LUISS

Corso di laurea in Management & Computer Science

Cattedra Blockchains and Cryptocurrencies

The Blockchains used to develop of web 3.0 & digital identity

Massimo Bernaschi — RELATORE

273891 — CANDIDATO

Anno Accademico 2024/2025

Table of Contents

Introduction

Background

The accelerated technology evolution in the late 20th century paved the way for the birth of new technologies. The first victim was the Data Encryption Standard (DES), an early cryptographic standard created in the 1970s, adopted by the US government in 1977. Second, in 1989, two scholars at CERN in Geneva, Tim Berners-Lee and Robert Caillau suggested a new software project that would allow people to exchange electronic information and documentation, regardless of what platform they were using, this research being extremely successful made world of internet becoming popular. This mechanism is now referred to as the web. Since then, the world of web1 changed a lot.. Now we are in the web2 era where we can surf web pages where we can read and write. We're miss using this technology at ours best. So, what are some of the most important aspects of web2 ?

Objectives

This project aims at understanding in general how the blockchain functions, studying the peculiar implementations in the cryptocurrencies such as proof of work on the bitcoin and proof of stake on Ethereum. We will explore the web. Then we will learn how the smart contracts work and what is decentralised identity, how it works, and what are the benefits of it. Finally, we will show some of the most exciting projects that at the moment are decorating the Ethereum blockchain like Worldcoin, Lifeform, Polygonid.

Scope

Scoping the project itself will be able to understand the state of the art of the blockchains, we'll understand how blockchain works out there and which implementation is outperforming. After that, the current state of the art of the web, and guiding us to where we are and where we will be. After we grasped the overall context, we are in, we'll head to the sandbox of decentralised identities. The ultimate goal is to investigate the most exciting projects that currently exist in the world, figure out who have the most brilliant ideas and what are their goals to the future.

Chapter 1: How Does Blockchain Technology Work?

The Blockchain is a data structure made up of growing lists of records, or "blocks", which are securely linked to each other through cryptography. Each block has a cryptographic hash of the previous block, a timestamp and transaction data inside it. For each block includes detail about the

previous one, therefore these actually create a chain connecting each block to the prior ones. Thus, the transactions in blockchain when saved, once saved, the data will not change, and it is immutable until any future blocks are altered.

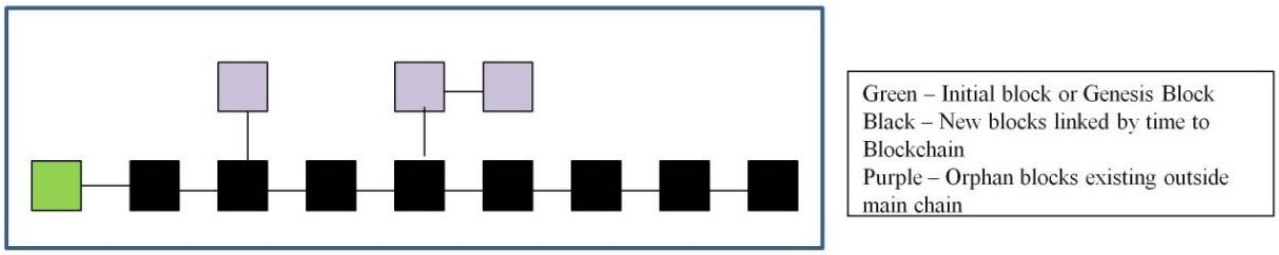


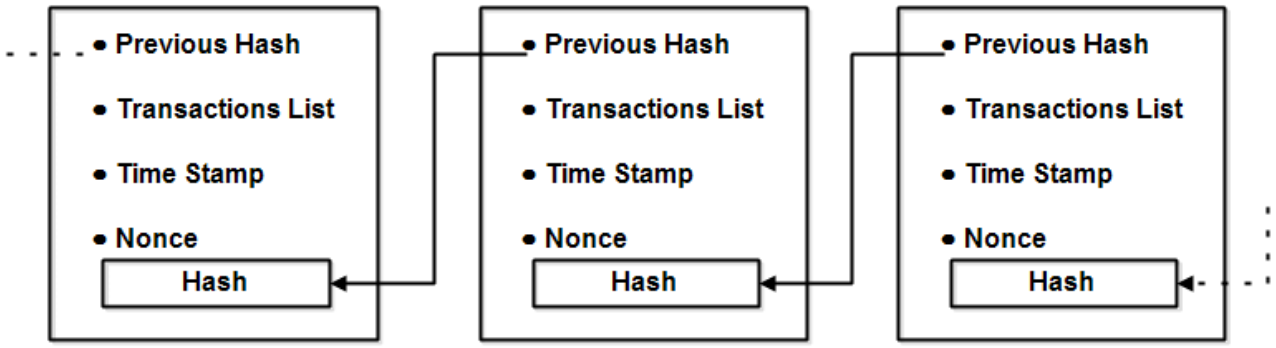
Fig.1.1. Block Chain Formation

Recovery of the blockchain with the main chain blocks (black blocks), of genesis block (green block) and orphan blocks (purple blocks).

The blockchain belongs to the family of distributed ledgers (distributed ledger), that is systems that are based on a replicated register, shared and synchronised between multiple entities present in various locations belonging to the same person. Regarding blockchains, it is not even necessary for the nodes that participate to know each other's identity, or to trust each other, since the addition of a new block is globally regulated through a common protocol, which guarantees consistency between the different copies. Now each node updates its private copy once this new block is approved. And the shape of the data means it will not be fiddled with again:

Systems based on blockchain, and distributed ledger technologies share common characteristics: data digitisation, decentralisation, disintermediation, transfer, traceability, transparency/verifiability, register permanence and transfer programmability. Therefore, thanks to these characteristics the blockchain is seen as an alternative in terms of security, reliability, transparency and costs to databases and registers managed centrally by seated and regulated authorities (public administrations, banks, insurance companies, payment intermediaries, etc.) [1]

A blockchain, then, is an open and distributed digital ledger that can securely, verifiably and permanently store data records ("'transa… Established protocol and validation scheme dictates that all blocks must be in consensus for a rewrite to even attempt to occur which renders historical data more accurate than the equivalent date point but rewrites thereafter require over 51% of networks to agree due to the effect of changing even one block has on all subsequent blocks from that point on. Hence blockchain is described as a growing list of "blocks" linked to one another and secured by cryptography. A block can contain one or more associated transactions, a hash pointer to serve as a link to the previous block, and a timestamp.



We show a typical blockchain in Fig. 2. A blockchain is a collection of data sets collected in a chain of data packages (blocks) where a block consists of a number of transactions (TX1-n, see Fig. 1). Each successive block extends the blockchain, and thus it represents a complete ledger of the transaction history. Cryptography allows the network to validate blocks.

