Smart Contracts

Marvin Koppka, Mustafa Erdogan

October 1, 2022

Abstract

Blockchain is currently one of the most prominent technologies. Whilst being used in the form of DeFi, NFTs, Web3 as well as logistic and supply chains these use cases stem from a decentralized ledger realized by chains of blocks consisting of transactions, hence the name Blockchain.Smart contracts are used in order to enable secure transactions without the need for third parties together with further automatized applications. Despite the recent uprise due to NFTs these technologies are however neither always reasonably applicable nor secure. In scope of this article a quick overview of the blockchain will be given, the workflow explained and smart contracts as well as security issues discussed.

- *Marvin*

1 Blockchain - Overview

Starting with the technology used as a basis for various other technologies such as Smart Contracts, Decentralized Finance (DeFi) and Non-Fungible Tokens (NFTs), Blockchains. While DeFi as a use case wont be discussed in this blogpost, Smart Contracts (Chapter 3) and NFTs (Chapter 3.2) will be covered in according chapters. Since Blockchains are the basis lets start with explaining blockchains. Blockchains are decentralized data structures consisting of blocks containing transactions, ommers and a block header (figure 1). Each block is chained to its prior block, hence the name blockchain. But what exactly is a blockchain?

The technology known as Blockchain is a combination of multiple technologies such as peer-to-peer Networks, consensus algorithms, hashing algorithms, as well as multiple data structures such as the chain of Blocks and merkle trees which will be introduced in the following. Thereby an overview will be given, whilst the workflow of Ethereum will be further emphasized on in Chapter 2.

1.1 Chain of Blocks

1.1.1 Blockchain - Structure

At first lets introduce the structure of a blockchain, thus the chaining of blocks. So what exactly is a block? Blocks of a Blockchain do vary in its structure depending on the Blockchain. As we are mainly focussing on Smart Contracts in this post we will discuss blocks for Ethereum. Blockchains exists on computers, also called nodes, forming a huge interconnected network (figure 5) with various benefits, which will be discussed in chapter 1.2. Each computer thereby tries to add new block to the existing blockchain and share information in this network. Adding blocks yields rewards and sustains the blockchain. For Ethereum these Blocks consists of a Block Header, a list of Transactions and an Ommers List (figure 1). The Transaction list consists of transactions from a temporary memory pool, basically a waiting area for transactions, a node in the network includes into a candidate block, a block trying to be added to the blockchain (figure 2). When the node is successful this candidate block gets appended to the Blockchain. Sometimes multiple blocks are ready to be added to the chain at the same time of which only one will be primarily used for the Blockchain. These other blocks are known as the Ommers or Uncles and yield a partial reward for adding blocks to the blockchain. Now we know what a block is and that blocks are appended to the blockchain by nodes in the network. But how are blocks chained?

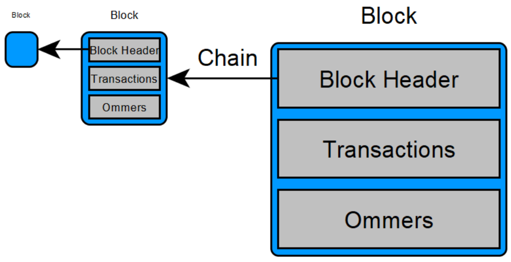


Figure 1: Block structure

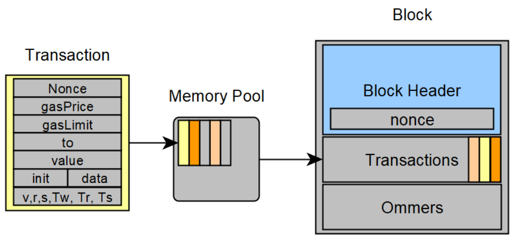


Figure 2: Transactions added to a block

1.1.2 Metadata of Blocks - Block Header

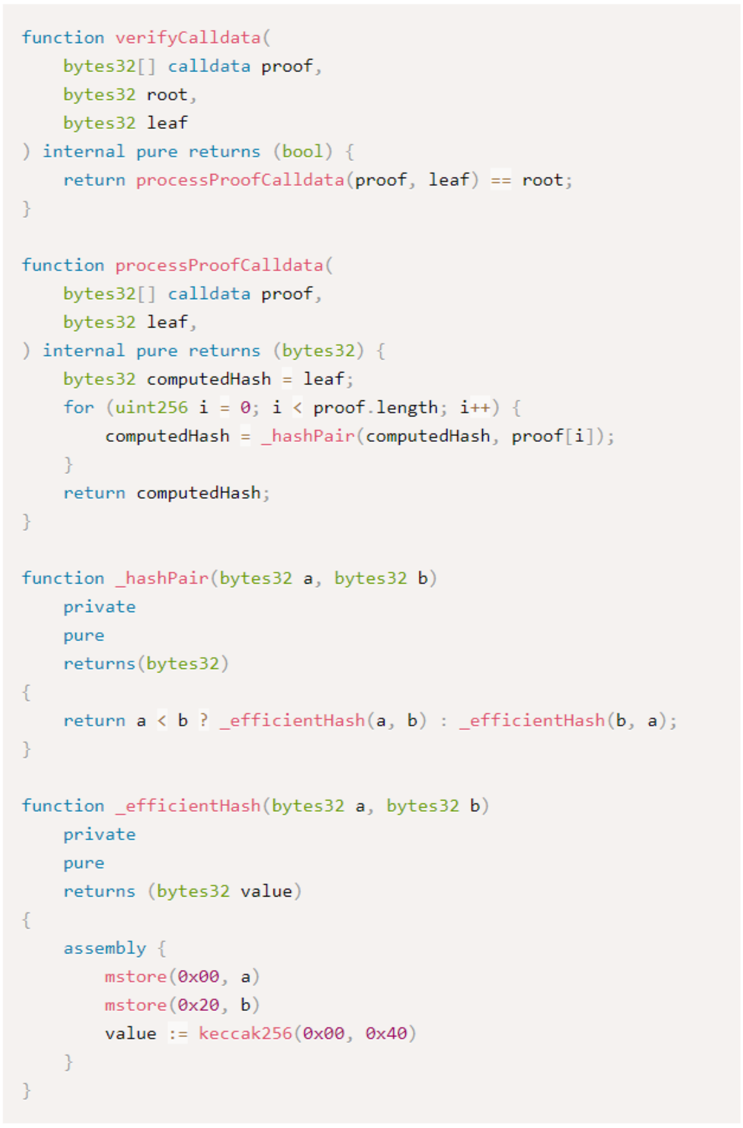
The Block Header is the most essential part of chaining blocks. For this chaining, which will now be referred to as hashing, a digital fingerprint of a block is derived by hashing its contents. An example of a hash function would be the modular operation, we have an input which can be any value and a divisor for the modular operation m. The output of this function would then take values in the range of 0 to (m-1), basically mapping all input values to this range. Now in order to avoid collisions of these mapping we choose the hash function to be complex and have a significantly higher range of mapping values.

Hashing blocks is done in order to notice and avoid changes in the blockchain. Therefore all blocks are hashed in a similiar fashion and the resulting hash used for the next block. If a block was tempered with the hash of the block and those following change accordingly, which due to the process of adding blocks and hashes of previous blocks would lead to an invalid blockchain. New blocks with valid hashes would have to be created and added to the blockchain until the previously non-tempered blockchain would have less blocks in order for the tempered one to be accepted.

Besides the hash value of the previous block the block header consists of multiple other data structures such as merkle trees, namely the World State Trie, Account storage content trie, Transaction Trie and Transaction Receipts Trie, which will be discussed in Chapter 2 and 3.

1.1.3 Merkle Tree

A Merkle tree also known as hash tree, is a tree in which each leaf node is labeled with the cryptographic hash of a data block, and each non-leaf node is labeled with the cryptographic hash of its child nodes' labels.Implementations of hash tree mostly are binary (each node has two child nodes) but they could also have more child nodes.Merkle trees which is aso known as Binary hash trees, are a common type of data structure in computer science. In bitcoin and other cryptocurrencies, They are used in bitcoin and other cryptocurrencies to more effectively and securely encrypt blockchain data.. It's a hash based mathematical data structure made up of hashes of various data blocks that summarize all the transactions in a block. The consistency and quality of the data are also verified, as well as quick and secure content verification across large databases.



```
function verifyCalldata(
    bytes32[] calldata proof,
    bytes32 root,
    bytes32 leaf
) internal pure returns (bool) {
    return processProofCalldata(proof, leaf) == root;
}

function processProofCalldata(
    bytes32[] calldata proof,
    bytes32 leaf,
) internal pure returns (bytes32) {
    bytes32 computedHash = leaf;
    for (uint256 i = 0; i < proof.length; i++) {
        computedHash = _hashPair(computedHash, proof[i]);
    }
    return computedHash;
}

function _hashPair(bytes32 a, bytes32 b)
    private
    pure
    returns(bytes32)
{
    return a < b ? _efficientHash(a, b) : _efficientHash(b, a);
}

function _efficientHash(bytes32 a, bytes32 b)
    private
    pure
    returns (bytes32 value)
{
    assembly {
        mstore(0x00, a)
        mstore(0x20, b)
        value := keccak256(0x00, 0x40)
    }
}
```

Figure 3: MerkleTree (8)

verifyCalldata function takes in the proof itself as bytes32 array, the Merkle root hash and the leaf we want to verify for inclusion. Simply, the root is going to be data you store in the smart contract and proof is going to be the data from anyone created off-chain, for proving to the contract which the leaf was part of the original tree.

