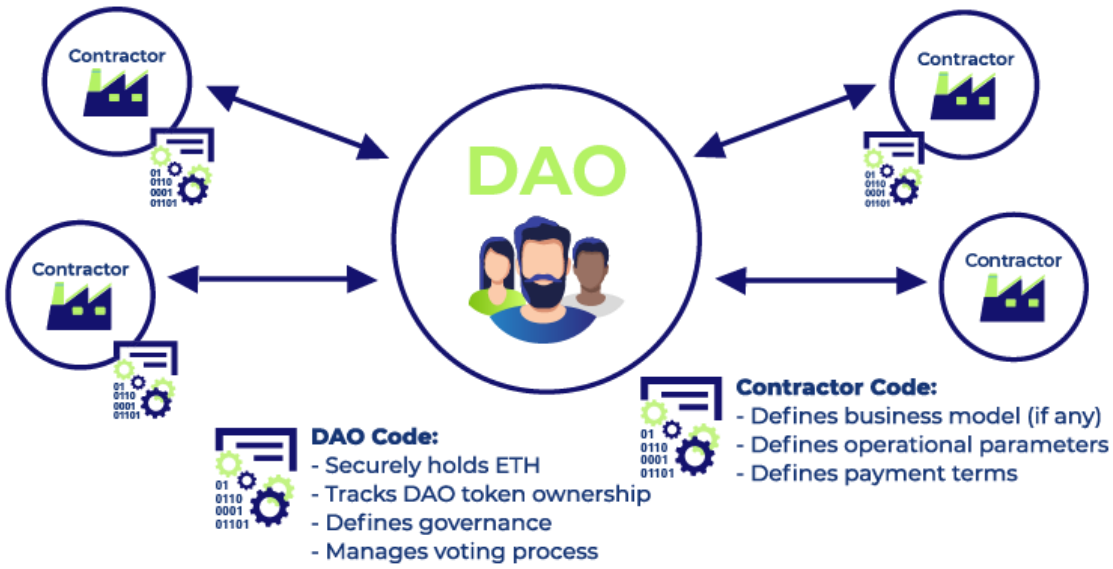
TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

5. Decentralised autonomous organisations (DAOs)

DAO (Decentalised Autonomous Organisation) is an organization governed by an informatic code and capable of working autonomously without a central authority. It is a safe way to collaborate (also with internet strangers) based on member-owned communities and an easy way to have a place to commit funds to a common project.

The rules are usually decided by a vote of the stakeholders. Generally, the way decisions are made within a DAO is through proposals. If a proposal is voted by the majority of stakeholders or meets the consensus rules of the network, it is implemented.

When starting a DAO, the user shall trust the DAO's code only, which is transparent and verifiable by everyone, thus there is no need to know and/or trust the other members.



15

The above and the specific feature of a DAO open up so many new *opportunities* for global collaboration. In fact, DAOs are fully democratized and allow to implement changes through a vote by the members. The services offered are automatically handled in a decentralized way leading to transparent and public activities.

Examples of possible uses of a DAO:

- *Charity* since DAO allows the acceptance of membership and donations from anyone, plus the group itself can decide how to spend money.

[15] https://www.yield.app/post/what-is-a-dao

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

- Creation of a *freelancer network* where the contractors collect their funds for office spaces and software subscriptions.
- Building of *ventures and grants* where to pool investment capital and votes. Moreover, money could be redistributed among DAO-members at a later stage.

The relevant Smart Contract of a DAO (deemed as its pillar) rules the organization and keeps the related treasury. As well known, once the contract is on Ethereum, nobody can change it (i.e. the rules): an amendment is immediately noticeable because everything is public. The only exception to that is making changes by voting.

The abovementioned mechanism also prevents from spending the money (treasury) without the groups' approval. As an inevitable result, the DAOs don't need a central authority: decisions are taken collectively and payments follows the votes.

TRANSITION
Funded by the Erasmus+ Programme of the European Union
TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

6. The DAO Attack

The DAO involved in the 2016 attack had a fund's value around $150 million in ether. Long story short, the hacker managed to drain 3.6 million Ether (ca. $50 million) into a personal account thanks to a chink in the DAO's code – this event was a solid proof that blockchain is not flawless.