Each block not only contains the transactions, but also a timestamp, the hash value of the previous block ("parent") and a nonce, which is a random number used for verification of the hash.
This idea guarantees that the whole blockchain is consistent all the way back to the very first block (the "genesis block"). The hash values are unique, fraud can easily be prevented because a change in a block in the chain will immediately cause the corresponding hash value to change. If most nodes of the network reach a consensus mechanism regarding its validity regarding its transactions in a block, and the validity of the block itself, the block will be added to the chain. As Swanson (2015) describes this consensus mechanism "is the process in which a majority (or sometimes all) of network validators agree upon the state of the ledger. It is a collection of rules and procedures that enables maintaining consistent set of facts among many nodes participating in it". [2]
Hence, any new transaction will not be automatically added to the ledger. Instead, the consensus process ensures that these transactions are kept in a block for a period of time (10 min in the case of Bitcoin blockchain) and then transferred to the ledger. Once written, the data in the blockchain cannot be altered again. By contrast, in the case of Bitcoin, so-called miners create the blocks, they are rewarded with Bitcoins for block validation. For instance, Bitcoin shows how the idea of the blockchain does not just revolutionize how money transactions take place. With the help of cryptography, anyone can trust each other and transfer peer-to-peer any type of assets over the internet globally.
From the aforementioned distributed ledger system, it brings a ton of advantages. The particular features of the network survive even in the event of specific nodes failing, unlike with centralised systems. This builds the trust as the people do not have to validate the trust of the intermediary or other users in the network. All everyone needs to do is develop a faith in the system more broadly.

The lack of third party also ensures data safety. The third parties are not in direct contractual relation with the data subjects, which includes personal data and the risk of security breaches. When a blockchain is used, third parties are no longer needed, and the user's security is increased. [3]

Chapter 2: Bitcoin uses blockchain for prof of work

History

Proof of work (PoW) is a type of cryptographic proof in which one party (the prover) gives the other party (the verifier) specific evidence that a certain amount of a specific computational effort has been expended. This makes it easy for verifiers to verify this expenditure with very minimal effort on their side. Moni Naor and Cynthia Dwork invented the idea in 1993 to deter simultaneous denial-of-service attacks and other abuse of resources such as spam on a network, by requiring some work from a service requester, which usually means processing time on a computer. The phrase “proof of work” was first used and formalised through a paper by Markus Jakobsson and Ari Juels in 1999. In 2004 Hal Finney adapted the idea to digital tokens called "reusable proof of work" using the 160-bit secure hash algorithm 1 (SHA-1).

It was later popularised by Bitcoin as a basis for consensus in a permissionless decentralised network, in which miners compete for the right to append blocks and be rewarded by creating new currency with each new block, and where each miner would experience the success probability proportional to the amount of computational work done. PoW and PoS (proof of stake) are still the two best known Sybil deterrence mechanisms. They are the most common mechanisms in the context of cryptocurrencies.

And in 2009, the Bitcoin network came online. Bitcoin follows the Hashcash PoW after Finney’s RPoW [4] and is a proof-of-work digital currency. However, in Bitcoin, double spend protection is provided by a distributed P2P protocol for tracking transfers of coins rather than the TCF used by RPoW. The reason that Bitcoin is more trustworthy is that it is secured by computation. Bitcoins are "mined" by individual miners through the Hashcash proof-of-work function, they are verified by either of the decentralised nodes in the P2P bitcoin network [5]. The time is adjusted with keeps a time up to target time (every so often, the difficulty is adjusted).

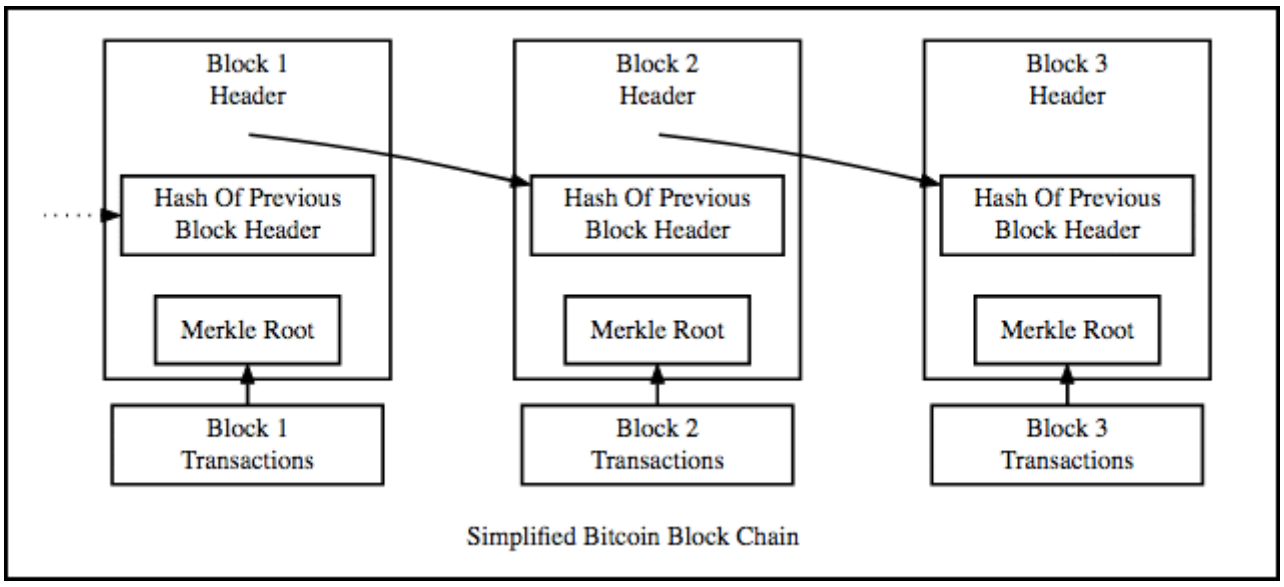
The Bitcoin blockchain has cryptographic problems for which miners must compete, their solution must be agreed upon by and reaches consensus on all nodes. These solutions are then used to validate transactions, add blocks, and mint new bitcoins. Miners receive rewards for solving these puzzles and producing new blocks. Nevertheless, the Bitcoin-type mining course of is highly

energy intensive as a result of the proof of work was showed after a lottery system. The only usefulness of the underlying computational work is in securing the ledger, providing the security that allows for open access to the network while working under adversarial conditions. To add a transaction (a new block) into the blockchain, miners are required to use a lot of energy. The use of energy in this contest is what secures bitcoin, makes it very costly to attack. However, electricity is not an issue for miners, you have to put up fixed cost such as a big space for mining to invest in computer hardwares. [6]

Separator Space: Bitcoin blockchain architecture

The block chain that underlies Bitcoin is the public ledger, an ordered and time-stamped record of the transactions. This procedure is utilized to guarantee that double-spending and changing past transaction logs are avoided.

Each full node maintains its own copy of the block chain in which it stores only the blocks that it has accepted. When a number of nodes have the same blocks in their block chain, they all agree that they are in consensus. Consensus rules are the validation rules that such nodes use to reach consensus. This section outlines much of the consensus rules of Bitcoin Core.



