International Telecommunication Union

ITU-T Technical Specification

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(03/2021)

ITU-T Focus Group on Environmental Efficiency for Artificial Intelligence and other Emerging Technologies (FG-AI4EE)

FG-AI4EE D.WG2-05

Guidelines on energy efficient blockchain systems

Working Group 2 – Assessment and Measurement of the Environmental Efficiency of AI and Emerging Technologies

Focus Group Technical Specification

Technical Specification ITU-T FG-AI4EE D.WG2-05

Guidelines on energy efficient blockchain systems

Summary

Several models have been introduced to calculate the urban energy system and to demonstrate the variants that calibrate the local energy efficiency. This Technical Specification focuses on the impact of blockchain in energy efficiency. More specifically, a literature analysis is performed with regard to the understanding of the blockchain energy demands and how these can be optimized.

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

Keywords

Energy efficiency; assessment; models; emerging technologies; AI; big data; smart and sustainable city.

Change log

This document contains Version 1 of the ITU-T Technical Specification on "*Guidelines on Energy Efficient Blockchain Systems*" approved at the ITU-T Study Group 5 meeting held online, 11-20 May 2021.

Editor: Leonidas Anthopoulos
University of Thessaly
Greece
E-mail: lanthopo@uth.gr

Editor: Ioannis Nikolaou
Fuelics
Greece
E-mail: ioannis.nikolaou@fuelics.com

Table of Contents

Technical Specification ITU-T FG-AI4EE D.WG2-05

Guidelines on energy efficient blockchain systems

Summary

Several models have been introduced to calculate the urban energy system and to demonstrate the variants that calibrate the local energy efficiency. This Technical Specification focuses on the impact of blockchain in energy efficiency. A literature analysis is performed with regard to the understanding of the blockchain energy demands and how these can be optimized. More specifically, the aim of this Technical Specification is to provide an overview of the energy demands of blockchain, to define the blockchain energy model and to depict the energy efficiency parameters that can be calibrated in order to enhance corresponding energy efficiency.

1 Scope

Energy efficiency is a crucial issue for present day and future city sustainability, especially due to the emerging appearance of smart cities (SC) and of cutting-edge technologies. Some emerging technologies, such as for instance blockchain and its role in cryptocurrency and contracting, may not take sustainability into consideration during their development. These technologies often require a huge amount of energy, leaving behind a significant environmental footprint. It is important to understand how to reduce the environmental impact of these technologies because this will contribute to the well-being of the market economy as well as to the quality of life of citizens and the users of these technologies[3]. In this regard, the definition of the blockchain energy requirements and of the means that can enhance blockchain energy efficiency would be useful. Thus, this work aims to define the blockchain energy efficiency model.

2 References

[1] Allessie, D., Sobolewski, M. and Vaccari, L. (2019*) Blockchain for digital government: An assessment of pioneering implementations in public services. EU Joint Research Center*.

[2] Guo, Y-M, Huang, Z-L, Guo, J., Guo, X-R, Li, H., Liu, M-L, Ezzeddine, S., Nkeli, M.J. (2021). *A bibliometric analysis and visualization of blockchain. Future Generation Computer Systems*, 116, pp. 316–332.

[3] ITU (2020), *U4SSC: Blockchain for smart sustainable cities. ITU Publishing: Geneva, Switzerland. Retrieved*, Jan. 2021 from http://www.itu.int/pub/T-TUT-SMARTCITY-2020-54

[4] ITU (2017), Distributed Ledger Technologies and Financial Inclusion. Technical Report. Retrieved, Jan. 2021 from https://itu.int/en/ITU-T/focusgroups/dfs/Documents/201703/ITU_FGDFS_Report-on-DLT-and-Financial-Inclusion.pdf

[5] Laurence, T. (2017*), Blockchain for Dummies*. John Wiley and Sons: New Jersey, USA.

[6] Nair, R., Gupta, S., Soni, M, Shukla, P.K., Dhiman, G. (InPress*). An approach to minimize the energy consumption during blockchain transaction. Materials Today: Proceedings*. https://doi.org/10.1016/j.matpr.2020.10.361

[7] Nakamoto, S. (2008), *Bitcoin: A Peer-to-Peer Electronic Cash System*. Retrieved, March 2021 from https://bitcoin.org/bitcoin.pdf

[8] Sedlmeir, J., Buhl, H.U., Fridgen, G. and Keller, R. (2020), *The Energy Consumption of Blockchain Technology: Beyond Myth*. Business & Information Systems Engineering, 62, pp. 599–608.

[9] Wood, G. (2018), *Why We Need Web 3.0* [online]. Retrieved, March 2021 from https://gavofyork.medium.com/why-we-need-web-3-0-5da4f2bf95ab

[10] Sandner, P (2020), *The Green Bitcoin Theory: How are Bitcoin, Electricity Consumption and Green Energy Related?* https://philippsandner.medium.com/the-green-bitcoin-theory-how-are-bitcoin-electricity-consumption-and-green-energy-related-b541b23424ab

[11] Buterin, V (2013), *Ethereum Whitepaper* https://ethereum.org/en/whitepaper/

[12] Recommendation ITU-T F.751.1 (2020), *Assessment criteria for distributed ledger technology platforms*. Retrieved, March 2021 from https://www.itu.int/rec/T-REC-F.751.1/en

[13] Recommendation ITU-T F.751.2 (2020), *Reference framework for distributed ledger technologies*. Retrieved, March 2021 from https://www.itu.int/rec/T-REC-F.751.2/en

[14] Recommendation ITU-T X.1400 (2020), *Terms and definitions for distributed ledger technology*. Retrieved, March 2021 from https://www.itu.int/rec/T-REC-X.1400-202010-P

[15] Technical Report FG DLT D1.2 (2019), *Distributed ledger technology overview, concepts, ecosystem*. Retrieved, March 2021 from https://www.itu.int/en/ITU-T/focusgroups/dlt/Documents/d12.pdf

[16] Recommendation ITU-T Y.4900/L.1600 (2016), *Overview of key performance indicators in smart sustainable cities*.