Now, processProofCalldata function is going to iterate through each element in the proof array.

We start by taking the leaf node. After that, in every step we update our computed hash by hashing it with the next element in the proof. Hashing both hashes together are going to always take the smaller value first. Openzeppelin is used assembly with the keccak256 opcode for more efficient hashing. Alternatively one could use Solidity: keccak256. We return back the computed hash.

Then we go back in verifyCalldata function we verify that the computed hash matches the expected root hash.

1.1.4 Encryption Mechanisms

As cryptography and encryption mechanisms are an important topic for blockchains we will cover those briefly. "Cryptography can, for example, also be used to prove knowledge of a secret without revealing that secret (e.g., with a digital signature), or to prove the authenticity of data (e.g., with digital fingerprints, also known as "hashes"). These types of cryptographic proofs are mathematical tools critical to the operation of the Ethereum platform (and, indeed, all Blockchain systems), and are also extensively used in Ethereum applications." (23)

In blockchains each Block is hashed with specific hashing algorithms. For Ethereum these encryption Algorithms are Keccak-256 (7) and the elliptic curve algorithm Secp256k1 (3) which are used in order to safely encrypt data to prevent frauds. Elliptic curves are used to derive digital signatures to ensure that a transaction is valid and the user initiating it owns the correpsonding assets. Keccak-256 is a one way mapping of data used as a fingerprint to ensure the authenticity as well as for consesus algorithm in order to guarantee the computational effort used for the creation of blocks. Therefore blocks need to be hashed below a certain target value, due to the nature of one way mappings this procedure needs to be brute forced leading to high computational effort.

We wont go into too much detail here, but as can be seen in figure 4 elliptic curves are used with private and public key pairs. A private key is a random 256 bit key created by the user from which a public key is derived. The address of the account the user creates is furthermore derived from the public key.

Next chapter the second technology necessary for blockchains is discussed, peer-to-peer Networks.

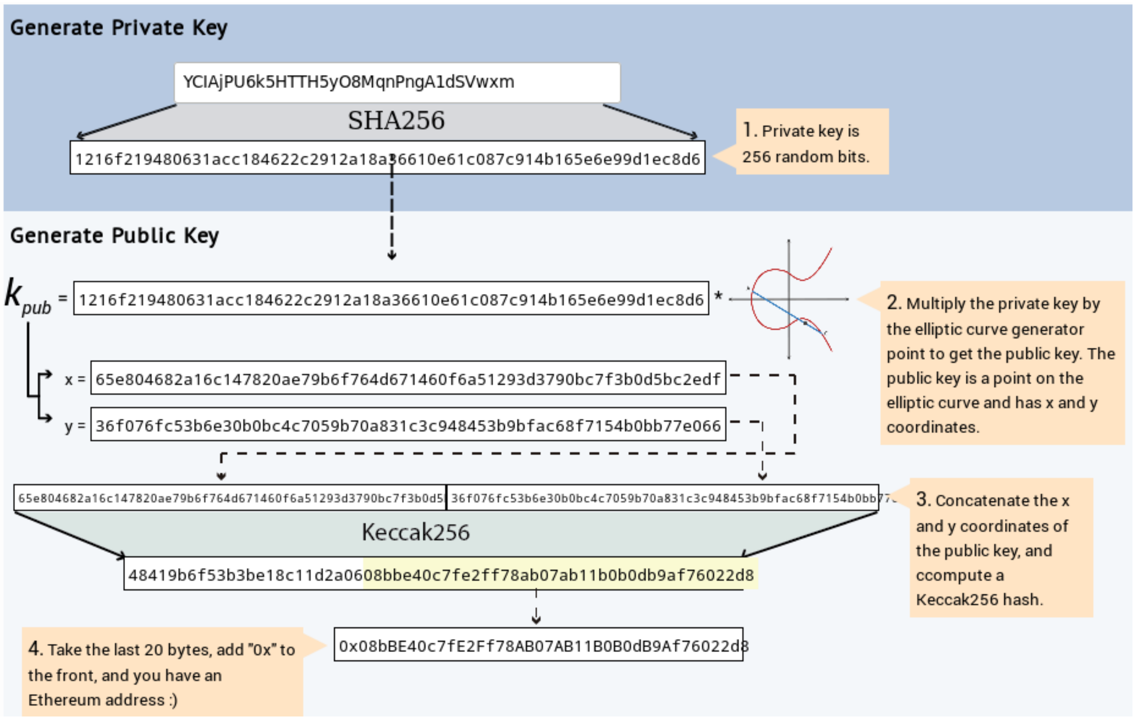

Generate Private Key
YCIAjPU6k5HTTH5yO8MqnPngA1dSVwxm
SHA256
1216f219480631acc184622c2912a18a36610e61c087c914b165e6e99d1ec8d6
1. Private key is 256 random bits.
Generate Public Key
k_{pub} = 1216f219480631acc184622c2912a18a36610e61c087c914b165e6e99d1ec8d6 *
2. Multiply the private key by the elliptic curve generator point to get the public key. The public key is a point on the elliptic curve and has x and y coordinates.
x = 65e804682a16c147820ae79b6f764d671460f6a51293d3790bc7f3b0d5bc2edf
y = 36f076fc53b6e30b0bc4c7059b70a831c3c948453b9bfac68f7154b0bb77e066
65e804682a16c147820ae79b6f764d671460f6a51293d3790bc7f3b0d5
36f076fc53b6e30b0bc4c7059b70a831c3c948453b9bfac68f7154b0bb7
3. Concatenate the x and y coordinates of the public key, and ccompute a Keccak256 hash.
Keccak256
48419b6f53b3be18c11d2a0608bbe40c7fe2ff78ab07ab11b0b0db9af76022d8
4. Take the last 20 bytes, add "0x" to the front, and you have an Ethereum address :)
0x08bBE40c7fE2Ff78AB07AB11B0B0dB9Af76022d8

Figure 4: Private Key, Public Key and Address Generation (20)

1.2 Peer-to-Peer Networks

Blockchains are upheld, shared and expanded in a decentralized fashion, the use of a peer-to-peer network consisting of nodes (figure 5). Nodes in the network thereby refer to computers running instances of a given Blockchain client software forming connections to other computers (2). Nodes upheld and expand the blockchain by adding blocks to their local copy of the existing blockchain and sharing newly found blocks with others. But nodes do not only propagate transactions and blocks among the network but also validate data in order to achieve consensus, thus strive for a single state of the blockchain. In order to interact with this decentralized data structure one needs to interact with its nodes, by either being a node or interact with one by the use of wallets or accounts (18). A wallet can send transactions to a node which then adds the transaction if verified to a temporary pool of transactions called the memory pool. This transaction gets propagated over the network enabling every node to include this transaction in the next block. A fee for the transaction hereby indirectly orders the transactions, as nodes are more likely to include transaction into the, to be added, canditate block if the fees they receive are higher. Nodes hereby add their own address as first transaction, in the case of Bitcoin called coinbase transaction and referred to as beneficiary address for Ethereum, in order to collect rewards for adding a valid block to the blockchain. Earlier the term “consensus” was mentioned. What exactly is consesus and how is it achieved?

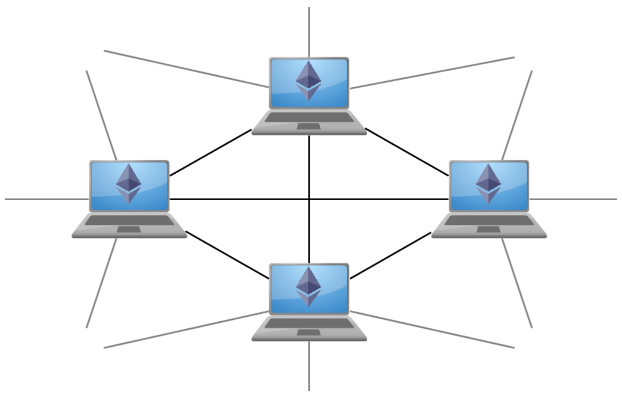


Figure 5: Peer to Peer Network

1.3 Mining: Consesus among the Network

Consesus algorithms exist in order to realize consesus among the network, thus agree upon a single state of the Blockchain. Each transaction gets approved before adding it to the memory pools by adhering the Blockchains rules and confirming the ownership of the currency for the transaction as well as validated after a given block has been found by other nodes receiving said block. Thus leading to adherance of single states and avoiding erronous ones. Additions of blocks to the Blockchain are thereby controlled by the consesus algorithm in use. Different algorithms exist, such as proof of work and proof of stake utilizing different mechanisms. Proof of work entails adaptable complex calculations, whilst proof of stake entails staking of currency.