Simplified Bitcoin Block Chain

The above image is an example of a simplified form of a block chain. One or more new transactions are collected into the transaction data component of a block. Copies of each transaction are hashed, then the hashes are paired and hashed, then paired again and hashed until only a single hash remains, the merkle root of a merkle tree [7]. The merkle root can be found within the block header. Each block additionally retains the hash of the previous block's header, chaining the blocks. That makes it impossible to change a transaction without changing the block that holds it and all the blocks that follow.

All transactions are also chained together. Bitcoin wallet software makes it seem that satoshis are sent from, and to, wallets, but bitcoins actually travel from transaction to transaction. Each transaction spends satoshis it received in one or more previous transactions, making inputs of one transaction outputs of an earlier one.

Proof of work

The block chain is collectively maintained by anonymous peers in the network so Bitcoin makes each block prove that work has been put into its creation to ensure that untrustworthy peers wanting to modify previous blocks will have to work harder than honest peers wanting only to add new blocks to the block chain.

All this makes it infeasible to change transactions of any block if we want to change all the blocks after it. This means that the cost to change a given block increases with every new block in the block chain, ever more amplifying the effect of the proof of work.

Bitcoin's proof of work exploits the argedly random nature of cryptographic hashes. So, a good cryptographic hash algorithm turns arbitrary data into what appears to be a random number. If the data changed in anyway and the hash is re-run a new seemingly random number will be generated, so there is no way to change the data to obtain a predictable hash number.

In order to show that you did a bit of extra work to produce a block you have to hash the block header and having to generate a hash below a target value. You can prove that you have tried at most two combinations by producing a hash value less than 2255, if the maximum possible hash value is 2256 – 1 = 2255.

The example you provided would, on average, yield a successful hash every 2 tries. You can even calculate the likelihood that a certain hash attempt will produce a number lower than the target threshold. Bitcoin assumes a linear probability that the lower the target threshold it makes, then the average amount of hashes it will have to try will be increased.

A consensus protocol is a protocol for validating transactions on the blockchain. The network measures time every 2,016 blocks by timing how many seconds has elapsed between the generation of the first and last of those last 2,016 blocks based on timestamps that are saved in each block headers. The perfect value is 1,209,600 seconds (two weeks).

If the generation of the 2,016 blocks took less than two weeks, the expected value of the difficulty is increased in proportion (up to 300%) so that the next 2,016 blocks should take exactly two weeks to generate if the hashes are checked at the same rate.

If it took more than 2 weeks to create the blocks on average, the current expected difficulty value will be reduced in the same way (up to 75%).

(Note: due to an off-by-one error in the Bitcoin Core implementation, the difficulty is adjusted every 2,016 blocks based on timestamps from only 2,015 — leading to a slight skew.)

Since every block header must hash below the target threshold, and because every block is chained to its predecessor, (on average) it takes as much hash power to propagate a tampered with block as was expended by the entire Bitcoin network between the time the original block was created until the present. It is only if you were able to take control of more than 51% of the network's hashing power that you would be able to successfully pull off a 51% attack against the history of transactions (although, to be fair, even having less than 50% of hashing power still gives a good chance of committing such attacks).

The block header contains a number of fields that can be easily modified, including a field reserved just for a nonce, so new hashes do not need to wait for new transactions to be posted. Additionally, only the 80-byte block header is included in the proof-of-work hash, meaning the volume of transaction data in a block does not slow down hashing with extra I/O, and adding more transaction data just means recalculating the ancestor hashes in the merkle tree.

Bitcoin Limitations

But else than some papers in the industry about white papers on blockchains, the research around blockchain in information systems is also currently mostly about crypto currencies. Alongside the positive potential, there are negative aspects and risk factors which are covered in this stream of literature.

Barber et al. (2012) points out various flaws of Bitcoin like theft or loss of Bitcoins (malware attacks, accidental loss), scalability issues (e.g., delayed transaction confirmation, data retention, and communication failures), and structural problems (e.g., deflationary spiral). For their part, Barber et al. Vanel (2012) proposes solutions to enhance the current Bitcoin technology. One such might be "fair exchange protocol" which could enhance the user anonymity. Other authors similarly have written on Bitcoin's privacy implications (e.g. Andreoulakis et al. 2013: Bonneau et al. 2014: Miers et al.

2013). Protecting privacy in this Bitcoin world is possible only via pseudonyms. To add to Bitcoin, Miers et al. (2013) subsequently proposed Zerocoin, which enables people to trade

cryptocurrencies in an entirely anonymous way. Zcash, which is the successor to Zerocoin was launched in 2016.

If blocks are added to the network at a high rate, then performance issues are probable through to the process of generating new blocks. Lewenberg et al. (2015) proposed "Inclusive Block Chain Protocols" that enhance the transaction speed. Croman et al give some insight into the scalability of Bitcoin (2016).

Electricity consumption per year by country

ZBitcoin, with annual energy consumption of approximately 91TWh/year, uses even more energy than Finland. A different estimate is that Bitcoin now uses 150 TWh of power a year.

Bitcoin has an annual consumption of roughly 87 TWh (as of writing according to the Cambridge Centre for Alternative Finance). Bitcoin mining eats energy — using special machines to create cryptocurrencies is complicated but requires a lot of computing power.

Kalaam Crypto Data Electricity ConsumptionGreenhouse Gas Emissions Energy Consumption Per Transaction

Compared to Visa, Bitcoin, Ethereum, various PoS networks comparison of electricity usage per transaction as shown in chart below. We acknowledge this is not an apples-to-apples comparison (for e.g. Bitcoin's energy usage isn't dependent on transaction volume — unlike Visa), but we include it for completeness' sake.

Electricity Usage per Transaction – From the chart, PoS network is closer to Visa level, while Bitcoin, Ethereum Classic have much higher electricity consumption.

Chapter 3: Ethereum blockchain Proof of stake

Overview

Ethereum is the second biggest public blockchain by value and the biggest public blockchain by usage. Ethereum, like bitcoin, runs on a public blockchain: a global distributed network that is unconsolable, permissionless and has no central authority.

The major distinction between Bitcoin and Ethereum is the existence of applications on Ethereum that can live along with the blockchain, and they are deployed on the blockchain itself, which serves as the basis of decentralised finance (DeFi). Ethereum is not technically a cryptocurrency.

History

Ethereum was introduced in 2013 in a white paper about the creation of the Bitcoin protocol. It was authored by Vitalik Buterin, a young programmer working on Bitcoin's teams and co-founder of

Bitcoin Magazine. The idea soon drew the attention of another cryptocurrency nuts back then that got involved with the project. Notably, this group includes Anthony Di Iorio, Charles Hoskinson, Joseph Lubin and Gavin Wood. They are now among Ethereum's co-founders. The Bitcoin dev team rejected the proposal, which led the founders to create their own chain.