3 Terms and definitions

3.1 Terms defined elsewhere

This Technical Specification uses the following terms defined elsewhere:

3.1.1 smart sustainable city: An innovative city that uses information and communication technologies (ICTs) and other means to improve quality of life, efficiency of urban operation and services and competitiveness, while ensuring that it meets the needs of present and future generations with respect to economic, social, environmental, as well as cultural aspects[16].

3.1.2 block: Individual data unit of a blockchain, composed of a collection of transactions and a block header[14].

3.1.3 blockchain: A type of distributed ledger which is composed of digitally recorded data arranged as a successively growing chain of blocks with each block cryptographically linked and hardened against tampering and revision[14]. Blockchain is an open and shared distributed ledger technology (DLT), which can record transactions between two parties efficiently, permanently and in a verifiable way. It consists of a shared digital data storage, replicated and synchronized across multiple devices in a network. The main objective of DLT is to establish trust, accountability and transparency, with no reliance on a single source of authority or in environments where there is a lack of trust between actors. It also promotes decentralization and data integrity[3].

3.1.4 consensus mechanism (also called consensus protocol): Defines strict rules for creating new blocks and adding new data to them without favouring one participant over another[3].

3.1.5 distributed ledger: A type of ledger that is shared, replicated, and synchronized in a distributed and decentralized manner[14].

3.1.6 proof of work (PoW): Consensus process to solve a difficult (costly, time-consuming) problem that produces a result that is easy for others to verify[14]. The most common consensus mechanism requires complex mathematical resolution to generate a new block[3].

3.1.7 proof of stake (PoS): Consensus process, where an existing stake in the distributed ledger system (e.g., the amount of that currency that you hold) is used to reach consensus[14]. Consensus mechanism that depends on the validator's economic stake in the network[3].

3.1.8 proof of elapsed time (PoET): A consensus mechanism that requires participants' identification[3].

3.1.9 proof of authority (PoA): A consensus algorithm that does not require any mining activity[3].

3.1.10 hyperledger: a private and permissioned blockchain or in other words, a centralized or semi-centralized model. In this type of blockchain, it is possible to allow access and permissions just to a group of participants[3].

3.1.11 smart contract: A program written on a distributed ledger system which encodes the rules for specific types of distributed ledger system transactions in a way that can be validated, and triggered by specific conditions[14]; Software program that is executed automatically and capable of carrying out the terms of the agreement between parties without the need for human intervention[3]; Pieces of software that execute a specified action based on the state of the system or a transaction that occurs[1].

3.1.12 stateful contract: A contract with specified states[14].

3.1.13 stateless contract: A contract lacking specified states[14].

3.1.14 token: A digital representation of value on a shared distributed ledger that is owned and secured using cryptography to ensure its authenticity and prevent modification or tampering without the owner's consent[14].

3.1.15 transaction: Whole of the exchange of information between nodes. A transaction is uniquely identified by a transaction identifier[14].

3.2 Terms defined here

This Technical Specification defines the following terms:

3.2.1 distributed ledger technology (DLT): A new type of secure database or ledger that is replicated across multiple sites, countries, or institutions with no centralized controller. In essence, this is a new way of keeping track of who owns a financial, physical, or electronic asset.

A technology that facilitates an expanding, chronologically ordered list of cryptographically signed, irrevocable transactional records shared by all participants in a network.

In other words, DLT refers to the protocols and supporting infrastructure that allow computers in different locations to propose and validate transactions and update records in a synchronized way across a network[4,1].

3.2.2 cryptography: Developed from safety communication technology, which is a combination of mathematics, computer, and information theory. Public key cryptography (also called asymmetric key) is a milestone in the development of modern cryptography, which mainly includes public keys and private keys.

3.2.3 cryptocurrency miners: Special transaction nodes that aggregate the outgoing transactions in the single block and are responsible for the validation process. They compete amongst each other to solve a cryptographic problem and gain the right to add the formatted block in the existing ledger of blockchain transactions.

4 Abbreviations and acronyms

This Technical Specification uses the following abbreviations and acronyms:

AI Artificial Intelligence

CPU Central Processing Unit

DLT Distributed Ledger Technology

FPGA Field-Programmable Gate Array

GPU Graphic Processing Unit

P2P Peer-to-Peer

PBFT Practical Byzantine Fault Tolerance

PoA Proof of Authority

PoET Proof of Elapsed Time

PoS Proof of Stake

PoUW Proof of Useful Work

PoW Proof of Work

PUE Power Usage Effectiveness

SDN Software Defined Networking

SGX Software Guard Extension

V2G Vehicle-to-Grid

5 Background

5.1 Blockchain mechanism in brief

Blockchain[1, 12, 13, 15] finds its origins in a paper published by an anonymous (group of) author(s) called Satoshi Nakamoto[7], who introduced a crypto-currency named *Bitcoin*. The idea of a Bitcoin was introduced as a purely peer-to-peer (P2P) electronic transaction network. This network allows direct financial transactions instead of using a financial institution as a trusted third party. To simplify, blockchain technology allows two actors in the system (called nodes) to transact in a P2P network and stores these transactions in a distributed way across the network. It registers the owners of the assets that are transacted and the transaction itself. A transaction is verified by the network with a 'consensus mechanism', which allows users in the P2P network to validate the transactions and update the registry in the entire network. A consensus mechanism is used to establish trust in the accuracy of the data in the system, which is traditionally established by an intermediary or an administrator in a centralized system. As such, the blockchains are composed of the following three core parts[5]:

– **Block**: A list of transactions recorded into a ledger over a given period. The size, period, and triggering event for blocks is different for every blockchain. Transaction can be seen as recording of data. Assigning a value to it (such as happens in a financial transaction) is used to interpret what that data means.

– **Chain**: A hash that links one block to another, mathematically "chaining" them together. The hash in blockchain is created from the data that was in the previous block. The hash is a fingerprint of this data and locks blocks in order and time.

– **Network**: The network is composed of 'full nodes'. Nodes can be seen as computers running an algorithm that is securing the network. Each node contains a complete record of all the transactions that were ever recorded in that blockchain.