Proof of Work describes the mechanism of providing work in the form of solving complex calculations. The block header of the candidate block therefore gets hashed together with an iterative value called nonce (number used once). The result of the hash needs to be below a specific target value chosen by the network and adapted relative to the computing power in the network (figure 8). If a fitting nonce value leading to a hash below the given target was found the block is valid and can be added and shared among the network. It will then be executed, verified and added by each node. Afterwards a new candidate block is put together, with new transactions and the cycle goes on. With PoW nodes are in this process normally referred to as miners. Usually miners however do not mine for themselves, as it is very unlikely for a single miner to solve the next calculation first. Instead to average the success rate miners do mine together and share the reward, called mining pools.

In the following chapter these concepts will be underlined by emphasizing on the workflow of the Ethereum Blockchain. Questions not yet answered are, what is Ethereum and why was this blockchain developed?

- *Marvin*

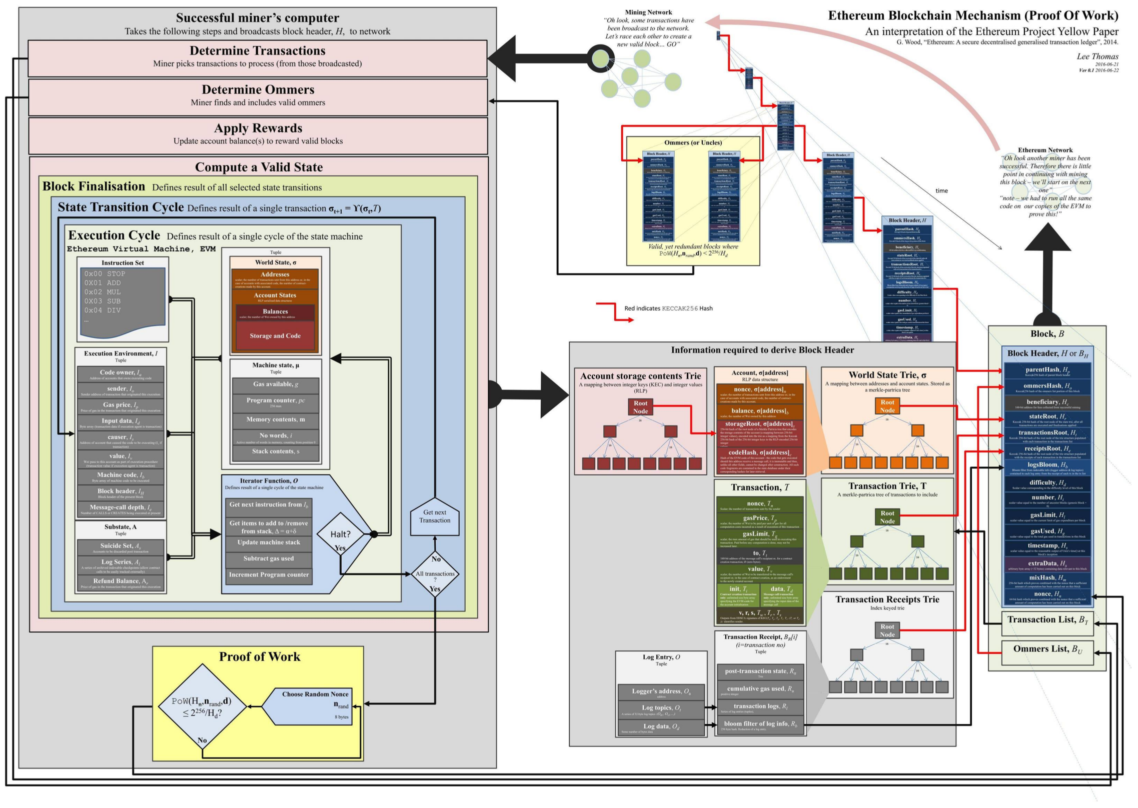
2 Ethereum:

2.1 Introduction:

Ethereum is a decentralized community driven open-source Blockchain created for various purposes such as Decentralized Finance (DeFi), more transparent supply chains, healthcare and smart contracts. With its special structure Ethereum saves additional information such as Transactions Receipts and the resulting state calculated by the Ethereum virtual machine (EVM) in multiple data structures called patricia merkle tries.

Due to additional information given by the Tries, such as the state, light clients - nodes that do not have a full copy of the blockchain - are not just able to prove inclusion of transactions but also verify given information about states. While this information would normally result in the verification of every single transaction in the entire chain, this can be apprehended by saving these states. As such, transactions in the Blockchain are more easily verifiable for certain kinds of queries (9). Additionally Smart Contracts enable automation of contracts effectively replacing the need for third parties, opening up the field of Initial Coin Offerings (ICOs) and creation of Tokens, such as non-fungible tokens (NFTs).

2.2 Workflow:



Successful miner's computer
Takes the following steps and broadcasts block header, H, to network
Determine Transactions
Miner picks transactions to process (from those broadcasted)
Determine Ommers
Miner finds and includes valid ommers
Apply Rewards
Update account balance(s) to reward valid blocks
Compute a Valid State
Block Finalisation
Defines result of all selected state transitions
State Transition Cycle
Execution Cycle
Defines result of a single cycle of the state machine
Ethereum Virtual Machine, EVM
Instruction Set
Execution Environment, I
World State, σ
Machine state, μ
Iterator Function, O
Substate, A
Halt?
Get next Transaction
All transactions?
Proof of Work
Choose Random Nonce
Mining Network
Ommers (or Uncles)
Ethereum Blockchain Mechanism (Proof Of Work)
An interpretation of the Ethereum Project Yellow Paper
Lee Thomas
Ethereum Network
Red indicates KECCAK256 Hash
Information required to derive Block Header
Account storage contents Trie
World State Trie, σ
Transaction, T
Transaction Trie, T
Transaction Receipts Trie
Transaction Receipt
Log Entry, O
Block, B
Block Header, H or B_H
Transaction List, B_T
Ommers List, B_U

Figure 6: Workflow of the Ethereum Blockchain based on the Yellowpaper (4) by Lee Thomas (16)

In the following the workflow of Ethereum will be explained by emphasizing on the given figure 6. Specific components of each data structure can thereby be inferred from the figure and hashes with Keccak-256 are indicated by red lines. So how does the workflow of Ethereum look like?

2.2.1 Mining Network:

Starting with the topside of the figure - the Mining Network. As previously discussed the network consists of several computers called nodes forming a peer-to-peer network. Each of these nodes run a blockchain client for access to the blockchain, albeit there are differences in nodes. A node can either be a archive, full or light node (2). Transactions added to the blockchain stem from nodes or wallets interacting with nodes.

These transactions get added to a temporary memory pool, which is simply a pool of transactions waiting to be added into the earlier discussed candidate block. Each node has its own memory pool, new transactions thereby are propagated along the network for nodes to add transactions to their local memory pool. What happens with these transactions?

2.2.2 Block Finalisation

For transactions to be included in the blockchain they need to be included in a block (left side of the figure 6 - Determine Transactions). The finalization of the block thereby involves four stages which as can be seen in the Yellowpaper (4) (11. Block Finalisation) vary depending on whether the block is finalized for mining or has already been mined and needs to be validated. In our case ommers and transactions need to be determined and added to the block. The first transaction in the list of transactions thereby is the application of rewards, the beneficiary address for miners to receive rewards. After this the process of mining start which is the fourth stage, computing a valid state and block nonce. But before explaining the process of consesus algorithms lets first exphasize on what a block looks like, which data structures are used and how transactions are added to the network.