In order to remedy the damage, three options have been proposed:

a) To do nothing, letting the stakeholders losing their investment;

b) "hard-fork", meaning restoring the blockchain as it was before the attack;

c) "soft-fork" by blocking the Ether stolen by the hacker indefinitely, in other words, freezing the hacker's account and then stealing the money back.

The option a) would have been fully compliant with the fundamental principles of the blockchain, such as: it is immutable, the code is the law and everything the code allows is legitimate. Whereas the solution b) would have represented exactly the opposite, destroying the integrity of the Ethereum blockchain.

Notwithstanding the foregoing, the development community proposed the soft fork (option c), however, the Ethereum community should have agreed on how to proceed and, therefore, the it was the one who took the final decision: the hard fork proposal has been voted and accepted by the majority. Afterwards, the solution has been completed and the funds were returned to the investors.

Though the majority of stakeholders agreed on the hard fork, not everyone did the same. As a result, two separate competing Ethereum blockchain arose:

- Ethereum Classic (ETC), the pre-forked blockchain version
- Ethereum, the blockchain that implemented the hard fork

Following the above, the DAO attack brought along a deep concern with the blockchain technology and arose serious question on this emerging technology (as well as the possible prevention measures). In fact, the strength of blockchain is that it is a ledger, a statement of truth as long as its resistance to censorship, change, demands or attack is guaranteed.

TRANSITION

Funded by the Erasmus+ Programme of the European Union

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

7. Forks

Like any software, also blockchain needs to be updated. These updates are called forks. Due to the fact that there is no central authority which decides to perform an update since the blockchain is decentralized, forks are implemented by the cryptocurrency programmers themselves or sometimes, when we talk about open-source projects it is sometimes desired that users with programming skills modify and improve the software according to their own ideas. Some forks are planned, others results of extreme situations.

Forks are currently implemented for Ethereum for its conversion in to Ethereum 2.0.

There are two types of forks:

- Hard Forks
- Soft Forks

7.1 Soft Forks

Soft forks are called soft because they don't change anything surrounding the actual structure of the protocol. Soft forks can be implemented by the developers or creators of the cryptocurrency to perform certain maintenance works, modify something cosmetically or change some of the rules surrounding the blockchain. They are a change to the protocol that is backward compatible. This means that the new rules do not exclude the protocol rules that already existed up to the time of the soft fork. As a result, all nodes are still capable of generating blocks and joining the blockchain.

Explained simply: a soft fork is the type of breakup where you remain friends with your ex. If the developers decide to fork the cryptocurrency and make the changes compatible with the old one, then the situation is called a soft fork.

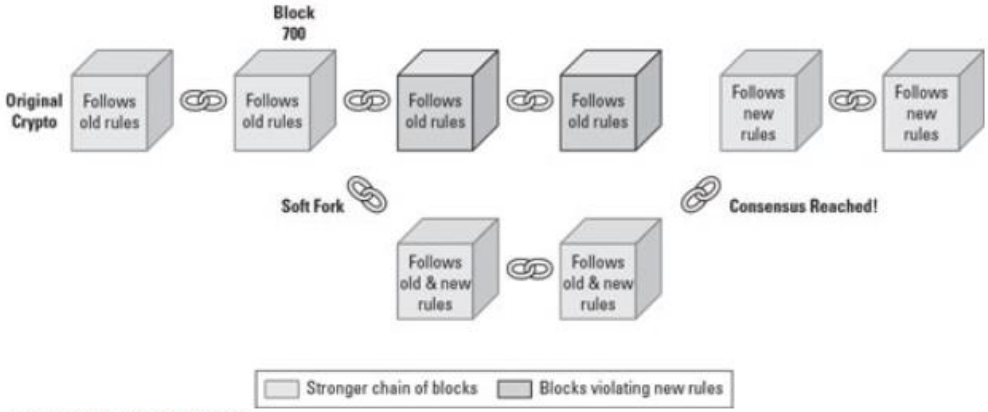
Example 1

"The protocol's already existing rules dictate that a block must reach 5MB, and then it is attached to the blockchain. However, now the protocol is updated, and the blocks are supposed to be only 3MB until they are pinned to the blockchain. Since the older nodes can append blocks with a storage capacity of 5MB, they are now also able to append blocks with 3MB to the blockchain. Thus, both old and new nodes can append blocks to the blockchain. However, if old nodes try to append a block with 5MB to the blockchain, the new nodes will be rejected because this block does not comply with the new rules of the protocol. Over time, the old nodes now also only follow the rules of the new protocol."[16]

[16] What are Soft Forks and Hard Forks? | Coinmonks (medium.com)

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

Example 2



An example of a soft fork.