The *consensus mechanism* on the other hand, is a process by which, nodes in a distributed network agree on proposed transactions. This mechanism provides a way to record information in the ledger in a manner that ensures data integrity, immutability and consistency. Consensus mechanisms are distributed network governance rules and protocols that enable the recording, completion and execution of transactions under certain conditions. Therefore, a consensus can be built upon the previous transaction, forming a sequence of transactions, similar to a ledger. In blockchains, multiple transactions are clustered into a block which mathematically refers to the previous block. In the case of Bitcoin, after a set time, a new block is created with the occurred transactions included in the block and validated across the network. This forms a chain of blocks: hence the name 'blockchain'.

Blockchain is an open and shared distributed ledger technology (DLT) and was just the computer science term for how to structure and share data or in other words a novel approach to the distributed database. It also promotes decentralization and data integrity. The innovation of blockchain comes from incorporating old technology in new ways. It is a decentralized and unreliably distributed database technology[2, 4] that a group of individuals controls, stores and shares information[5]. Another definition for blockchain suggests *a data structure that makes it possible to create a digital ledger of data and share it among a network of independent parties*. The main objective of blockchain as a DLT is to establish trust, accountability and transparency, with no reliance on a single source of authority (trusted third party) or in environments where there is a lack of trust between actors. The removal of central authority from database structure is one of the most important and powerful aspects of blockchains[5].

When data is recorded in a blockchain, it is extremely difficult to change or remove it. When someone wants to add a record to a blockchain, also called a transaction or an entry, users in the network who have validation control verify the proposed transaction. This is where things get tricky because every blockchain has a slightly different spin on how this should work and who can validate a transaction. Figure 1 illustrates a simplified data structure and the main elements in a blockchain.

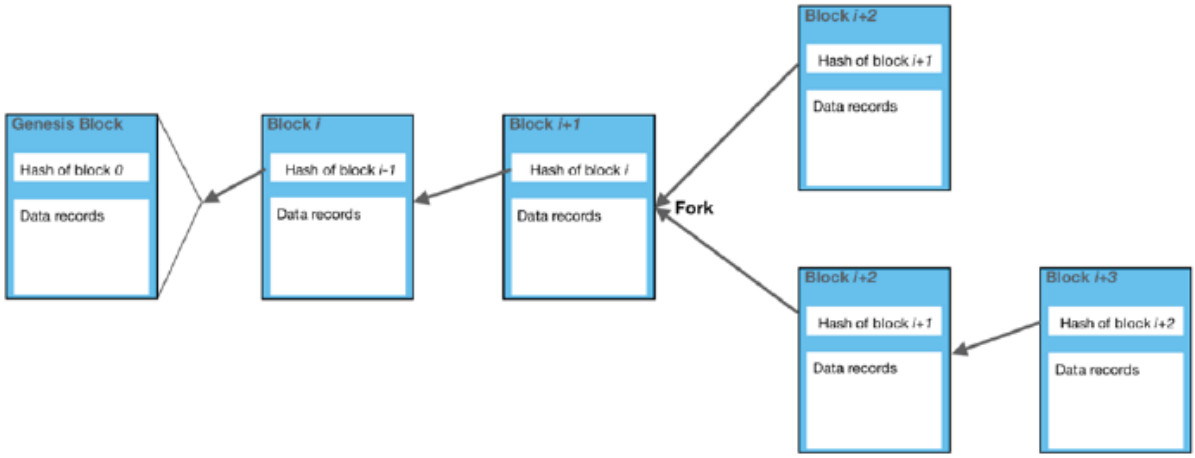

Figure 1 – Simplified data structure[3]

The mechanism used to discernibly relate the blocks is called the *hash functions*, which consists of cryptographic functions that map a bit string of arbitrary length to a fixed-length bit string in such a way that it[3, 15]:

1 is computationally infeasible to find any data input that maps to any pre-specified output (i.e., digest);

2 is computationally infeasible to find any two distinct data inputs that map to the same output; and

3 the smallest change of input, even a single bit, will result in a completely different output.

Building a blockchain from the ground up Although the mechanism and architecture of the various blockchain flavours are well documented, many of the core concepts of this technology are not well understood and, in some cases, misunderstood by non-blockchain experts. Equally importantly, the reason behind some of the design choices of the blockchain technology are not clear without extensive experience in this field. In the next few clauses, we will go through the process of creating a fictional, simplified, blockchain from the ground up. During this process we will discover the reasons behind some of the fundamental design choices of the blockchain architecture and clarify in a non-technical fashion the terms that will be used later in the discussion about the blockchain energy consumption.

5.1.1 State

Our use case starts with Alice, Bod, Chuck and Dave who decide to provide a service to give anyone the ability to store information in such a way that it cannot be modified. None of them trust each other, and they also assume their users do not trust anyone. They agree on an initial state of their service (***S₀***) and a protocol (***Π***) to use to talk to each other to ensure everyone has the same understanding of the state at any point in time, see Figure 2.

Erin and Frank, two users of this service ask to change the state with new information $S_0 \rightarrow S_1 \rightarrow S_2$ with the last state containing the value of the three variables x, y and z.

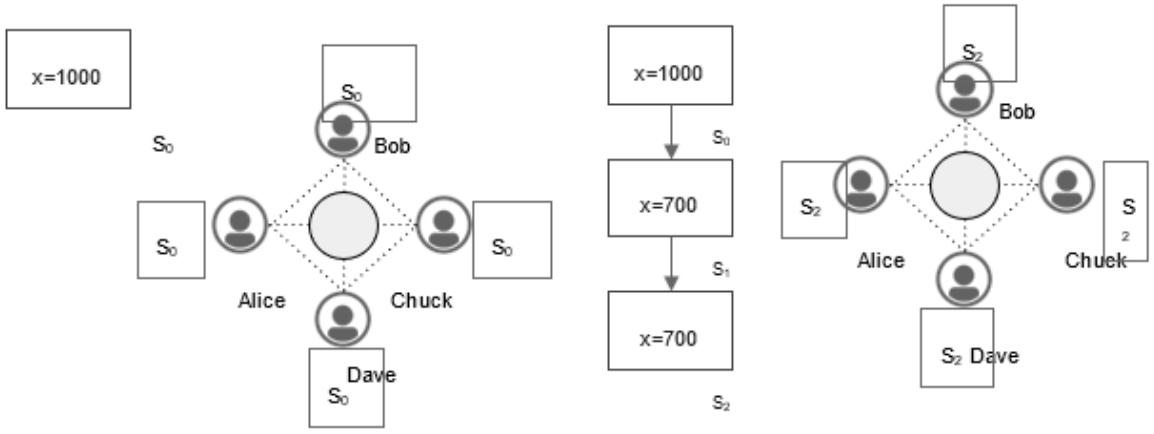

Figure 2 – The transaction scenario

Yet another external user, Grace, asks for the value of the state variable *x* and the service responds with *x = 700*. Due to the lack of trust among all participants Grace however has no way of verifying that this value is correct as the service stores only the latest version of the state. The state service needs improvements.