2.2.3 Accounts and Blocks

As shown in figure 1, 2 and 6 blocks consist of a Block header, transactions and an ommers list. Transactions can thereby be added by Ethereum accounts, namely Externally owned accounts (EOA) and contract accounts. Accounts as previously discussed are generated from a random 256 private key, whilst the corresponding public key is derive with elliptic curve algorithms and addresses for the account are then again derived from the public key. Externally owned Accounts are essentially accounts created and controlled by users or basically controlled by anyone who holds the corresponding private key. While EOAs can receive, hold, send Ether or tokens and interact with deployed smart contract on their own, contract accounts can only send transactiontzhs as a reaction of receiving transactions. Thus EOAs can trigger transactions of contract accounts initiated by transactions, albeit contract accounts can also trigger transactions of other contract accounts if the first initiation was by an EOA. Contract accounts are smart contracts deployed to the network, that are controlled by code. Smart contracts will be emphasized on in Chapter 3

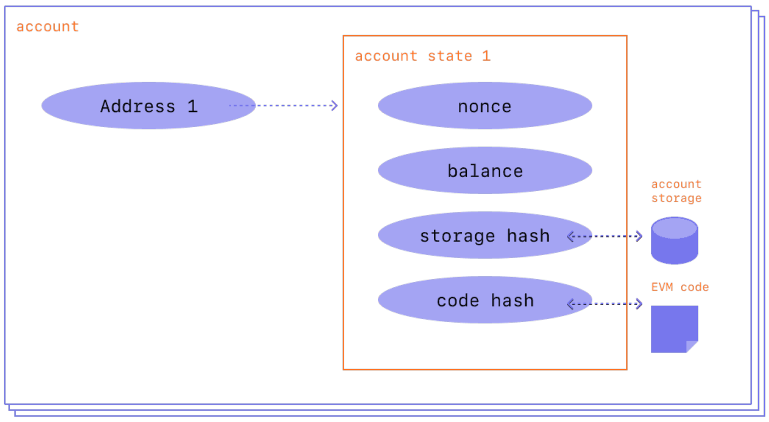


Figure 7: Ethereum - Accounts

As depicted in figure 7 each account has four fields, the nonce, balance, storage hash and code hash equal to figure 6 (right side of the figure). The code hash field is a hash of the EVM code of the account, which is executed if this address receives a message call. For EOAs this code is the hash of an empty string (4). Code of accounts are stored in the state database. The storage root is "a 256-bit hash of the root node of a patricia merkle tree that encodes the storage contents of the account", introducing the data structure called patricia merkle trees that are important for the block header, in which the root hashes of these trees are stored. Without going into too much detail, a merkle tree is an efficient data strucuture for key-value mappings that enables fast and secure validation of the given data. Due to the structure of the tree alterations in the tree yield different hash values in nodes above the alteration, such as the root node. Because of this property Ethereum light clients reap the earlier mentioned benefits. Ethereum however does not only use this single merkle tree, it also stores the root hashes of three others, namely the World State, Transaction and Transaction Receipts Trie. The world state trie is a "mapping between addresses and account states" whereas each leaf node represents an account state. The account state therefore refers to the given state in time, thus the specific balances, nonce values and the hashes for the storage root and code at any given time. Specific changes in time, thus the transition of the state due to the transaction is stored in the transaction receipts trie, while the transaction trie is derived from the transaction list.

Now we know what a block looks like, but how is such a block added to the blockchain?

2.2.4 Consensus Algorithms

As previously mentioned the process for additions of blocks to the blockchain depend on the consensus algorithm in use. At the time of writing Ethereum still uses the Proof of Work consesus protocol Ethash, which will for the sake of completeness and issues as a result of Proof of Work be covered. Consensus algorithms are used in order for the network to agree on a single state, this consensus can thereby be upheld in different ways.

How can this consensus be upheld? What are the approaches? Although there are multiple consensus algorithms the two we will introduce in the following are again Proof of Work (PoW) and Proof of Stake (PoS).

Proof of Work:
For PoW nodes in the peer-to-peer network are referred to as miners. To recap, each miner builds a candidate block from pending transactions in the temporary memory pool. The miners can choose their own transactions they want to add into the candidate block and then start with the mining process. Mining describes the process of hashing the canditate block until a valid hash could be found. For a candidate block to be valid, the hash of the block needs to be below a certain target value. This target value adapts with the computational power existing in the network and is updated in intervals. As shown in figure 8 a value called number used once (nonce) in the block header is changed in an iterative way until a valid hash value is found. Due to several resistances of Keccak-256 (figure 13) mining is done by brute force, thus changing the nonce value until the resulting hash is valid. The newly found valid block is propagated along the network.

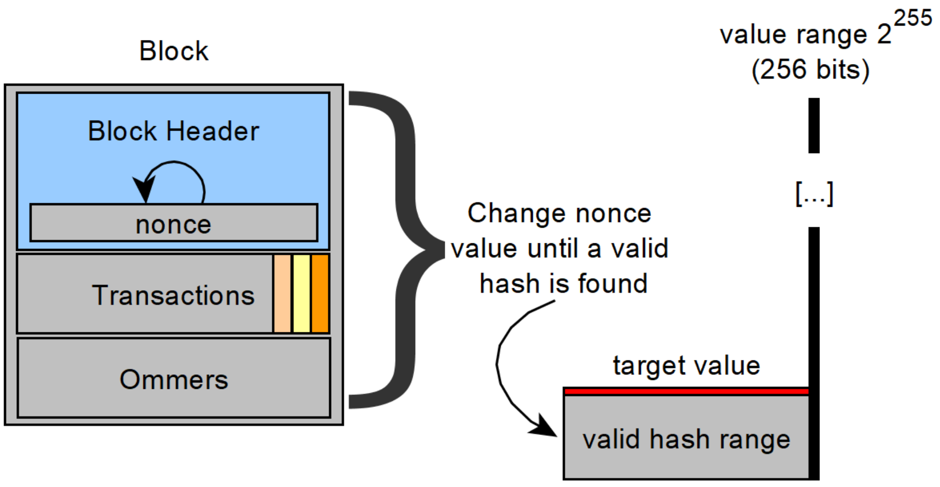

Block
Block Header
nonce
Transactions
Ommers
Change nonce value until a valid hash is found
value range 2^{255} (256 bits)
[...]
target value
valid hash range

Figure 8: Finding a valid hash below the target value by adapting the nonce value

Due to consensus among the network, the combined computational effort leads to immensely increased safety as well as reducing the likelyness of attacks such as the double spending attack in which the same currency is spent multiple times by tempering with the Blockchain. In order for nodes to find a valid hash for candidate blocks, in the case of PoW, high computational effort is necessary. As the name implies this computational effort represents a Proof of Work put into the adherence of the block added to the chain. As only the first valid block is appended everyone else's computational effort is neither credited nor meaningful, with the exception of ommers, leading to a huge energy consumption. For this reason and a major speedup, generated by the absence of safety formerly achieved by slow additions of blocks, Ethereum will change to Proof of Stake, on the 15th of September known as "The Merge" (1).

Proof of Stake:
With PoS nodes are referred to as validators minting or forging blocks. Each validator thereby stakes coins acting as security deposit by reducing said coins in case of the approbation of fraudulent blocks by the validator. As a result building trust in validators as stakes are higher than benefits gained by approving wrong transactions. The chance to be chosen as validator increases with the size of the staked coins. A chosen node validates transactions in the block and adds it to the blockchain for which it is rewarded with the fees of the included transactions. Fees as well as staked coins are released when a node discontinues being a validator, albeit the release is delayed in order to still deduct coins if necessary.

Due to PoS, the energy consumption of Ethereum was reduced by 99.95% and the time necessary for adding block could be significantly reduced (1). Furthermore PoS decreases the need for high hash rates, thus expensive and huge amounts of hardware. Additionally it might yield increased decentralization as mining in pools is suspended together with the reduced necessity of expensive hardware, yielding more distributed nodes contributing to the network. While majority attacks are still possible, the benefit is outweighed by the loss of the 51% staked coins necessary, together with possible fluctuations of the coin as a result of such an attack.

The block has now been added to the blockchain via PoW or PoS and is propagated along the network for other nodes to be validated and included into their copy of the blockchain. As stated in the yellowpaper (4) and the left side of the figure 6 the block is being finalized in four stages. As mentioned earlier depending on whether the block is being mined or propagated and validated the stages slightly vary. Compared to mining, where the ommers and transactions were determined, the reward applied and a valid state and nonce computed with consensus algorithms, the valid block was shared among the network and is bein validated and verified. Each node in the network therefore validates the ommers, transactions and applies the rewards and verifies the state and nonce of the block.

But how are these states being verified?

2.2.5 Ethereum Network

The verification of the state includes running the code on the Ethereum virtual machine (EVM), the backbone for computing states. Important to note is, that every node goes through that step of validating and verifying mentioned earlier. As a state machine the EVM basically computes the state of the blockchain after each transaction, thus applying changes in the fields of accounts involved in the transaction. After every transaction and the transition in states is applied the state transition cycle is concluded and the final state derived from it. At this point every transaction and the states are validated and verified. The cycle then repeats and new transactions are chosen from the transaction pool for the next candidate block. Since we only covered the EVM briefly check (17) for more information, albeit this in-depth knowledge is normally not required for grasping the concepts of Ethereum.