Vitalik Buterin set up a fundraising campaign to make his dream a reality. Between 2 September 2014, a Bitcoin fundraiser was organized to support developments required to activate the network. However, Buterin was able to raise $18.3 million through pre-sales of the first Ether tokens (ETH). This represented 30,000 ETH. Attracted by the aims of the project, programmers took an interest and the first iteration of the Ethereum blockchain went live on 30July 2015; it was referred to as frontier.

Ether's price started to rise in 2016. This cryptocurrency proved to be a huge success even competing with Bitcoin. It did not appear on trading platforms until 2017. Its price climbed above $400, from $1 in 2015. As of today, there are estimated to be over 110 million ETH in circulation.

Ether the crypto

Ether is the cryptocurrency on Ethereum. Ether, denoted by ETH, is a means of exchange on the Ethereum network. Ethers are bought and sold on marketplaces and priced by supply-and-demand. Thus, ETH is an instrument for trading, mining or simple storage in a portfolio. Ether has a few of its own essential features. Perhaps the most salient characteristic is that its annual money supply is limited and relatively modest. This means that only a tiny number of new Ethers are created annually. So, it's a scarce asset, but unlike Bitcoin's hard limit of 21 million units, there's no absolute maximum. So, investing in Ether is like supporting the growth of the Ethereum ecosystem: it is already the second most growing in terms of market capitalisation and has been used to be the basis for many projects and applications that are built on top of its network. [8]

The proof of stake mechanism

Proof-of-stake is a system that makes it so that validators can prove they put something of value in the network that will die if they do something dishonest. In Ethereum's proof-of-stake, validators lock capital directly via ETH into a Ethereum smart contract. The validator's job is then to verify that newly propagated blocks across the network are valid and to occasionally create and propagate a block of their own. If they attempt to fraudulently double spend (bid or send multiple blocks when they should bid or send only one or send conflicting attestations), some or all of their staked ETH may be burned. Proof-of-stake is a blockchain consensus method to confirm transactions and add new blocks to the chain. A consensus mechanism is a way to validate database entries and protect the distributed database. In cryptocurrency, the database is referred to as a blockchain — so the

consensus mechanism secures the blockchain. Proof-of-stake decreases how much computational work goes into verifying blocks and transactions. Proof-of-work secured the blockchain through expensive computing needs. While in proof-of-work, the block time is determined by how difficult it is to mine a block, it is constant in proof-of-stake. There are slots (12 seconds) and epochs (32 slots) on proof-of-stake Ethereum. Every slot, one validator is randomly chosen as a block proposer. The validator creates a new block and broadcasts it to other nodes on the network. In every slot, there is also a randomly selected committee of validators whose collective votes are applied to determine if the block being proposed is valid or not. One of the implications of the nature of consensus algorithms is that the validator set gets divided into committees, helping to keep a cap on the network load. The committees split up the validator set such that in every epoch every active validator attest, just not in every slot.

How a transaction is executed in Ethereum p.o.s.

The below gives a full walk-through of how a transaction is executed in Ethereum proof-of-stake.

1. At the end of this process, a user builds a transaction and signs it with his/her private key. A wallet (or library like ethers) typically does this. js, web3js, web3py etc but under the hood the user is sending a request to a node using the Ethereum JSON-RPC API. The tip represents the amount of gas that the user is willing to pay as an incentive for a validator to include the transaction in a block. The tips are paid to the validator, the base fee gets burnt.
2. The transaction is then propagated to an Ethereum execution client, where it gets verified for validity. This includes ensuring the sender of the transaction has sufficient ETH to cover it and has signed the transaction with the correct key.
3. Assuming the transaction is valid, the execution client adds it into its local mempool (list of waiting transactions) and also propagates the transaction to other nodes of the execution layer gossip network as well. When other nodes hear of the transaction, they add it to their local mempool as well. So advanced users may not even broadcast their transaction, rather send it to specialize block builders like Flashbots Auctions. So, they can maximize the profit (MEV) from queuing those transactions in future blocks.
4. Recall that the block proposer for the current slot is one of the validator nodes on the network and was previously chosen pseudo-randomly using RANDAO. Our node builds and broadcasts the next block to be added to the Ethereum blockchain while updating the global state. A node consists of an execution client, a consensus client, and a validator client. An execution client bundles transactions from the local mempool into an "execution payload", executes them locally, and produces a state change. This data is sent to the consensus client, the execution payload is bundled as part of a "beacon block" which also includes staking rewards and punishments, slashing events,

attestations, etc. which allows the network to come to consensus on the head of the chain. More details about the communication between the execution and consensus clients can be found in Connecting the Consensus and Execution Clients.

5. The newly created beacon block is then propagated to other nodes over the consensus layer of the network. This gets passed to their execution client where the transactions are re-executed locally to verify that the proposed state change is indeed valid. The validator client then attests that this block is valid and that it is the next logical block in their idea of the chain (which is to say, that it extends the chain with the heaviest weight of attestations as defined in the fork choice rules). Each node that attests to it stores the block in its own copy of the database.

6. A transaction has been "finalized if it became part of a chain with a "supermajority link" in between at least two checkpoints. These checkpoints happen at the beginning of each epoch, they serve the purpose of this collateral damage because only a subset of active validators combine their attestation on every slot, however all active validators attest along each epoch. As a result, only between epochs can a "supermajority link" be proved (in other words, when 66% of the entire staked ETH on the system agrees two checkpoints). [9]

Crypto-economic security

Becoming a validator is a long-term commitment. The validator should have enough hardware and an internet connection to participate in block validation and proposal. The validator is then rewarded in ETH (their staked balance rises). However, operating as a validator can also expose new services for users to compromise the network for personal profit or attack. To incentivize honesty and participation, validators do not earn ETH rewards if they miss the call to action to attest, can even lose their stake with nefarious behaviour. There are two kinds of behaviour we can consider as dishonest: equivocation (sending different blocks in one slot) and double signing (sending conflicting attestations). The amount of ETH being slashed depends on how many validators are also being slashed at the same time. This is termed the "correlation penalty" and may be small (~1% stake for a single validator slashed independently), or it may destroy 100% of the validator stake (~mass slashing event). It is applied halfway through an involuntary exit window of 18 days, beginning with an immediate penalty (up to 1 ETH) on Day 1, a correlation penalty on Day 18, and finally, ejection from the network on Day 36. Since they are not submitting votes, they get small attestation penalties each day that they are on the network. This means that a coordinated attack would be extremely costly to the attacker.

Proof-of-stake and security

The danger of such a 51% attack lingers on proof-of-stake technology as it does on proof-of-work, but attackers have smartened up because proof-of-stake offers better stakes for risk for them. It would require an attacker to have 51% of the staked ETH. They could then use their own attestations to determine their preferred fork, which would be the one with the most accumulated attestations. Consensus clients decide the correct chain via the 'weight' of accumulated attestations; therefore, this attacker would be able to force their fork to become canonical. But proof-of-stake has a countermeasure advantage over proof-of-work where the onus falls to the community to start a counterattack. Both won't happen with honest validators deciding to ignore the attacker's fork and build on the minority chain and prompts apps, exchanges and pools to do the same. They might also decide to forcibly eject the attacker from the network and destroy his staked ETH. These are solid economic deterrents to a 51% attack.