5.1.2 Chain of state changes

Alice, Bob, Chuck and Dave decide to improve their service by storing information about the state changes. In this way, any external user would be able to verify that the latest state is valid by going through all the state changes. The easiest solution would be to store the complete state each time it changes but that would not scale well, as the size of the state increases.

A better option is to store only the initial state and then store just the changes to that state. The order of the state changes is critical to reach the correct last state, therefore they also store in each state change a link to the previous, parent state change (P = S*i*). This way any external user would be able to verify the latest state simply by starting from the initial state and "replaying" all state changes with the right order. They agree on these changes and implement them in their protocol *Π*.

When Grace asks for the value of x, the service responds with x = 700 as before. If Grace wants to validate that this is correct, she can get the whole sequence of state changes from any of Alice, Bob, Chuck or Dave (see Figure 3) and confirm that the value of x is 700.

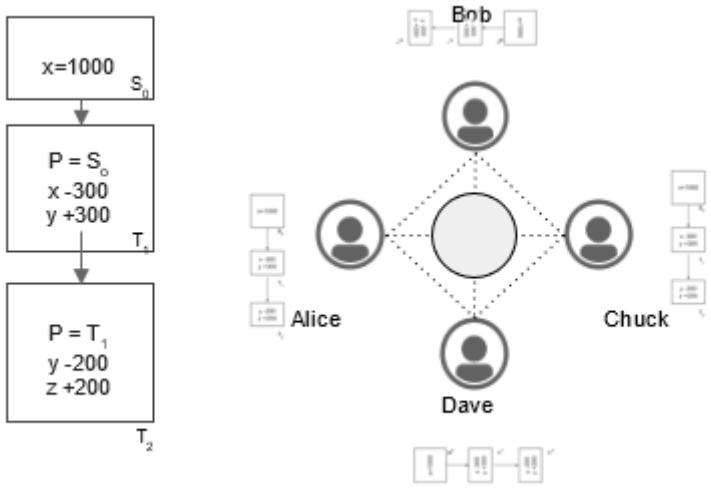

Figure 3 – Chain of state

However, Grace is still suspicious that any of the state changes may have been modified by any or all members of the group so she asks for proof that each state change has not been tampered with. To provide further assurance Alice, Bob, Chuck and Dave (we will be calling them the "network" from now on) decide to use the value that links the sequence of state changes in a way that not only points to the right parent but it can also be used to validate that the parent state change has not been modified (see Figure 4). The tool they use to achieve this is a cryptographic hash function[7]. When a new transaction is performed and added to the chain, they link it to the previous one using the hash of the previous transaction, which in turn includes the hash of its parent and so on. In these terms, when a state change is modified, the hash of this state becomes invalid and so does the next stage change that includes it.