After this overview of Ethereum an important aspect is still fairly untouched, being use-cases. Most of these use-cases which are more native to Ethereum are acquired by the use of Smart Contracts, such as NFTs, ICOs and more. As such the next Chapter will introduce Smart Contracts.

- *Marvin*

3 Smart Contracts

3.1 What Are Smart Contracts And Advantages?

- Smart contracts is a type of code that allows for verification and execution of a contract easily and automatically.These contracts work on Blockchain system instead of server.Smart contracts are defined rules and penalties depending on agreement and automatically enforce those obligations.Also,many smart contracts could implement together.Even though they work independently.

  When the predetermined conditions verify and reach,computer systems execute the set actions.Actions such as transferring funds to appropriate parties, sending notifications are completed the blockchain is updated. Moreover, like those cases, only the parties are permitted to see the transaction results.Consequently,there is no third-party interference.

  As required,contracts could set with more than one stipulations, with focusing on each participant's satisfaction and competent task execution. Members require to determine how their transactions and execution data should be represented on the blockchain by agreeing on the rules that control the transactions.

  In that case, the developer could program and customize smart contracts in the chain. For example, the Ethereum allows developers to access the EVM which is Ethereum Virtual Machine for execution code for smart contracts. Moreover, the organizations which implement blockchain for businesses offer a web templates, and other online tools that help them form a basic structure for smart contracts.

  To mention about advantages of smart contracts,firstly,transparency is first key advantage of smart contracts.The participants have the same information in the mean time, that to minimise possible manipulation of the stipulation of the contract. While smart contracts are based on blockchain, they guarantee the immutability of data, allowing contracts and agreements to make without the need to know each other and preventing potential breaches of conditions or errors in the management and execution of the contract.

  Since transactions are duplicated so that all parties involved have record,this transparency provides the parties with security and trust because they always have access to the data and information to relate to the contract is available to them throughout the life of the contract.

Secondly,autonomy is another advantage of contracts that provide trusted third parties or human intervention, which offers autonomy and independence to the parties in the process.This feature of smart contracts brings cost reduction and process speed advantages.Third advantage is speed.Because of absence of intermediaries reduces the cost economically as same as the time cost. Compared to contracts that are manually executed and in the presence of a third party, the time commitment is lower because it is completed automatically.So for understanding NFT's better we will focus on Merkle Tree.Why is used for on NFT?

3.2 Ownership

NFTs come in many forms but the most typical one is a metadata file that contains information that has been encoded with a digital version of the work that is being tokenized. The second type involves uploading the complete piece of work to the blockchain;these are less typical as it is expensive to upload information to the blockchain.A piece of code that is written on blockchain is most typical sort of NFT. There are many different bits of information in that code.The ERC-721 standard for NFTs specifies elements that must be present, but some that are optional.The tokenID, a number that is generated when token is created, is the first fundamental component of NFT.The contract adress, a blockchain address that can be seen anywhere in the globe using a blockchain scanner is the second fundemantal component.Only one token in the world exists with that combination of TokenID and contract adress,making the token unique due to combination of its component.These two numbers are the NFT's only fundamental components.Other significant components,however,may also be included in the contract.One is the creator's wallet address,which links the NFT back to its originator.Yet,how we know originality of NFT?

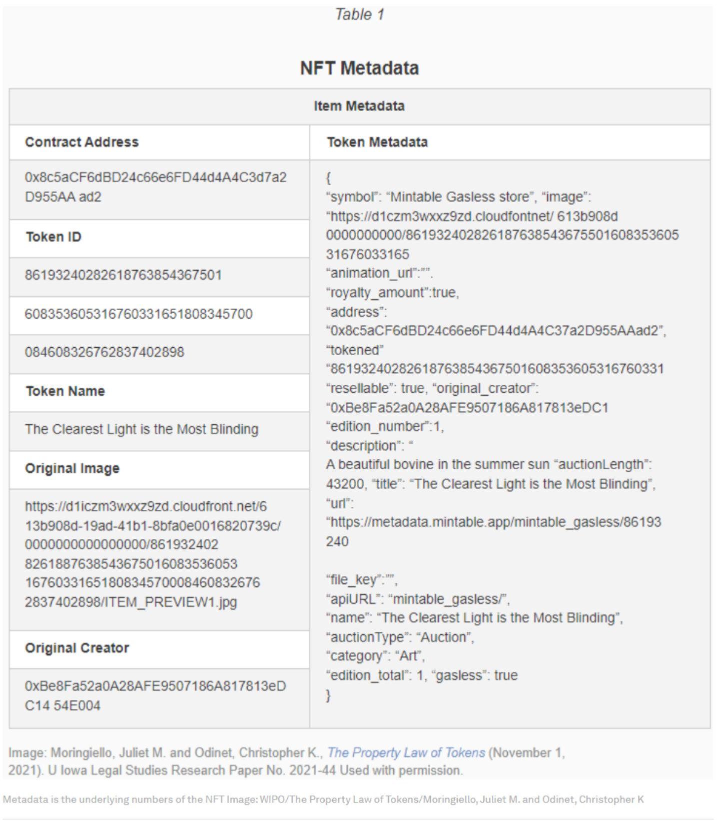


Table 1

NFT Metadata

Item Metadata	
Contract Address	**Token Metadata**

0x8c5aCF6dBD24c66e6FD44d4A4C3d7a2 D955AA ad2	{
	"symbol": "Mintable Gasless store", "image":
	"https://d1czm3wxxz9zd.cloudfontnet/ 613b908d 0000000000/861932402826187638543675501608353605 31676033165
	"animation_url":"".
	"royalty_amount":true,
	"address":
	"0x8c5aCF6dBD24c66e6FD44d4A4C37a2D955AAad2",
	"tokened"
	"861932402826187638543675016083536053167603 31
	"resellable": true, "original_creator":
	"0xBe8Fa52a0A28AFE9507186A817813eDC1
	"edition_number":1,
	"description": "
	A beautiful bovine in the summer sun "auctionLength":
	43200, "title": "The Clearest Light is the Most Blinding",
	"url":
	"https://metadata.mintable.app/mintable_gasless/86193 240
	
	"file_key":"",
	"apiURL": "mintable_gasless/",
	"name": "The Clearest Light is the Most Blinding",
	"auctionType": "Auction",
	"category": "Art",
	"edition_total": 1, "gasless": true
	}

Token ID	
861932402826187638543675O1	
608353605316760331651808345700	
08460832676283740 2898	
Token Name	
The Clearest Light is the Most Blinding	
Original Image	
https://d1iczm3wxxz9zd.cloudfront.net/6 13b908d-19ad-41b1-8bfa0e0016820739c/ 0000000000000000/861932402 826188763854367501608353605 3 1676033165180834570008460832676 2837402898/ITEM_PREVIEW1.jpg	
Original Creator	
0xBe8Fa52a0A28AFE9507186A817813eD C14 54E004	

Image: Moringiello, Juliet M. and Odinet, Christopher K., *The Property Law of Tokens* (November 1, 2021). U Iowa Legal Studies Research Paper No. 2021-44 Used with permission.

Metadata is the underlying numbers of the NFT Image: WIPO/The Property Law of Tokens/Moringiello, Juliet M. and Odinet, Christopher K

Figure 9: Metadata (10)

3.2.1 Copyright

Most of NFTs are a metadata files that have been encoded with works that may or may not be covered by copyright protection (you could theoretically create an NFT of a trademark), or it could even be a work that are in the public domain.An NFT may be created from anything that can be converted into digital format;the original work is only required at the initial phase of the process to create the unique combination of the tokenID and the contract address.Therefore,in theory,copyright has very little to do with NFTs.

Moreover, there is rising interest in NFTs from a copyright perspective.This is partly due to the fact that many works being exchanged as NFTs,such as works of art,are covered by copyright,but it is also unclear about what it is exactly that you get when you buy and NFT. One of the main issues is the sometimes prevalent misunderstanding of the the rights that buyers acquire when they purchase an NFT. Some buyers believe they acquire the underlying work of art together with all of its rights,They are actually only purchaising metadata related to the work not the work itself. However, copyright may well be relavant, at least for some NFTs. For example, one possible usage of these tokens might be in some sort of digital rights management scheme. While most NFTs do not involve a transfer of rights, the seller occasionally offers to turn the token into an actual transfer of copyright ownership of the original work. However, it is challenging to determine if this is complies with the formalities required by law to transfer copyright.Now,we will explain limitations of contract.What are the limits of contracts?

3.3 Limitations of Smart Contracts

3.3.1 Unrevisability

It is hard to change the way a smart contract processes and fixing a code errors may be time consuming and expensive.