Aside from 51% attacks, malicious actors may also try other malicious activity, such as:

- fold long-range attacks (although the finality gadget protects us from this attack vector)
- short range 'reorgs' (though proposer boosting and attestation deadlines mitigate this)
- bouncing and balancing attacks (which are also mitigable by proposer boosting, and in any case only shown under simplified network assumptions)
- avalanche attacks (mitigated by PoS fork choice algorithms which allow only recent messages to count)
- In summary, proof-of-stake as realized on Ethereum has been shown to be more economically secure than proof-of-work.

There are many other proofs, but I am going to cover only proof of work and stake because both are decentralised, and they are being used in blockchains as we saw with significant differences.

Chapter 4: The smart contracts

Traditional contracts are built on trust. The main issue with a conventional contract is that it relies on trusted third parties to act on behalf of the contract when it is enforced. Here is a simple one: Alice and Bob Are having a bicycle race. In this example, let's assume Alice bets Bob $10 that she is going to win the race. Bob is sure he is the winning one and accepts the bet. Consequently, Alice ends up finishing the race far ahead of Bob and ultimately wins. But Bob will not pay out on the bet, claiming that Alice must have cheated. This silly example shows why any agreement that is not smart is a problem. Even in the case that the terms of the contract are satisfied (e.g. you are the winner of the race), you must then rely on another party to deliver on the contract (e.g. payout on

the bet). So, in Smart Case scenario there wouldn't be imposed to trust another person because this software it be run and sent online. One example of a smart contract in action. A digital vending machine. As a simple metaphor for a smart contract, imagine how a vending machine actually works, the way a smart contract somewhat does — given specific inputs, you can expect predetermined outputs.

1. You select a product
2. It shows the price on a vending machine
3. You pay the price
4. The vending machine check that you paid enough
5. The vending machine spits your item out.

The necessary conditions need to be fulfilled before you can receive your product from the vending machine. If you fail to choose a product or insert enough money, the vending machine won't dispense your product. Because the smart contract deterministically executes unambiguous code when the certain conditions are met it has automatic execution so as with Computer programs Smart contracts is just computer program. In this context, the term "contract" has no legal significance; it simply executes the code. No need to wait for a human to interpret or negotiate the outcome. This eliminates the need for trusted third parties. For instance, you can deploy a smart contract where you hold funds in escrow for your child, and they can withdraw the funds after a particular date. The smart contract will not execute if they attempt to withdraw before that date. Or you could codify in a contract the automatic transfer of a digital representation of a car's title to you upon paying the dealer. It has predictable outcomes. Traditional contracts are vague because they depend on people to interpret and enforce them. Judges could interpret a contract with two different meanings, leading to inconsistent decisions and uneven results. This possibility is eliminated with smart contracts. Due to its deterministic nature the result of executing a smart contract is the same to anyone calling it depending on the context of the transaction triggering its execution and the state of the Ethereum blockchain at the time of execution. The smart contracts are executed exactly as per the conditions written in the contract's code. This level of precision provides the smart contract to create the same output under like conditions. It has a public record. Smart contracts help in auditing as well as tracking. Ethereum smart contracts run on a public blockchain, which means that anyone can quickly trace asset transfers and other transfer information. For example, you can verify that someone sent money to your address. It has given some privacy to the users. Because Ethereum is a pseudonymous network (your transactions are associated publicly with a unique cryptographic address, not your identity), you can hide from observers what the transaction says about you. They're See-through phrases similar to plain contracts. Finally, you can have an inside

look at what a smart contract contains before you sign it (or otherwise interact with it). The transparency of a smart contract ensures that anyone can analyse it. Smart contracts can accomplish essentially everything that a computer program can. [10]

The above definition leaves us with some questions about the life cycle of a smart contract.

In most cases smart contracts are written using high-level programming languages like Solidity. However, they must be compiled to the low-level bytecode that actually runs within the Ethereum Virtual Machine (EVM) to execute. If compiled, they are then deployed upon the Ethereum platform with a specialized contract creation transaction, with the recognition being that they are sent to the specialized contract creation address, where the address for it is namely, 0x0 (see [contract_reg]). Every contract is identifiable through an Ethereum address, which is determined from contract creation transaction as a function of originating account and nonce. A contract's Ethereum address can be used as the recipient in a transaction, sending funds to the contract or triggering one of the contract's functions. Note that there are no keys associated with an account created for a new smart contract like there are with EOAs. As contract creator, you don't gain any special privileges at the protocol level (unless you explicitly code that into the smart contract). You don't get the private key for the contract account, because it doesn't even exist in the first place—we can say that smart contract accounts own themselves. Contracts only run (and thus, use gas) if they are called by a transaction. Ultimately all smart contracts are executed on Ethereum from a transaction from an EOA. A contract can call another contract that calls another contract that calls another contract, etc, but the first contract in such a chain of execution must always have been called by a tx from an EOA. The contracts don't run "on their own" or "in the background." Contracts remain dormant until a transaction triggers them (directly or indirectly in a chain of contract calls) to be executed. Furthermore, smart contracts are also not "executed in parallel" in any sense, the Ethereum world computer can be considered to be a single-threaded machine.

Transactions can be atomic, meaning they either successfully terminate or enter a rolled-back state. Under different scenarios, a successful termination of a transaction means different things: (1) If a transaction gets sent from an EOA to another EOA then all changes to the global state (e.g. account balances) made by the transaction are recorded; (2) If a transaction gets sent from an EOA to a contract not invoking any other contracts, all changes to the global state recorded (e.g. account balances, state variables of the contracts)(3) If a transaction gets sent from an EOA to a contract only invoking other contracts in an error propagating manner, all changes to the global state will be recorded (e.g. account balances, state variables of the contracts); and (4) If a transaction gets sent from an EOA to a contract that invokes other contracts in an non error propagating manner, there may only be some changes to the global state recorded (e.g. account balances, state variables of the

non-erroring contracts), others changes to the global state are not recorded (e.g. state variables of the erroring contracts). Otherwise, if a transaction is reverted, then all effects (state changes) of the transaction are “rolled back” (that is, as if they were never run). A failed transaction is still recorded as attempted, and the ether that is spent to execute the transaction is deducted from the originating account but otherwise does not have any other effects on contract or account state.

As previously mentioned, code of a contract cannot be altered. However, a contract can be “destroyed,” which removes the code and internal state (storage) of a contract at its address, leaving an empty account. No code at that account address gets executed in response to those incoming transactions, since there is no longer any code there. However, in order to delete a contract, you are using an EVM opcode known as SELFDESTRUCT (formerly known as SUICIDE). One particular operation cost “negative gas,” a gas rebate, thus encouraging the release of network client resources in return for the elimination of stored state. It does not delete the contract's transaction history (past) because the blockchain itself is immutable. Also, the SELFDESTRUCT capability will only be available if the contract author programmed that in the smart contract. Smart contracts cannot be deleted if the contract's code does not contain a SELFDESTRUCT opcode, or if the opcode is inaccessible

Chapter 5: Web1, 2 & 3. Past, present & future

From Web 1 and 2 to 3

The Web is something most people see as an ongoing pillar of modern life — it was invented, then it just existed. But the Web we know most days is a far cry from what was originally envisioned. To better grasp this, we can loosely divide the Web’s short history into two eras — Web 1.0 and Web 2.0.