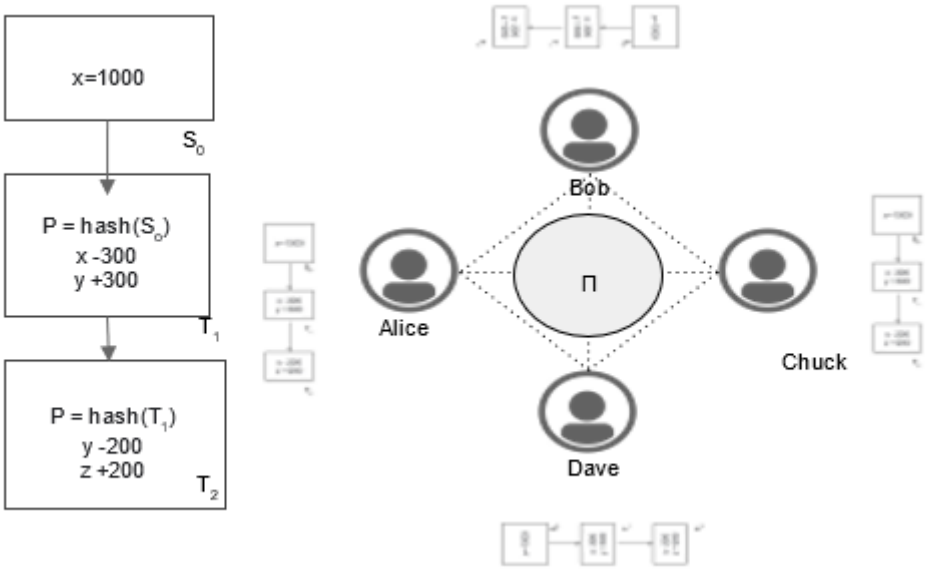

Figure 4 – Chain of state change

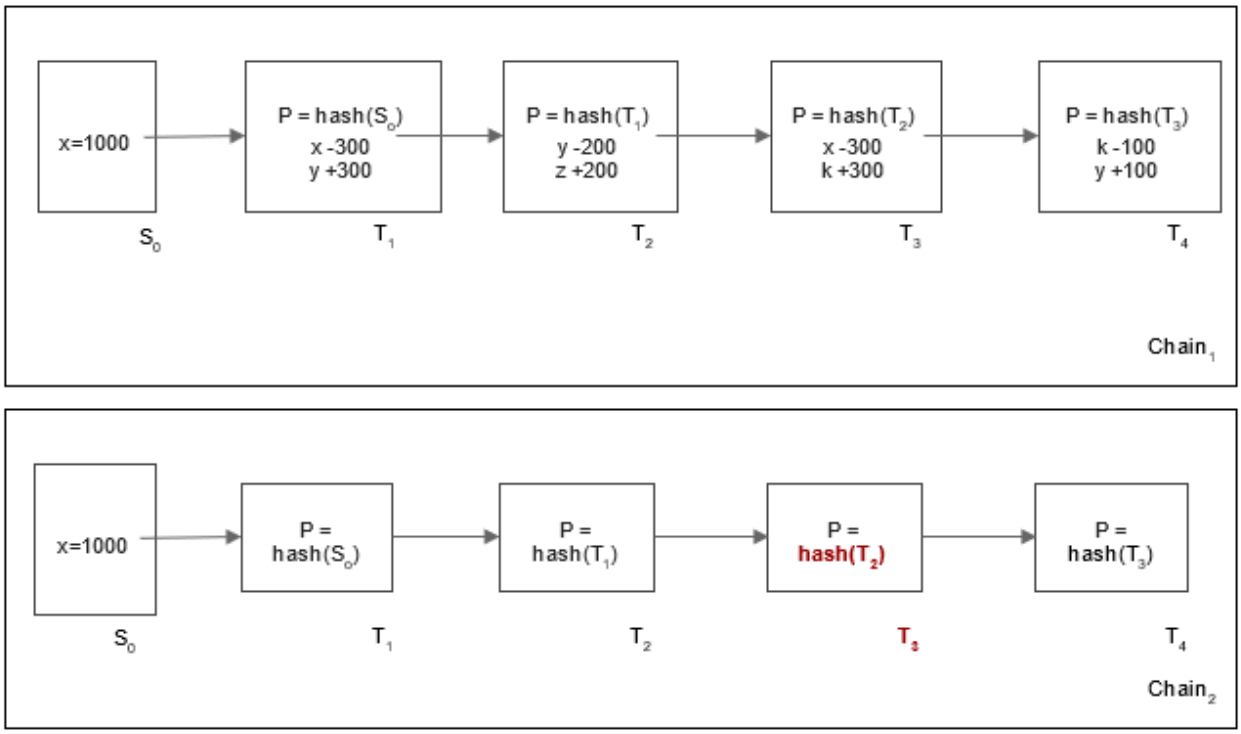

Figure 5 – An example of state validation

In the example shown in Figure 5, if the T_2 contents are modified, the parent hash of the subsequent transaction is automatically invalidated. Now anyone who wants to validate the current state can do so by:

- starting from the initial state;
- for each state change;
- calculate the hash of the parent transaction;
  - compare it with the parent hash stored in the current state change4
  - replay the state change.

Following these steps, Grace can confirm that the valid chain of state changes is *Chain*$_1$ and value for *y* after transaction T_4 is $y = 200$ and the value of $y = 100$ that results from *Chain*$_2$ is not valid.

5.1.3 Block chain

As the number of transactions grows, this process does not scale well, so the network decides to bundle a sequence of state changes in blocks. Following the same principles, each block of state changes includes a hash of the previous block so that it can establish both the block sequence and the block validity. Within each block the state changes are stored as before (see Figure 6).

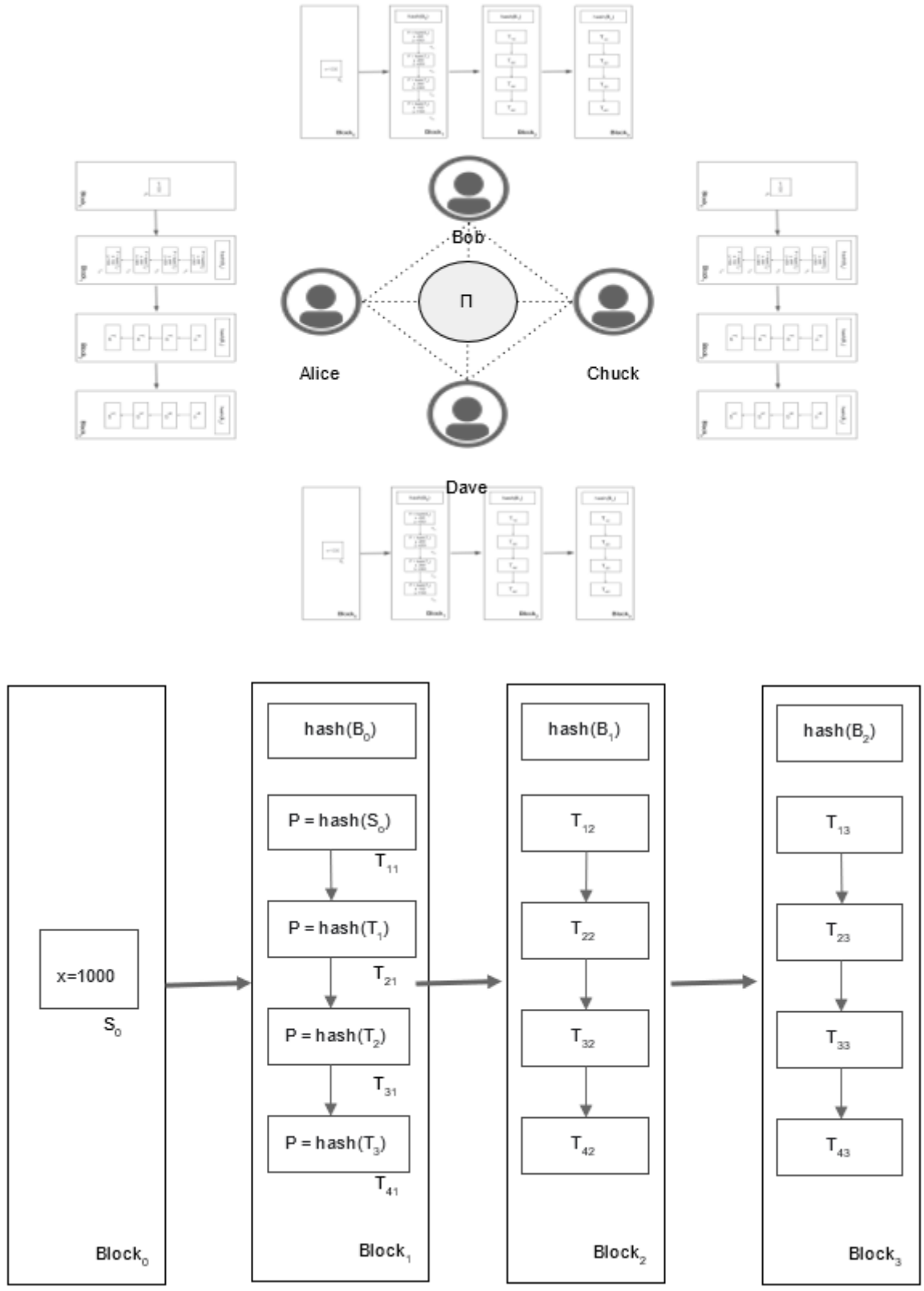

Figure 6 – A chain of blocks

When a state change is modified, the block's internal sequence of state changes is invalidated which in turn invalidates the whole block and the next block. Now, when Grace receives the state block changes from Chuck and runs the verification algorithm, when she arrives at $Block_2$ she cannot accept

it because the hash of the modified $Block_1$ is different to the one stored in $Block_2$ as the parent hash (see Figure 7).