"A soft fork is set to happen at block 700. The majority of the community may support the stronger chain of blocks following both the new and old rules. If the two sides reach a consensus after a while, the new rules are upgraded across the network. Any non-upgraded nodes who are still mining are essentially wasting their time. The community comes back together softly."[17]

7.2 Hard Forks

Hard forks are huge changes due to the fact that they are a change to the protocol that is not backward compatible. It is the opposite of a soft fork: here the protocol rules are changed so that the old nodes can no longer pack transactions into blocks of the new regulation since the old rules contradict the new ones. Therefore, miners have to decide whether to update their nodes to the new protocol or keep the old protocol's nodes running.

Hard forks are usually implemented under extreme conditions, so only if they are necessary; they are rarely planned. This makes sense because there are usually no legitimate reasons to implement a hard fork in a normally functioning cryptocurrency.

Ethereum is an exception.

There are two cases when a hard fork is implemented.

1. Most users perceive it as an improvement to the old protocol and will therefore join the new blockchain and continue to transact there. Hardly any participants will still follow the old protocol, and so it will die out over time.

[17] Cryptocurrency Forks or Investment Splits - dummies

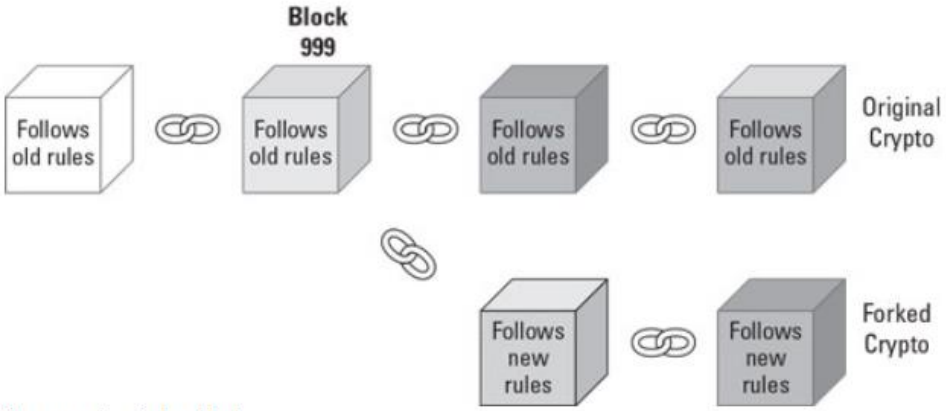
TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

2. Many users disagree on whether the update will lead to an advantage and, therefore, whether they want to join the new protocol. If the protocol is now updated, two blockchains will form from the previous blockchain. Therefore, one cryptocurrency will become two. All blocks registered on the blockchain up to that point can now be viewed on the old blockchain and the new blockchain. Now the number of coins of the old blockchain are stored on both the old and the new blockchain. However, from the time of the fork, transactions will only be published on the blockchain on which they are made.

Example 1

"Suppose the protocol's pre-existing rules prescribe 5 MB as the block size, as in the above example. However, the new protocol specifies that a data size of 7MB per block must be reached for it to be attached to the blockchain. In this case, the old nodes are not compatible with the new protocol. The miner must now decide whether to continue running his node on the old protocol or update it to the new one."[18]

Example 2



An example of a hard fork.

"The community can say that the new protocol will go live when block 999 is published to the cryptocurrency blockchain.

When the currency reaches that block number, the community splits in two. Some decide to support the original set of rules, while others support the new fork. Each group then starts adding new blocks to the fork it supports. At this point, both blockchains are incompatible with each other, and a hard fork has occurred. In a hard fork, the nodes essentially go through a contentious divorce and don't

[18] What are Soft Forks and Hard Forks? | Coinmonks (medium.com)

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

TRANSITION

Funded by the Erasmus+ Programme of the European Union

ever interact with each other again. They don't even acknowledge the nodes or transactions on the old blockchain".[19]

7.3 The Ethereum Hard Forks

It is important to mention Ethereum's Hard Forks since they are so detrimental and significant to Ethereum's well-being. There have already been several hard forks in the history of Ethereum and there are others planned for the near future.