3.3.2 Crack

The idea of good faith says that parties will deal fairly and refrain from obtaining unethical gains from a contract. However, it is challenging to guarantee that the terms are followed in accordance with what was agreed upon when utilizing smart contracts.

3.3.3 Third Party Assessment

Even while smart contracts aim to do away with third parties, it is impossible to do so. In contrast to the functions they play in conventional contracts, third parties take on new responsibilities. For instance, attorneys won't be required to draft individual contracts, but developers will still need them to comprehend the conditions when writing the programs for smart contracts.

3.3.4 Uncertainity Terms

Smart contracts can't always manage unclear terms and conditions since contracts often include phrases that aren't always clear.

After explanation of limitations now,we will go over functionality of smart contracts.What is it?

3.4 Functionality of Smart Contract

"Smart contracts work by following simple "if/when... then..." statements that are written into code on a blockchain. A network of computers executes the actions when predetermined conditions have been met and verified. These actions could include releasing funds to the appropriate parties, registering a vehicle, sending notifications, or issuing a ticket. The blockchain is then updated when the transaction is completed. That means the transaction cannot be changed, and only parties who have been granted permission can see the results."(5)

Currently, Ethereum has the highest level of popularity among other smart contract platforms,several other cryptocurrency blockhains,including as Tron,Polkadot,Avalanche are also capable of supporting them.Anyone can create a smart contract and deploy to a blockchain.There are several programming language used to create contracts(including Solidity,Web Assembly).Each smart contract on the Ethereum network has its code stored on the blockchain,to allow anybody with an interest to check the contract's code and current state of contract to verify its functionality.

Moreover,along with the blockchain and transaction data,every computer on the network(or "node") keeps a copy of all active smart contracts and their current state.All nodes in the network execute a smart contract's code when it receives funds from a user in order to reach a consensus about the outcome and value flow that results.Due to this,smart contracts may run safely without the need for a centralized authority,even when users conduct complex financial transactions with unidentified entities.On the Ethereum network, you often have to pay a charge called "gas" to execute a smart contract(so named because these fees keep the blockchain runnig). Smart contracts are often unchangeable once they deployed onto a blockchain,even by their creator.in2016,by Saturday, June 18th, the attacker had succeeded in draining more than 3.6 million ether into a "child DAO" that shares The DAO's architecture. Ether's cost decreased from over $20tojustunder13$. To stop the theft of further ether, a number of users attempted to divide The DAO, but they were unable to secure the requisite votes in time. All of the ether was in a single address since the creators didn't anticipate this much money, which was a stupid decision. We assume the attacker stopped after learning about the fork proposal (see below). In actuality, that attack or another one like it may continue at any moment. It's highly plausible that the assailant was holding a sizable short position on ether at the time of the attack, which he or she paid out when the price of ether had approximately doubled. Despite the fact that there is ether in the child DAO, the attacker could have have made his money. The Ethereum Foundation has options that might potentially render the ether in this DAO useless.Now, at this point emphasizing to standards of ERC is important.

3.5 ERC Standards

"ERC is essentially an acronym for Ethereum Request for Comments. In general, ERCs are specifications for Ethereum applications, such as token standards, name registries, library formats, and package formats. An author may create an Ethereum Blockchain app with an ERC token, but they will need to clarify their standard and gain community approval." (12)

Applications and smart contracts can communicate with tokens predictably according to standardized ERC standards.The most common ERC standard is ERC-20 right now.It makes it easier to develop,use,and trade Ethereum-based tokens.

3.5.1 What Are Standards?

Ethereum Request for Comments(ERC) is a document that smart cotract programmers use to write smart contracts on the Ethereum blockchain platform.These documents specify the requirements that Etherum-based tokens must meet. To be familiar with the basics of Ethereum,the ultimate Ethereum guide should be checked out.The Ethereum developer and community utilize this procedure,which is known as the Ethereum Improvement Proposal,to examine these papers.The document's author may then make revisions as a consequence of their feedback. Multiple purposes are provided by a variety of ERC standards.Following is the one of common standard in terms of transfers and balances.

- ERC-20:The typical API for fungible tokens, which includes functionality for tracking transfers and balances.
- ERC-721: The most widely used non-fungible token (NFT) standard.Non-fungible tokens can not be split up,however fungible tokens can. NFTs may be owned and transacted by one person or they may be assigned to another party.NFTs can signify ownership over digital or physical assets

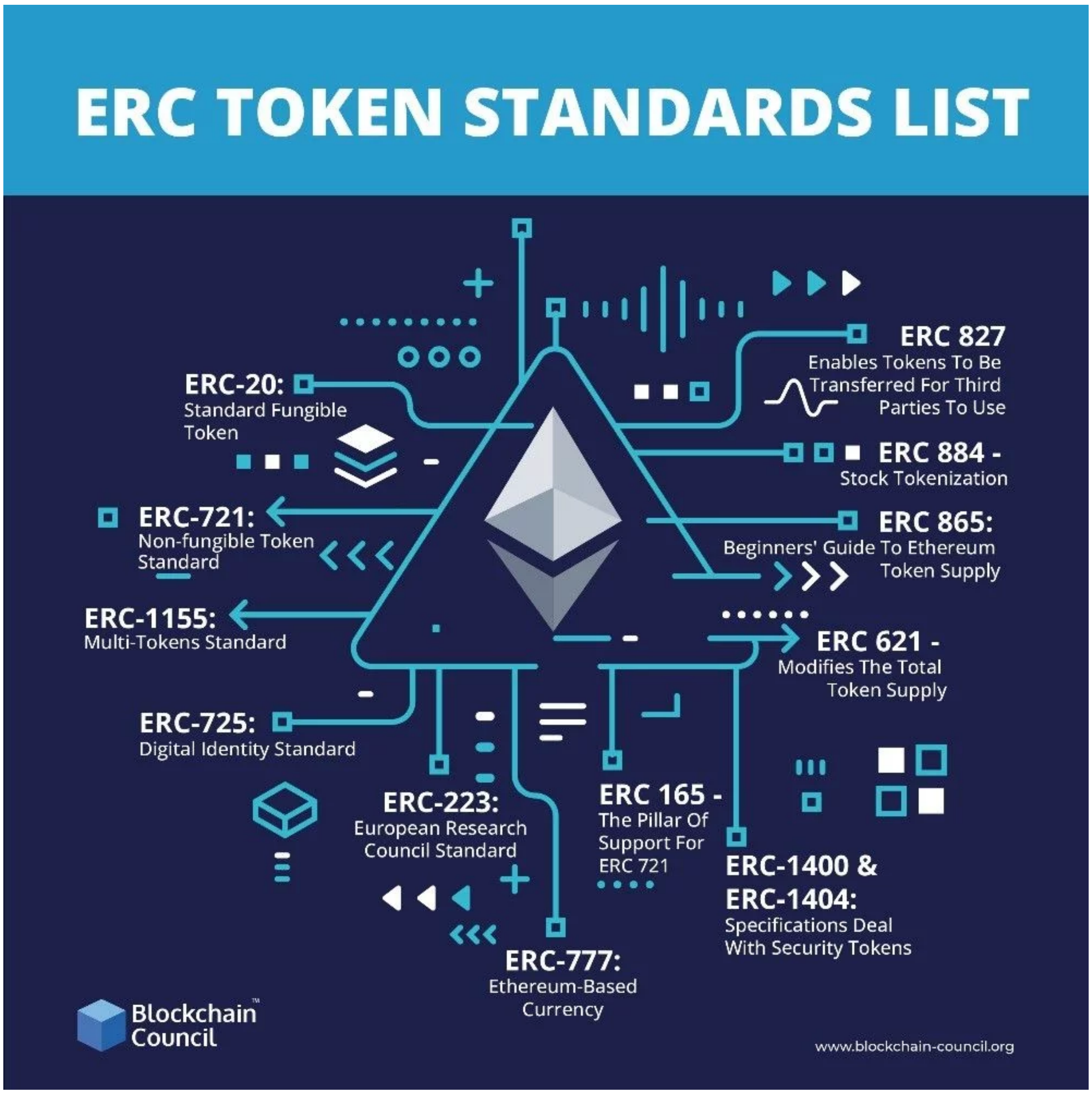

ERC TOKEN STANDARDS LIST
ERC-20: Standard Fungible Token
ERC-721: Non-fungible Token Standard
ERC-1155: Multi-Tokens Standard
ERC-725: Digital Identity Standard
ERC-223: European Research Council Standard
ERC-777: Ethereum-Based Currency
ERC 165 - The Pillar Of Support For ERC 721
ERC-1400 & ERC-1404: Specifications Deal With Security Tokens
ERC 621 - Modifies The Total Token Supply
ERC 865: Beginners' Guide To Ethereum Token Supply
ERC 884 - Stock Tokenization
ERC 827 Enables Tokens To Be Transferred For Third Parties To Use
Blockchain Council
www.blockchain-council.org

Figure 10: Standards List (13)

3.6 Reentrancy Attacks

– Reentrancy Attacks: It is occured between two smart contracts and it is one of the most destructive attacks in Solidity that we have been using for smart contracts.If explain basically,one of function makes an external call to untrusted contract.Later,untrusted contract makes recursive call back to original contract to drain funds.Those attacks are most well known attacks on blockchain because of 2016 DAO hack on Ethereum chain.The hack works by having attacking smart contract continiously works and calling the withdraw function before no updating time to update balance of the vulnerable smart contract.