However, nothing prevents Chuck from altering the parent hash of $Block_2$ to match the hash of the modified $Block_1$. This in turn would invalidate $Block_3$ but, again, nothing prevents him from modifying the parent hash of $Block_3$ to match the new hash of $Block_2$. So if Chuck, or anyone else in the network, is willing to go through this trouble, he can modify the hashes of all blocks following the modified $Block_1$ and create a new block chain that is valid.

Now when Grace asks for the block chain and applies the verification algorithm, she will end up with a different final state depending on which node she contacted and will have no way of knowing which one is the correct one. Note that from Grace's perspective, a majority rule would not be sufficient as the fact that three out of the four network nodes provide the same block chain does not necessarily mean that it is the right block chain. It is equally plausible that Alice, Dave and Bob collaborated and decided to alter $Block_1$ after it was created, and it is actually Chuck that has the only replica of the block chain that is not modified. Grace still has no way of knowing if the correct value for y is 100 or 200 (see Figure 8).

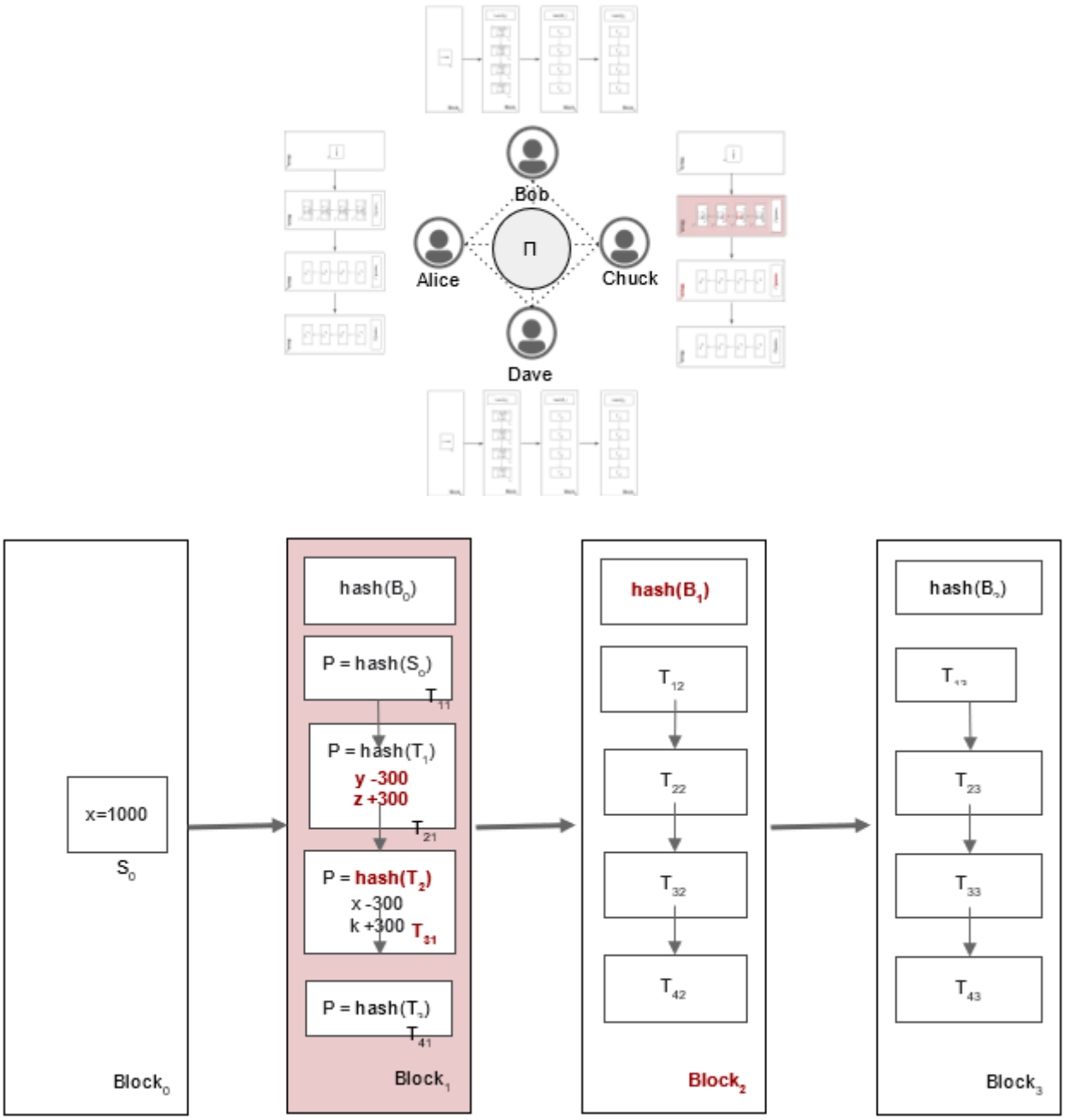

Figure 7 – A change in a chain of blocks

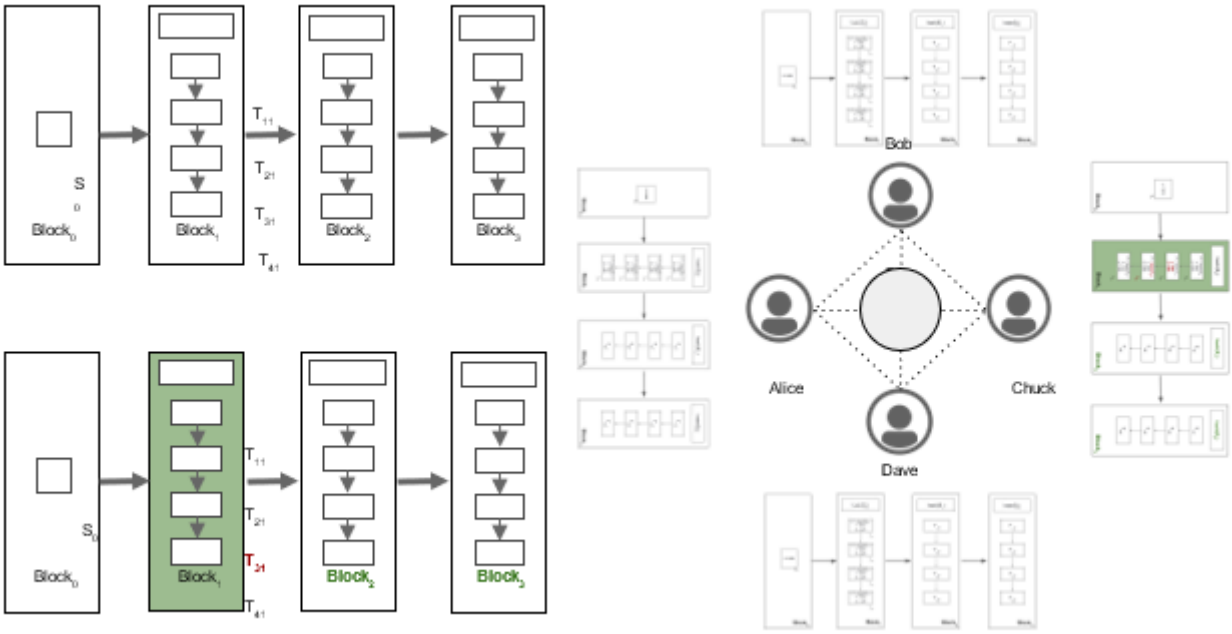

Figure 8 – The validation process

5.1.4 Proof of work

In order to improve the network's credibility, Alice, Bob, Dave and Chuck agree to change their protocol to make it harder for any network participant to change any past block in a way that it will pass the validation rules and not be detected. The first step is to make the creation of a block of state changes require a lot of work but keep its verification as easy as before. The second step is to have each member of the network accept by definition that the chain of blocks that has the most work spent on it is the valid chain.