Probably the most discussed hard fork of Ethereum was performed on July 20, 2016, necessary after the DAO attack. To ensure that the blockchain's security could be guaranteed again for everyone, a large part of the community decided to create a hard fork. Therefore, new blockchain (Ethereum) was created, and the old blockchain (Ethereum Classic) remained.

Other Ethereum Hard Forks:

- 2021

  (In Progress) London

  The London upgrade is scheduled to go live on Ethereum in August 2021, on block 12,965,000. It will introduce EIP-1559, which reforms the transaction fee market, along with changes to how gas refunds are handled and the Ice Age schedule.

  (In Progress) Altair

  The Altair upgrade is the first scheduled upgrade for the Beacon Chain. It is expected to go live in 2021. It will add support for "sync committees", which can enable light clients, and will bring inactivity and slashing penalties up to their full values.

  Berlin (15.04.2021 Block 12,244,000)

  The Berlin upgrade optimized gas cost for certain EVM actions and increases support for multiple transaction types.

- 2020

  Beacon Chain genesis (01.12.2020 Block number 1)

  The Beacon Chain needed 16384 deposits of 32 staked ETH to ship securely. This happened on November 27, meaning the Beacon Chain started producing blocks on December 1, 2020. This is an important first step in achieving the Eth2 vision.

  Staking deposit contract deployed (14.10.2020 Block number: 11052984)

[19] Cryptocurrency Forks or Investment Splits - dummies

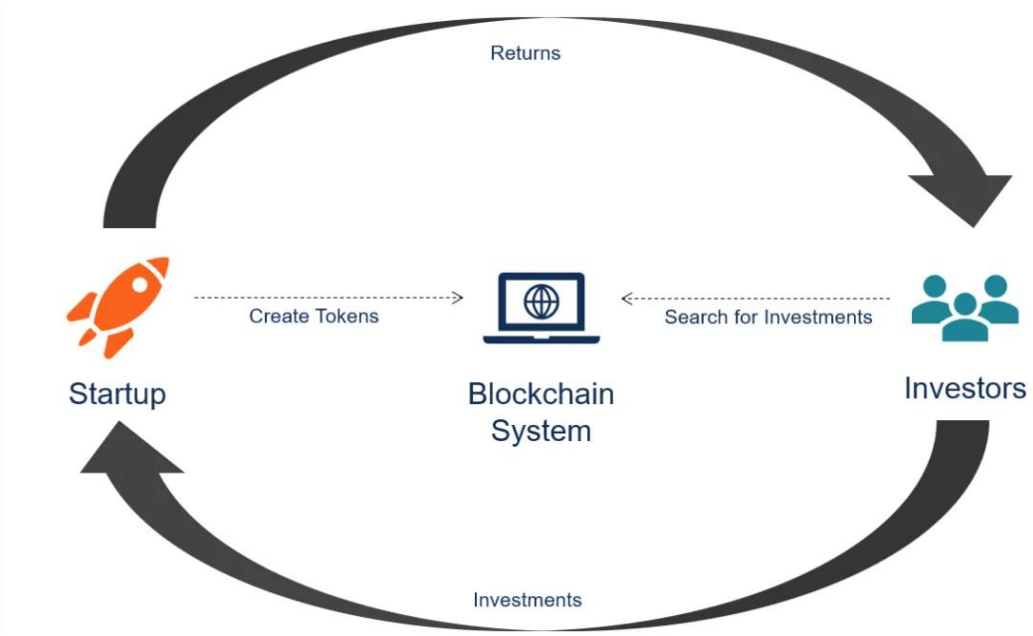
TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

The staking deposit contract introduced staking to the Ethereum ecosystem. Although a mainnet contract, it had a direct impact on the timeline for launching the Beacon Chain, an important Eth2 upgrade.

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

8. Initial Coin Offering (ICOs)

An initial coin offering (ICO) is a type of capital-raising activity in the cryptocurrency and blockchain environment. The ICO can be viewed as an initial public offering (IPO) that uses cryptocurrencies. However, it is not the most precise comparison, as there are some crucial differences between the two fundraising activities. Startups primarily use an ICO to raise capital.