  It is only possible if the smart contract is developed to handle transactions in such a way that it first verifies its balance against another smart contract that is vulnerable, then sends the cash, updates its balance, and finally updates its balance. A new window opens up between transferring the funds and updating the balance, allowing an attacking smart contract to make another call to withdraw its funds. This cycle continues until all of the funds have been used.But for giving idea that what would cause about reentrancy attack is on September 2021 AMM pools hacked and it caused 3.5 million dollar worth damage to pools owner and explosion was through reentrancy attack.

  Last but not least,Keccak algorithm should be explain because It is important while using solidity and it is a cryptographic function.What is Keccal algorithm in other words hash function?

- Mustafa

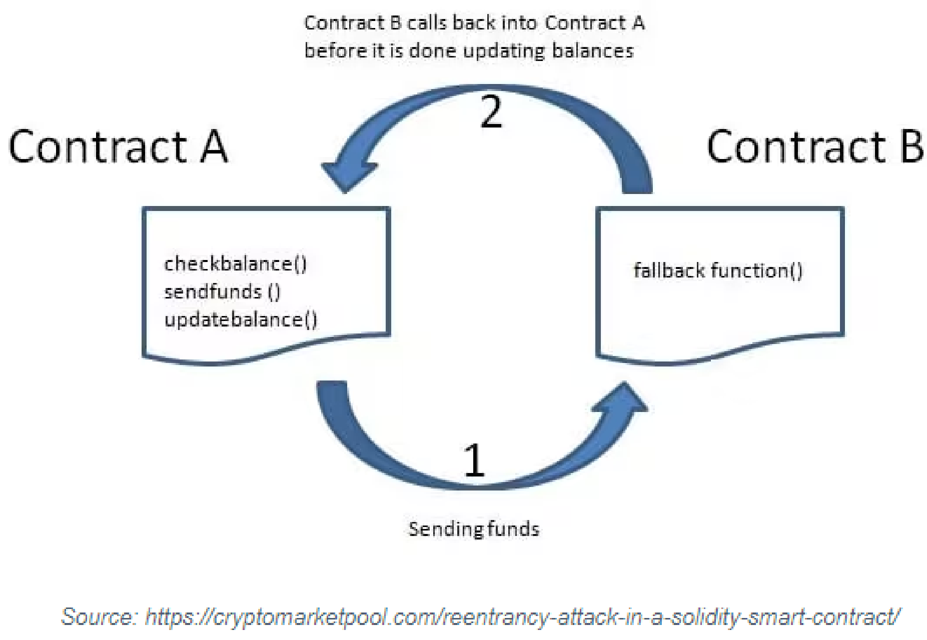

Contract B calls back into Contract A before it is done updating balances
2
Contract A
Contract B
checkbalance()
sendfunds ()
updatebalance()
fallback function()
1
Sending funds
Source: https://cryptomarketpool.com/reentrancy-attack-in-a-solidity-smart-contract/

Figure 11: Reentrancy Attack (11)

- Mustafa

4 Risks and Issues:

4.1 Introduction:

The goal when Satoshi Nakamoto created the first cryptocurrency in 2009 was to establish a decentralized payment system that would completely change the way we conduct business.The goal of Bitcoin,according to Nakamoto's white paper, was to faciliate quick and borderless transactions.

Although not in manner Nakamoto desired,Bitcoin has become mainstream more than ten years later.Cryptocurrencies now serve as speculative assets rather than transaction tools,attracting investors who think they may profitably sell their holdings in future.Therefore,a consequence of the demand for bitcoin is high electricty use. The estimated yearly electricty usage of Bitcoin is 127 terawatt-hours (TWh). This is greater than Norway's whole annual electricity use.It is a fact that , Bitcoin needs electricty more than Ethereum does,roughly 11 times as much (707 kilowatt-hours (kWh) every transaction. Bitcoin's proof of work consensus process,which requires computers to perform difficult math problem to valide transactions,consumes more energy than most people realize. This method didn't use a lot of power in the early days of Bitcoin. However, when more people compete to answer the math riddles, it will become considerably tougher due to the fundamental nature of bitcoin technology. 1.7 times as many transactions per second (TPS) as Bitcoin's 4 TPS. According to a May 2022 analysis from the Cambridge Digital Assets Program, the U.S. owns the majority of the worldwide Bitcoin mining business with almost 38 percentage of the hashrate recovery (CDAP). Contrarily, CDAP also discovered that China, with more than 20 percentage of the worldwide market share, is the second-largest Bitcoin mining center, despite Beijing's crackdown on the practice, which aims to eradicate it within its borders. Moreover,In **figure** 2, there is comparison between VISA and Bitcoin that how much energy consumed ?

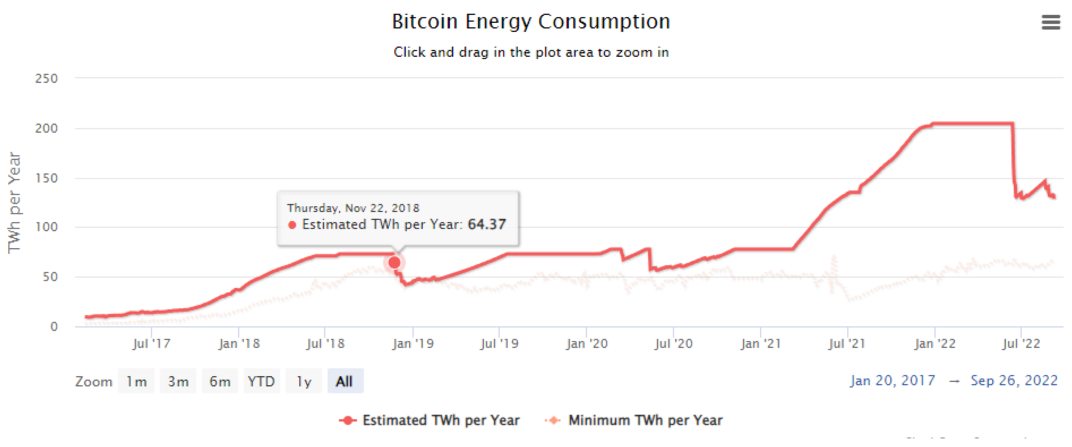


Figure 12: Energy Consumption

- *Mustafa*

Due to “The Merge” of Ethereum, thus the change from PoW to PoS the energy consumption was reduced by 99.95% (1) resulting in a breaking point for blockchain, as no running blockchain has previously changed its consesus algorithm. If more blockchains were to adapt to PoS the energy consumption and need for hardware could be further reduced.

4.2 Decentralisation Issues:

Another issue is the longevity of specific Blockchains as well as impacts of majority attacks (51% attacks). Due to the structure of Blockchains and the usage of peer-to-peer networks in order to uphold it and achieve consesus computing power and users are essential. Without users the Blockchain could neither be sustained nor reasonably agreed upon a single state, as a result majority attacks could be enforced easier than on big Blockchains, such as Ethereum or Bitcoin in which 51% computational power would be hard to achieve by single individuals. Blockchains are therefore highly dependent on its users. Currently rewards and fees are an incentive for miners to produce new blocks, this however, will not be always the case, as some cryptocurrencies do aim for a maximum quantity of its currency, in which case fees remain as the sole incentive which is likely less attractive for miners.

Furthermore not only the lack of decentralization due to reducing numbers of miners poses an issue, also the combination of miners forming mining pools, thus centralization, imposes the threat of majority attacks. In some cases only two or three mining pools would have to combine in order to achieve 51% mining power which could then in return alter the Blockchain, albeit limited, to their needs (22). In addition to decentralization issues Blockchains face further issues, especially security risks, such as Reentrancy attacks, hashing algorithms and On- and Off-Chain Storage.

Besides security risks, Blockchain and especially Smart Contracts do face and cause some issues, namely the absence of a severability clause and impacts on society. The absence of the severability clause refers to a clause which is added to complex contracts, which prevents mistakes in the contract to fully invalidate it. This however happens to Smart Contracts. If a Smart Contract is invalid it cannot be changed anymore, even though it might entail crucial bugs. As such the Blockchain gets flooded with immutable, unoptimized and partially faulty code. The ERC Standards mentioned in 3.2 however do prevent crucial bugs from happening, such as Reentrancy attacks, which lead to the Dao Hack on the Ethereum Blockchain in 2016.While explained issues of decentralization what are these attacks?