The fact that each block is connected to the previous one and a change in a past block invalidates the chain of blocks after that means that if a malicious network member modifies a past block, he will have to spend the work needed to create that block and then create at least as many blocks after that as the length of the current longest chain hoping to get the rest of the network nodes to accept his version of the state.

5.1.4.1 A cryptographic puzzle

The mechanism that the network will use to make the block creation harder, is to put an arbitrary requirement on the value of the hash of each block: The hash of a block has to be less than a specific number or, equivalently, the hash of the block has to start with a specific number of zeros[1].

As the hash of the block's content is given and the probability it conforms with this rule is practically zero, the block creator is allowed to introduce a random number within the block, known as "nonce", that will lead to the generation of a different hash value for the block. It is impossible to predict beforehand what the hash will be, so it is impossible for the block creator to select this random value to generate the hash of the block that conforms with the rule. The only way to achieve this is by trying

---

1 https://www.blockchain.com/btc/block/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
https://www.blockchain.com/btc/block/00000000001452cf808a54c268874dcb3deb37edd514034b0f55ec261e8f2097
https://www.blockchain.com/btc/block/000000000000000000009af9003cdb916613c4267286b6808e4c8eda62e0e2f1e?page=1

different nonce values until it finds the one that leads to the hash of the block having the desired properties (see Figure 9).

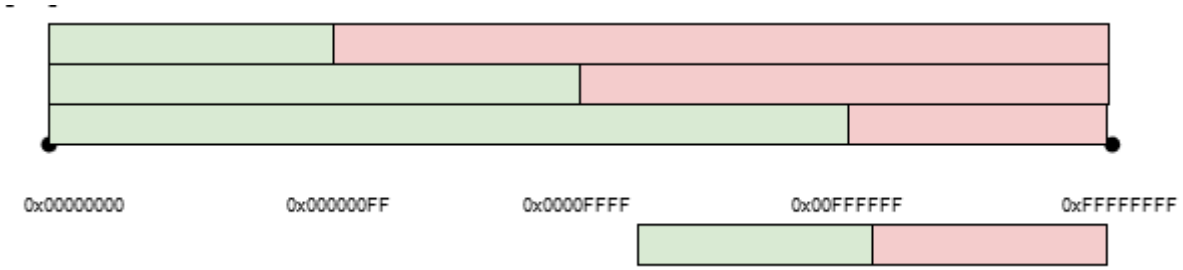

Figure 9 – The cryptographic puzzle

The hash values distribution is uniform so the number of hash calculations it will take to find the "right" hash depends on the selection of this number, or equivalently on the selection of the number of zeros at the beginning of the hash. The more zeros, the harder it will be to find a "right" hash and the block creator has to do more work to find it. The number of zeros is therefore equivalent to the "difficulty" of the block creation (see Figure 10)