20

IPO	ICO
IPO are a form of crowdfunding which targets the general public. The public offering is a democratized form of investing due to the fact that almost anyone can become an investor. However, due to regulatory concerns, private ICOs are becoming a more viable option with regards to public offerings.	In private ICO, only a limited number of investors can participate in the process. Generally, only accredited investors (financial institutions and high net-worth individuals) can participate in private ICOs, and a company can choose to set a minimum investment amount.
Initial Public Offering	Initial Coin Offering
Issues shares of a company	Issues cryptocurrency tokens
Regulated by Authorities	Unregulated
Centralized by stock exchange	Decentralised

[20] https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/initial-coin-offering-ico/

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

8.1 How does an ICO work?

21

1. Identification of investment targets and creation of the White paper.

Every ICO starts with the company's intention to raise capital, therefore the company needs to identify the targets for its fundraising campaign and creates the relevant materials about the company or project for potential investors. A professional whitepaper outlines what is sold during the ICO - usually tokens - and its assigned value, the total amount of capital needed and the terms of the contract. A whitepaper contains project details such as the offering, outlining the benefits for investors, a solid road map of the project and a timeframe for when the project is expected to yield earnings to investors.

2. Promotion campaign

A company runs a promotion campaign to attract potential investors. Campaigns are commonly executed online to achieve the widest investor reach. However, currently, several large online platforms such as Facebook and Google ban the advertising of ICOs.

3. Creation of tokens and token sale

Tokens are representations of an asset or utility in the blockchain. They are fungible and tradeable and should not be confused with cryptocurrencies because they are just modifications of existing cryptocurrencies. Tokens are created using specified blockchain platforms. This process is quite simple because a company is not required to write the code from scratch as in the creation of new cryptocurrency: existing blockchain platforms that run existing cryptocurrencies such as Ethereum allow the creation of the tokens with minor modifications of the code.

After the creation of the tokens, they are offered to the investors. The offering may be structured in several rounds.

[21] What are ICOs and IEOs in blockchain space? — Bitpanda Academy

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

4. Company or Project issues tokens

The company can then use the proceeds from the ICO to launch a new product or service while the investors can expect to use the acquired tokens to benefit from this product/service or wait for the appreciation of the tokens' value. Tokens generally do not provide an equity stake in a company. But most of them deliver their owners some stake in a product or service created by the company.

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

GLOSSARY

Account: an account is an entity with an Ether balance that can send transactions on Ethereum.

Blockchain: public database, updated and shared within many computers among the network. The blockchain is the technology behind Ethereum: the digital record of transaction sets up a chain of blocks. Note that block's data cannot be amended/changed without changing all subsequent blocks, such process would require the consensus of the entire network.

C++: is an object-oriented computer language developed by Bjorne Stroustrop. It is part of the evolution of the C family of languages. C++ is pronounced "see-plus-plus."

Client: application on the computer able to "run" a node. Widely used Clients are Geth and OpenEthereum.

Cryptographic code: it is used to convert ordinary plain text into unintelligible text; a piece of information (e.g. letter, word, phrase) is substituted with another object, but not necessarily with the same type.

DAO (decentralised autonomous organisation): a sort of community, implying an effective/safe way to work with people around the globe and manage treasury for a specific scope.

Dapp: Decentralised application that uses blockchain platforms and the relevant P2P networks.

Digital signature: it is a mathematical technique used to validate the authenticity and integrity of a digital document a software of a message; it binds a person/entity to the digital data. This binding can be independently verified by receiver and any third party.

EVM bytecode: is a low-level programming language which is compiled from a high-level programming language such as solidity. It is not readable for humans but readable for the machine.

Ether: cryptocurrency of Ethereum.

Ethereum Account: a user's public and private key pair.

Ethereum Address: it is based on the public key generate by the wallet, is what get sent to Dapps and is what money/tokens are sent to.

Ethereum Virtual Machine (EVM): global virtual computer. Every participant on the Ethereum network stores and agrees on its state. Moreover, participants can demand the execution of arbitrary code on the EVM, keeping in mind that code execution changes the state of the EVM.

Gas: essential element to the Ethereum network since it is the fuel that allows it to operate.

Key: number that identifies the user. there are two type of keys: public and private. The former proves the ownership of an account and helps to prove that a transaction is actually signed by the sender. The latter is used by others to verify the sender by checking the digital signature made with relevant private key.