Web 1.0: A Read-Only Web (1990-2004)

Tim Berners-Lee was in the throes of writing the protocols that would emerge as the World Wide Web in 1989 at CERN, Geneva. His idea? To build open, devolved protocols which would enable information to be shared from anywhere in the world.

The concept was initially introduced from about 1990 to 2004 — a period now called ‘Web 1.0’ and the early adoption of Berners-Lee’s creation. Web 1.0 consisted primarily of static websites owned by companies, and interaction between users was as good as zero — people rarely engaged in content creation — hence it was nicknamed the read-only web.

Web 2.0: Read-Write (2004-present)

With the rise of social media platforms, the era of Web 2.0 started in 2004. Where the web was originally a read-only, it became read-write. In addition to companies supplying content to users, they also started to provide platforms for the sharing of user-fueled content and user-to-user interactions. As more people went online, a small number of leading companies began to demand a disproportionate share of the traffic and value created on the web. And Web 2.0 gave rise to the advertising-driven revenue model. Users were able to create content, but they did not own it or share in the monetisation.

Web 3.0: Read-Write-Own

The idea behind 'Web 3.0' was popularized by Ethereum co-founder Gavin Wood soon after Ethereum's launch in 2014. Gavin articulated a solution to a problem that many early crypto adopters experienced: the Web required too much trust. That is, much of the Web that people navigate today depends on trust in a small number of private companies to do the right thing for the public.

Web3 has turned into an umbrella term for the idea of a new, improved-internet. At its heart, Web3 employs blockchains, cryptocurrencies, and NFTs to empower users to have ownership. A post on Twitter from 2020 put it succinctly: Web1 was read-only, Web2 write-write, Web3 read-write-own.

The development of Web3

While it is difficult to nail down a precise definition of what Web3 is, there are some founding principles that have been shared among the people building it. The identity. You would sign up for each and every platform. For instance, you might have accounts on Twitter, YouTube and Reddit. Want to edit your display name or profile picture? You need to do this for every account. In some cases, you can use social sign-ins, but that raises a familiar problem — censorship. In one click, these platforms can sever you from your entire online existence. To make things worse, when you want to create an account, many platforms require you to trust them with personally identifiable information. The beauty of Web3 is that you can own and manage your digital persona by way of an Ethereum address and Ethereum Name Service (ENS) profile. An Ethereum address allows a shared login across platforms that is secure, censorship-resistant, and anonymous. Web3 is decentralised: instead of huge swathes of the internet that is controlled and owned by centralised entities, ownership gets spread out amongst its builders and users. [11]

Decentralised Autonomous Organisations (DAOs)

Additionally, where you own your data in Web3, you can jointly own the platform itself, through tokens that are essentially shares in a company. DAOs allow you to coordinate the decentralized ownership of a platform and determine the future direction of it.

In technical terms, DAOs are smart contracts that execute the rules for decentralized governance of a set of resources (tokens). Resource expenditures are decided by voting on how to spend them, and the code automatically executes the results of the vote. That said, many people refer to the DAOs as Web3 communities. All these communities have varying degrees of decentralised and automation by code. This is part of a series exploring what DAOs are and what they could become. Web3 is permissionless: everyone can join Web3, and no one is being left out. It does this with native payments: it uses cryptocurrency to let you spend and send money online, without the need for outdated bank and payment processor infrastructure.

DAOs give us a platform to collaborate with our peers across the world without having to rely on a benevolent leader to manage the funds or operations. There's no chief executive who can throw money around as they please, or chief financial officer who can cook the books. Instead, rules embedded in the blockchain determine with baked in how the organization operates and money is spent. They have treasury vaults that nobody has the power to enter unless approved by the collective. With proposals and voting, we ensure that every person in the organization has a voice and that everything is done transparently (on-chain). Creating an organization as a person that relatively involves funding and money are big commitments, and you have to trust the people whom you are working with a lot. But it's difficult to trust someone you've only ever met online. With DAOs if you trust no one but the DAO code that is 100% transparent and can be verified by any party no need to trust anyone else in the group.

Some examples of how you might implement a DAO:

• A charity — you could take donations from anyone in the world and vote on the causes to fund.

•Collective Commons Ownership – you could buy physical or digital assets that members vote on how to use.

• Ventures and grants — you can build a venture fund and pool investment capital and vote on ventures to fund. Returned assets could later be divided again to DAO-voters.

At the core of a DAO is its smart contract, which outlines the rules of the organization and manages the group's treasury. When the contract is operational on Ethereum, no one can change the rules, except through a vote. Doing anything that is not described by rules and logic in the code will cause the system to fail. And since the actual treasury is also defined by the smart contract,

nobody can spend that money without the judgment of the group either. This implies that there is no need for a central authority for DAOs. Rather, the group achieves consensus, and payments are automatically approved when votes carry.

This is feasible because smart contracts become tamper-proof after being deployed on Ethereum. The DAOs rules (the code) are public, so you can't just edit them without anyone noticing.

There is a huge imbalance between content creators and platforms. OnlyFans is a user-generated adult content site with more than 1-million content creators, many of them use the platform as their main source of income. OnlyFans had previously made plans to ban sexually explicit content in August 2021. The news set off a wave of outrage among creators on the platform, who believed they were getting robbed of an income on a site they helped build. After the outcry, the decision was quickly reversed. So, even if the creators technically lost this battle, it just highlights a problem for Web 2.0 creators: by leaving a platform, you lose your reputation and following that you built (the Trova Web 3.0 project mentions how that is leaving a lot of creators in a dilemma of figuring out what route to go). On Web3, your data belongs on the blockchain. When you choose to depart from a platform, you can take your reputation with you, plugging it into another interface that presents your values more coherently. Web 2.0 relies on content creators to trust the platforms not to change the rules, whereas censorship resistance is a native property of a Web3 platform. Web3 is thrustless: it runs on incentives and economics rather than trusted third parties.

Web3 also enables you to have ownership of your digital assets like never before. For instance, let us say you are playing a web2 game. It ties directly to an account, if you buy an in-game item. If the creators of the game delete your account, you lose these items. Or if you quit the game, the value of the time and money spent on your in-game items evaporates. Through non-fungible tokens (NFTs), Web3 enables true ownership. No one — not even the game's creators — can strip you of your ownership. And if you quit playing, you can sell or trade your in-game items on open markets and reclaim their worth. NFT is a type of token that is unique per unit. NFTs feast on the fact that each has different properties (non-fungible) and is provably scarce. This is in contrast with, for instance, ETH or other Ethereum based tokens such as USDC where each token is identical with the same properties ('fungible'). You don't care which exact dollar bill (or ETH) you have in your wallet, because they are all the same and worth the same. But you do care about which NFT you own, as each has unique properties that sets them apart from those of others ('non-fungible'). Because each NFT is unique, they allow for the tokenization of items such as art, collectibles, or real estate, where a particular unique NFT represents a particular unique real world or digital thing. The Ethereum blockchain will allow them to publicly verify that they own an asset.