- *Marvin*

4.2.1 Keccak Algorithm

According to the Keccak Reference (6) and Keccak-256 Submission (7), Keccak-256 is a hash function based on a sponge construction which takes any size of input and generates a chosen size of output. Its safety is thereby regulated "By translating these computation complexities into physical quantities such as time or energy, both are simply out of reach and will remain so in the foreseeable future" (7). A private key is made of 256 random bits. An address is made of 160 bits (figure 4). To bring the unlikeliness of a collission of either a private key or even an address into perspective lets do some math. First off, due to the collission and preimage resistances of hash functions (figure 13) two values mapping to the same hash can not be found easily and neither can hashes for certain pre defined outputs of the hash significantly increasing the security of hash algorithms. Currently 7.97 billion people exist on this planet which is approximately equivalent to $2^{33} \approx 8.59 * 10^9$. If all people on planet earth would generate one million private keys per day for 10000 years it would amount to $3 * 10^{22}$ with $2^{256} \approx 1 * 10^{77}$ and $2^{160} \approx 1.46 * 10^{48}$. In fact it would take about $4.66 * 10^{29}$ years to generate a collision of an address. As private keys have even more bits the years necessary are astonishingly high. Before conclusion but in next subsection we will see storage of NFT's.How an NFT store and differences of On-Off Chain Storage?

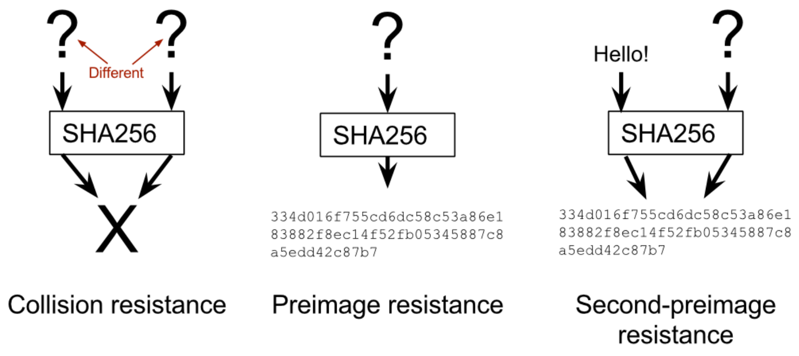

?
?
Different
SHA256
X
Collision resistance
?
SHA256
334d016f755cd6dc58c53a86e1 83882f8ec14f52fb05345887c8 a5edd42c87b7
Preimage resistance
Hello!
?
SHA256
334d016f755cd6dc58c53a86e1 83882f8ec14f52fb05345887c8 a5edd42c87b7
Second-preimage resistance

Figure 13: Resistances when hashing (19)

- *Marvin*

4.3 NFT On-Chain and Off-Chain Storage

NFT Generation Using Neural Networks Another rising trend of today's world is the NFTs (Non-fungible Tokens). NFTs are a trending technology supported by a protocol called ERC-721 (15). The most common use case of NFTs is to create and distribute art pieces that carries information like *owner history* on a blockchain and is known to be duplicate-proof.

NFTs are a form of digital media.Moreover,all NFTs are not same which means that are created and stored differently.

On-Chain NFTs

- On Chain NFTs are tokens that are written on Blockchain(Ethereum),implementation of metadata and smart contract both are on blockchain.Basically,the information of these On Chain NFTs are on the mainnet and then stored on blockchain. This information also includes the transaction hash of the generated NFT, which makes NFT more unique.

The core information of NFTs are metadata.Such as the NFT's unique traits and where the digital copy stores,description, and so on. This information is on live on blockchain.However,this metadata of NFT is integrated with itself.

What Are Off-Chain NFTs?

Some NFT projects store their contracts on the blockchain in order to reduce gas costs, while their meta data is off the chain. Less gas fees would be needed to keep NFT meta's that are stored off the chain. When NFT meta data is stored off-chain, a transaction for that NFT takes place, but only the smart contract data is called up because it is not connected to the blockchain.

Some projects keep the meta data for their NFTs on the cloud using services like Drive, Dropbox, and AWS.

The “interplanetary file system” (IPFS) nodes provide additional storage for meta data.

It is a web data protocol designed to address issues with HTTP and FTP.

Some NFT Projects run their own IPFS nodes, and those have a lot of items in them.

Even with the IPFS technology, a problem still exists right now. The NFT file can still be deleted by the NFT creator at any time, severing the connection between the file and the blockchain on which the NFT is based. Your ownership of an off-chain NFT is still recorded on the blockchain even if the NFT file is removed after you buy it, but you will no longer have access to it. It would be similar like owning something that was once real but is now gone.

- Mustafa

Another issue concerned with NFTs are impersonation attacks, also called sleepminting (21), in which a NFT is being minted to a different address than the transaction senders’. The creator of the NFT is therefore displayed as this address, confusing buyers, as certain creators do have a higher innate value to their creations, such as CryptoPunks. If bought, the previous transaction sender initiates another transaction for the NFT to be transferred to the buyer and receives the funds.

Evidently some issues and risks for this “decentralized payment platform that would revolutionize the way we pay everything” (14) still persists for Blockchains and resulting technologies. Former issues of banks were partially solved, but new issues were created. As such existing issues shifted.

- Marvin

5 Conclusion

Finally,this work summarizes how blockchain secure and all of its features should design securely yet, when go over smart contracts,developer rely on standards.However,specifically NFTs are explained that bring many benefits to blockchain.But there are many risks on and off chain NFTs.As a result of this risks,developers need to investigate On-Chain NFTs despite high costs.Then,need to be aware of rising securities of NFTs.For buyers,they need to make sure that their NFTs On-Chain.

- Mustafa

References

[1] Ethereum merge, . URL https://ethereum.org/en/upgrades/merge/.

[2] Ethereum nodes and clients, . URL https://ethereum.org/en/developers/docs/nodes-and-clients/.

[3] Chapter 4. cryptography - elliptic curve cryptography, . URL https://www.oreilly.com/library/view/mastering-ethereum/9781491971932/ch04.html#:~:text=Ethereum%20uses%20the%20exact%20same,%2C%20called%20secp256k1%20%2C%20as%20Bitcoin.

[4] Ethereum yellowpaper, . URL https://ethereum.github.io/yellowpaper/paper.pdf.

[5] Functionality of smart contract. URL https://www.ibm.com/topics/smart-contracts.

[6] Keccak reference, . URL https://keccak.team/files/Keccak-reference-3.0.pdf.

[7] Keccak-256 submission, . URL https://keccak.team/files/Keccak-submission-3.pdf.

[8] Merkle tree, . URL https://soliditydeveloper.com/merkle-tree/.

[9] Blogpost of vitalik buterin - merkle tries, . URL https://blog.ethereum.org/2015/11/15/merkling-in-ethereum.

[10] Nft metadata. URL https://www.weforum.org/agenda/2022/02/non-fungible-tokens-nfts-and-copyright/.

[11] Reentra. URL https://cryptomarketpool.com/reentrancy-attack-in-a-solidity-smart-contract/.

[12] Standards of smart contract, . URL https://www.blockchain-council.org/ethereum/erc-token-standards/.

[13] Standardlist of smart contract, . URL https://www.blockchain-council.org/ethereum/erc-token-standards/.

[14] Blockchains energy consumption. URL https://www.forbes.com/advisor/ca/investing/cryptocurrency/bitcoins-energy-usage-explained/.

[15] erc721. URL https://ethereum.org/en/developers/docs/standards/tokens/erc-721/.

[16] Ethereum block architecture, . URL https://ethereum.stackexchange.com/questions/268/ethereum-block-architecture.

[17] Ethereum virtual machine, . URL https://ethereum.org/en/developers/docs/evm/.

[18] Ethereum wallets, . URL https://ethereum.org/en/wallets/.

[19] Cryptographic hashes and bitcoin. URL https://freecontent.manning.com/cryptographic-hashes-and-bitcoin/.

[20] Private and public keys on ethereum. URL https://www.massmux.com/private-and-public-keys-on-ethereum/.

[21] Nfts - sleepminting. URL https://timdaub.github.io/2021/04/22/nft-sleepminting-beeple-provenance/.

[22] Pentagon finds vulnerabilities. URL https://www.techrepublic.com/article/pentagon-finds-concerning-vulnerabilities-on-blockchain/.

[23] A. M. Antonopoulos and G. Wood. *Mastering ethereum: building smart contracts and dapps.* O'reilly Media, 2018.