TRANSITION

Funded by the Erasmus+ Programme of the European Union

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

Node: Every computer in the network is known as "node". Each node has the same data and, due to the “consensus” mechanism, new blocks and chain shall be agreed upon by every node (computer) of the network. Nodes are therefore the real-life machines which store the EVM state. They communicate with each other to pass on information about the EVM state and new state changes.

Opcodes: Also called operation codes are numeric codes which contain the instruction that represent the operation which needs to be performed.

Peer-to-peer (P2P) platform: Platform that allows users to interact without going through an intermediary.

Proof-of-Work consensus: Consensus mechanism (process through which reaching an agreement about information on the network) currently used by Ethereum.

State: it is what is representable by a computer at a given moment. The state is an enormous data structure which keeps all accounts linked by hashes and reducible to a single root hash stored on the blockchain.

Token: A type of digital asset that may be exchanged within a blockchain. A token is used as a representation of assets (material and digital) or rights (ownership to an asset or access to a service).

Fungible token (interchangeable tokens): like voting tokens, staking tokens or virtual currencies.

Non-fungible token: like a deed for artwork or a song.

Transaction: Cryptographically signed instructions from accounts that may either result in message calls or in contract creation.

Turing-complete virtual machine: a machine that, given enough time and memory along with the necessary instructions, can solve any computational problem.

Wallet: A product that lets you manage your Ethereum account: view your account balance, send transactions, and more. The most popular Ethereum wallet is Metamask.

TRANSITION

Funded by the Erasmus+ Programme of the European Union

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

BIBLIOGRAPHY

Bibliography paragraph 1

- https://ethereum.org/en/history/#whitepaper
- https://coinmarketcap.com/alexandria/glossary/ethereum-virtual-machine-evm
- https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf
- https://ethereum.org/en/developers/docs/intro-to-ethereum/
- https://www.bitdegree.org/learn/ethereum-virtual-machine
- https://ethereum.org/en/developers/docs/nodes-and-clients/
- https://ethereum.org/en/wallets/
- https://ethereum.org/en/nft/
- https://medium.com/@markmuskardin/mastering-the-fundamentals-of-ethereum-for-new-blockchain-devs-part-iii-wallets-keys-and-4cd3175b535b
- https://ethereum.org/en/developers/docs/evm/
- https://cointelegraph.com/ethereum-for-beginners/ethereum-wallets
- https://medium.com/@roberto.g.infante/smart-contracts-the-brain-of-dapps-cf886bb1bbb6
- https://www.gemini.com/cryptopedia/blockchain-types-pow-pos-private#section-blockchain-types
- https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/

Bibliography paragraph 3

- https://ethereum.org/en/developers/docs/dapps/
- https://ethereum.org/en/dapps/
- https://www.bitpanda.com/academy/en/lessons/what-is-a-dapp/

Bibliography paragraph 4

- https://ethereum.org/en/developers/docs/dapps/
- https://www.bitdegree.org/learn/ethereum-virtual-machine
- https://ethereum.org/en/developers/docs/gas/

TRANSITION

Funded by the Erasmus+ Programme of the European Union

TRANSITION – Project Reference: 2019-1-MT01-KA202-051255
This project is funded by the European Union
ERASMUS+ Program – Key Action 2
Cooperation for innovation and the exchange of good practices

- https://medium.com/@roberto.g.infante/smart-contracts-the-brain-of-dapps-cf886bb1bbb6

Bibliography paragraph 5

- https://ethereum.org/en/developers/docs/dapps/
- https://ethereum.org/en/dao/
- https://www.yield.app/post/what-is-a-dao

Bibliography paragraph 6

- https://www.nytimes.com/2016/06/18/business/dealbook/hacker-may-have-removed-more-than-50-million-from-experimental-cybercurrency-project.html
- https://www2.deloitte.com/ie/en/pages/technology/articles/DAO-Attack-Analysis.html
- https://www.planetcompliance.com/nutshell-dao-steal-60-million-worth-cryptocurrency/
- https://ogucluturk.medium.com/the-dao-hack-explained-unfortunate-take-off-of-smart-contracts-2bd8c8db3562
- https://www.nytimes.com/2016/06/18/business/dealbook/hacker-may-have-removed-more-than-50-million-from-experimental-cybercurrency-project.html
- https://www.coindesk.com/understanding-dao-hack-journalists
- https://www.gemini.com/cryptopedia/the-dao-hack-makerdao#section-the-dao-hack