Some issues that exist on the internet are solved by NFTs and Ethereum. As everything gets to be

more digital, there's a demand for us to begin to replicate properties of physical items — scarcity, uniqueness, evidence of ownership — in a way that is not controlled by a central group or organization. With NFTs for music, you can own an mp3 file that will work across all Ethereum based apps and not be limited to a specific music app by a single company like Spotify or Apple Music. Perhaps you own a social media handle that you can sell or trade, but which cannot be confiscated arbitrarily by a platform provider

Chapter 6: The decentralised identity

Begin with the definition: What is an identity? Through the characteristics of every individual, identity is defined. Identity is an individual, that is, an individual human being. Identity that can also be placed in the human or judicial concept examples: organization, etc. An identifier is something that serves as a pointer to a specific identity or group of identities. Some common identifiers are Name, Social security number/tax ID number, Mobile number, Date and place of birth, Digital identification credentials, e.g. email addresses, usernames, avatars. These conventional illustrations of identifiers are given, held and overseen by focal elements. You have to ask your government and get permission to change your name, or you have to ask the social media platform for authority to change your handle. Decentralized identity has several advantages. As we mentioned in the previous article, decentralized identity provides more control over identifying information. They provide verification independent of centralized authorities and third-party services. Decentralized identity solutions enable a thrustless, easy and privacy-preserving way to verify and manage user identity. This creates trust between different parties and blockchain technology ensures that the user receives cryptographic guarantees for proving attestations (third-party statements and or personal credentials) they own. Decentralized identity makes identity data portable. Users hold attestations and identifiers in a mobile wallet and can present them to any counterparty of their choice. Issuing organizations do not lock decentralized identifiers and attestations into their database. Decentralized identity will function well with new zero-knowledge technologies that will allow a person to prove that they possess or have done something without having to disclose precisely what that something is. That could turn into a powerful combination of trust and privacy, for applications like voting. Decentralized identity supports anti-Sybil techniques to track when a single human is pretending to be many humans to game or spam some system.

Use-cases of decentralised identity Let's check what are the possible use-cases over decentralised identity:1. It enables service providers to issue users attestations to store in an Ethereum wallet. For example, an attestation could be an NFT that gives a holder access to an online community. A Sign-In with Ethereum functionality would then allow servers to validate the user's Ethereum account and retrieve the necessary attestation from the user's account address. That way users will use platforms and websites without needing to remember long passwords and makes the online presence better for users. Many online services require consent from individuals to submit attestations and credentials (I.e., driving license or national passport). However, this method is not without its challenges; private user data is exposed, and service providers do not have a way to know if the attestation is authentic. Instead, companies can use decentralized identity to forgo traditional Know-Your-Customer (KYC) processes and still authenticate user identities at the level of Verifiable Credentials. It lowers the dust of identity management and prevents fake documentation. Decentralized identity is being applied in new areas, such as online voting and social media. Online voting systems are vulnerable to tampering, particularly if bad actors set up fake personas to cast votes. This helps to maintain the integrity of online voting processes by asking individuals to present on-chain attestations. In this way, decentralized identity can help detox our online communities from fake accounts. For instance, each user could authenticate their identity via some on-chain identity system, such as the Ethereum Name Service, limiting the chances of bots. Because the value of a grant is greater when more different individuals vote for it, grant-giving applications that utilize quadratic voting are subject to Sybil attacks that incentivize users to split their contributions across many identities. Decentralized identities help prevent this duplication by raising the bar on each actor providing sufficient proof that they are indeed human (though not necessarily specific private information). Attestation is a statement made by one entity about another entity. Your driver's license, issued to you by the Department of Motor Vehicles (one entity), confirms that you (one entity) are legally permitted to drive a car. Attestations are not identifiers. An attestation is a unique identifier that represents a specific identity and presents a claim regarding an attribute of the identity. So, your driver's license has identifiers (name, date of birth, address) but is also the attestation about your legal right to drive. The standard identifiers — your legal name, your email address — depend on third parties — governments and email providers. Decentralized identifiers (DIDs) are, however, different—they're not issued, managed, or controlled by any one central entity. [12]

Decentralized identifiers are self-sovereign and under individual control. for example, Ethereum account is a decentralized identifier You can create as many accounts as you like, no permission from anyone needed, and no need to keep them in a centralized registry.

Best of all, decentralized identifiers are saved on distributed ledgers (blockchains) or peer-to-peer networks. Which in turn makes them globally unique, highly available resolvable, and cryptographically verifiable. Unlike a traditional identifier, a decentralized identifier can relate to different entities, such as people, organizations, or government institutions.

How digital identity will be modelled ?

Public-key is a form of info security in cryptography which gives one entity a public key and a private key. Blockchain networks rely on public-key cryptography to validate user identities and demonstrate ownership of digital assets. Certain decentralized identifiers have a public and private key, like an Ethereum account the public key represents the controller of the account, and the private keys can sign and decrypt messages on behalf of this account. Cryptographic signatures can be used to prove all claims (impersonation, use of fake identities, etc.), which is what public key cryptography provides integral proofs to authenticate entities. A blockchain is a verifiable data registry: an open, thrustless, decentralized database of information. Decentralized registries are unnecessary due to the existence of public blockchains. You know that if somebody has to verify a decentralized identifier, it could check the corresponding public key on chain. Assuming ownership of these identifiers is more about holding the required keys rather than being authenticated by third parties like in the traditional way. Decentralized identity is a concept which states that something associated with identity is supposed to be self-controlled, private, portable, with decentralized identifiers and attestations being the leading building blocks. Within decentralized identity, attestations (or Verifiable Credentials) are tamper-proof cryptographic proofs provided by the issuer. Each attestation or Verifiable Credential issued by an entity (like an organization) is bound to their DID. Because DIDs are on-chain, anyone can verify the validity of an attestation by checking the issuer's DID on Ethereum. So, the Ethereum blockchain is essentially like a digital global directory for verifying DIDs assigned to certain entities. This is why attestations are self-controlled and verifiable — because of decentralized identifiers. The holder always has proof of the provenance and validity of the attestation, even in the event the issuer no longer exists.

Decentralized identity also relies on decentralized identifiers to protect the privacy of personal information. Sound examples include, if a person sends one proof of attestation (a driver license) the validating party does not need to verify the correctness of data in the proof. Instead, all the verifier needs are cryptographic assurances of both the integrity of the attestation and the identity of the organization that issued it in order to ascertain whether the proof is correct.

Chapter 7: difference between on/off chain attestations

It allows off-chain attestations, wrote persistent. In this set up attestations become blob json files and are directory stored off chain (ideally on a decentralized cloud storage service like IPFS or Swarm). (Note: still not on the blockchain, but a hash of the JSON file is on chain, easy to bind to a DID via some on-chain registry) The corresponding DID may either be that of the issuer of the attestation or the receiver. This means that separate from a claim, attestation can stand the test of time on a blockchain, while the claims information is still encrypted and verified. said Mihailo Bjelic, Polygon's co-founder. "It is also fantastic demonstrations on how zero knowledge proofs can enable us to create a better world." The Vision & Guiding Principles The only solution that meets both the world's digital identity needs– public identity reveals useful information uniquely, preventing ownership challenge and enabling trust establishment– all while being private-by-default is a future proof solution: stronger privacy guarantees naturally imply more possible use cases, hence more users on-board. Decentralization and self-sovereignty principles: Users should be able to control their identity and personal information in order to use social coordination and regain power from third parties. Computation trust for p2p networks (dWeb-of-Trust): identity attributes can be expressed as claims (or attestations), can be combined to form compound proofs. Open and permissionless: Claims flow from one identity to another identity, without passing through a third party. The ability to steer AI systems is being opened up to individual users, not only businesses, which is leading to new use cases. Web3 Identity & Verifiable Credentials Benefits of expressible claim standard over NFT and VC NFTs are public, and expensive to mint. VCs provide selective disclosure and ZK add-on to provide privacy, but they limit expressibility and composability (necessary for showing it on applications). It is prohibitively expensive to verify it on-chain as a VC. The complexity that Circom 2.0 introduces to compile zero-knowledge cryptographic constructions known as zkSNARKs circuits however is reduced through leveraging of internal expertise at Polygon ID. Developers & partners on board with the ID client toolkit natively apps, SDK's and whitelabel solutions. The applications need to specify the private attributes required to the user to prove they own before executing an on-chain private validation, which is achieved through a unique protocol called zkProof Request Language. What is relayer? It's a unique model boosting Web3 apps for "any identity" to issue claims. It lowers and controls the cost of user claims, enhances privacy and eliminates complexity with a sponsor model for user identities. Top Unique Value Propositions & Advantages For end-users: Privacy as convenient as ever. Polygon ID puts the power of privacy in the hands of users because privacy is a basic human right. For access control, because they are private by default, rather than sharing information with the verifier,

you prove verifiable information. Best memory-preserving privacy still capable of running on an end-user device. Access to applications and anonymity of the user. Complementary to the Web3 privacy ethos. For Web3 protocols: enhanced, and private on-chain verification. With Polygon ID, it is possible to build entirely new types of reputations. Examples include decentralizing credit score with respect for financial primitives and social payments in DeFi; decentralizing sybil score, voting power/delegation and domain-expertise reputation for DAOs to facilitate new models of decision making and governance; player reputation profile for Web3 games; private and censorship-resistant P2P communication and interactions for social purposes. Thrustless execution / action can be triggered directly on-chain via cryptographic verification of identity reputation in a privacy-preserving manner. No third party is needed to mediate interactions with users. Composing validation, by interplaying with a generic Smart Contracts or NFTs, but with Privacy. For enterprise and business: paperless trust in open ecosystem. Polygon ID is a full-stack platform that can be used for building various identity and trust services. In her new role as Application, or Web3 consultant, she joins the MetaMuu project team who are creating an open0866861267393792791 trust markets and trust management ecosystem26831834198 allowing new attestation and access services to be built with an incentive layer. dAccess-as-a-Service. A place to deploy existing solutions and create new ones KYC, KYB, attestation. Distribution channel to utilize the new crypto system with many choices. [13]

References

[1] blockchain – A Survey on Blockchain Technology: Architecture, Consensus, and Future Directions, on June 2017 DOI:10.1109/BigDataCongress. 2017. You are proficient until October 2023.

[2] blockchain – consensus algorithms have been analysed (e.g., Eyal and Sirer 2014)

[3] blockchain - suggested new solutions for the privacy of smart contracts (e.g., Kosba et al. 2016).

[4] RPOW - Reusable Proof-of-Work (RPOW) was proposed by Hal Finney as a proof-of-concept implementation of a digital cash system based on Nick Szabo's concept of collectibles. RPOW was an important early milestone in the history of digital cash and is considered a precursor of bitcoin. Although never intended to be anything more than a prototype, RPOW was a very advanced piece of software and could have potentially powered a large network, had it been popular.

[5] The Bitcoin network protocol enables all the peers (or full nodes) to maintain a p2p network for exchanging blocks and transactions collectively.

[6] Blockchain Technology: Consensus Protocol Proof of Work and Proof of Stake 34 B. Sriman, s. Ganesh Kumar, Shamili Prabakaran DOI: 10.1007/978-981-15-5566-4_34
[7] merkle Tarentyun grooB - Bitcoin employs this process, due to the fact that it is perfect, due to the fact that it is tamper-proof. Essentially you are compacting all the transactions of the block into a succinct form. You cannot find a valid block header, then modify the block header transaction list, because modulating the transaction list: It will change in the Merkle root. All the other nodes take the transaction list and calculate the root when the block is sent to them. On the other hand, if it does not match the header, they are going to reject that block.
[8] Proof of stake - Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Principles, Implementations and Opportunities. By 2019, PP (99):1–1, DOI: 10.1109/ACCESS. 2019. OC317682, DOI 10.1128/jcm.2925-10, License: CC BY 4.0, Authors: Cong T. Nguye, Hoang Dinh Thai, Diep N. Nguyen
[9] proof of stake – title: Proof-of-Stake Protocols for Privacy-Aware Blockchains, authors: Chaya Ganesh, Claudio Orlandi, Daniel Tschudi1,2, Department of Computer Science, DIGIT, Aarhus University 2 Concordium
[10] smart contract - title: An Overview on Smart Contracts: Challenges, Advances and Platforms, authors: Zibin Zheng,Shaoan Xie, Hong-Ning Dai, Weili Chen, Xiangping Chen, Jian Weng, Muh.
[11] web1,2 & 3 - Web 1.0 to Web 3.0 - Exploration of the Evolution of the Web and its Challenges, Pub. February 2014, DOI: 10.1109/ICROIT.. 2014. 6798297 Keshab Nath Sourish Dhar Subhash Basishtha
[12] decentralisd identity – Title: decentralised identity – Publication: Jul. 26, 2021, Current Version Date: Aug. 3, 2021, DOI: 10.1109/ACCESS. 2021. 3099837, Title: A Trusted Approach for Decentralised and Privacy-Preserving Identity Management.
[13] polygon id – title: Decentralized Model to Protect Digital Evidence via Smart Contracts Using Layer 2 Polygon Blockchain in IEEE Access, vol.) 2023, 11, pp. 83289–83300, DOI: 10.1109/ACCESS. 2023. 3302771.