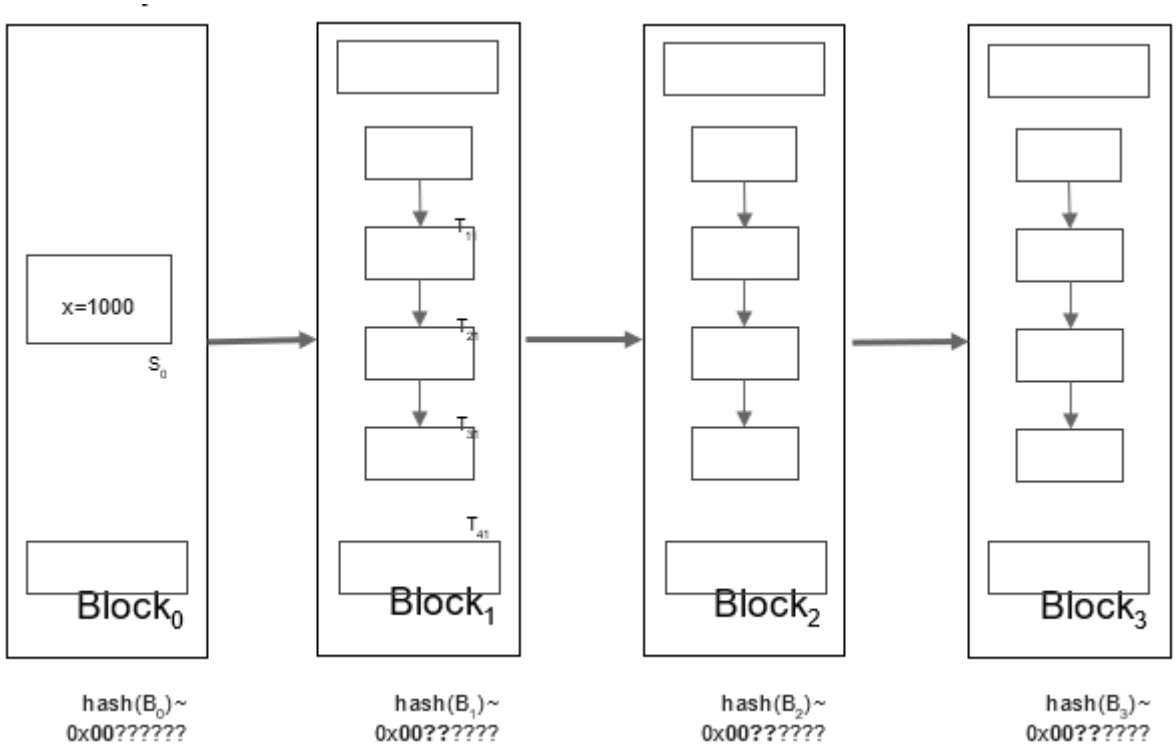

Figure 10 – Inserting zeros in the hash

Given the number of hashes per second the block creators can calculate, the network can adjust the block creation difficulty so that the time it takes to find the hash and create the block is within a desired time interval.

In case Chuck decides to modify $Block_1$ as before, he needs to expend the work needed to calculate new values for nonce1, nonce2 and nonce3. While doing these calculations, the rest of the network members will be working on the right chain and will have created new blocks making it even more difficult for Chuck to catch up.

Now Alice, Bod, Dave and Chuck can detect if any of them is tampering with the block chain. As long as they can verify the chain is correct and that they have the longest chain, they can be certain that the state of the system is correct.

Going back to Grace, she can now ask the network for the longest block chain and apply the verification algorithm. If the block hashes are correct and she has the longest replica of the block chain, she can be certain that the values stored have not been modified.

5.1.5 Blockchain consensus

The proof-of-work (PoW) algorithm described above is the key that allows the network to agree on the state of the block chain and at the same time to be certain that it cannot be changed without the need for any assumptions of trust among the network members. It belongs to a family of algorithms called "consensus algorithms". In essence, the blockchain consensus is:

A single opinion of what happened, when it happened and what should happen because of it.

5.1.6 Block finality

Now that Grace can finally trust the network Alice, Bob, Chuck and Dave have created, she decides to add her state change to it. She creates a request and publishes it to the network. Her request is added to a pool of pending state changes coming from other users, like Erik and Frank. A network member, for example Alice, picks it up, together with other state changes and starts the process to create a new block by looking for the nonce that will lead to the block's hash that meets the agreed criteria.

At the same time, every other member of the network is also trying to create a block with a subset of the pool of state changes that may or may not include the change Grace requested. Let us say that Alice was lucky, manages to find the nonce first and creates the new block. Can Grace now be certain that her state change is irreversibly stored in the blockchain?

Actually, she cannot. The reason is that it is possible for another member of the network to be luckier. For example, Bob could create a new block roughly at the same time Alice did and then create another one before Alice. Bob's version of the blockchain will be one block longer and the network will have to accept this as the version to continue adding blocks as agreed in the protocol. But Bob's version of the blockchain may not have included the state change of Grace yet as it may be still in his pool of pending requests.

The only way Grace can be relatively certain that her state change is irreversibly written on in the block chain is if she confirms it in a block that has at least a few more blocks after it. The more blocks, the higher the certainty. The state of the blockchain when a block can be considered final is called blockchain finality.

5.2 A taxonomy of blockchains

The blockchain we designed from scratch can be categorized as a *permissionless*, PoW block chain. There are two dimensions we can use to categorize a block chain: (a) the *access control* and (b) the *consensus algorithm*.

(a) According to access control, the blockchains are classified as follows[15]:

Permissionless blockchains: In a permissionless blockchain anyone is allowed to join the network and create blocks. The only requirement is that they have to follow the rules of the agreed protocol. They are also labeled *public blockchains* (i.e., Bitcoin), which are large, distributed networks that are run through a native token.

Permissioned blockchains: In a permissioned blockchain (i.e., Ripple) an external authority decides who can create blocks. There are control roles that individuals can play within the network. They are still large and distributed systems that use a native token. Their core code may or may not be open source. A class of this category *private blockchains*, tend to be smaller and do not utilize a token.

Their membership is closely controlled. These types of blockchains are favoured by consortiums that have trusted members and trade confidential information.

Table 1, Figure 11 and Figure 12 below summarize the pros and cons from the different types of blockchain[3].

Table 1 – Comparison of the alternative blockchain types

Blockchain type	Description	Examples
Public permissionless blockchains	Open to everyone with an Internet connection to participate in the blockchain consensus mechanism, to transact and observe the full transaction log	Bitcoin Litecoin Ethereum
Public permissioned blockchain	Allows everyone with an Internet connection to see the transaction log, but only a restricted number of participants can contribute to the consensus mechanisms	Ripple Private version of Ethereum
Private permissioned blockchain	Restricts transactions and access to view the transaction log to the participating nodes in the system. The architect (or owner) of the blockchain is able to determine who can contribute to the blockchain system and which nodes can participate in the consensus mechanisms	Rubix Hyperledger
Private permissionless blockchain	Restricted in terms of who can transact and see the transaction log. The consensus mechanism is open to anyone	Exonum (Partially)

Figure 11 – A centralized VS a distributed/decentralized transaction system

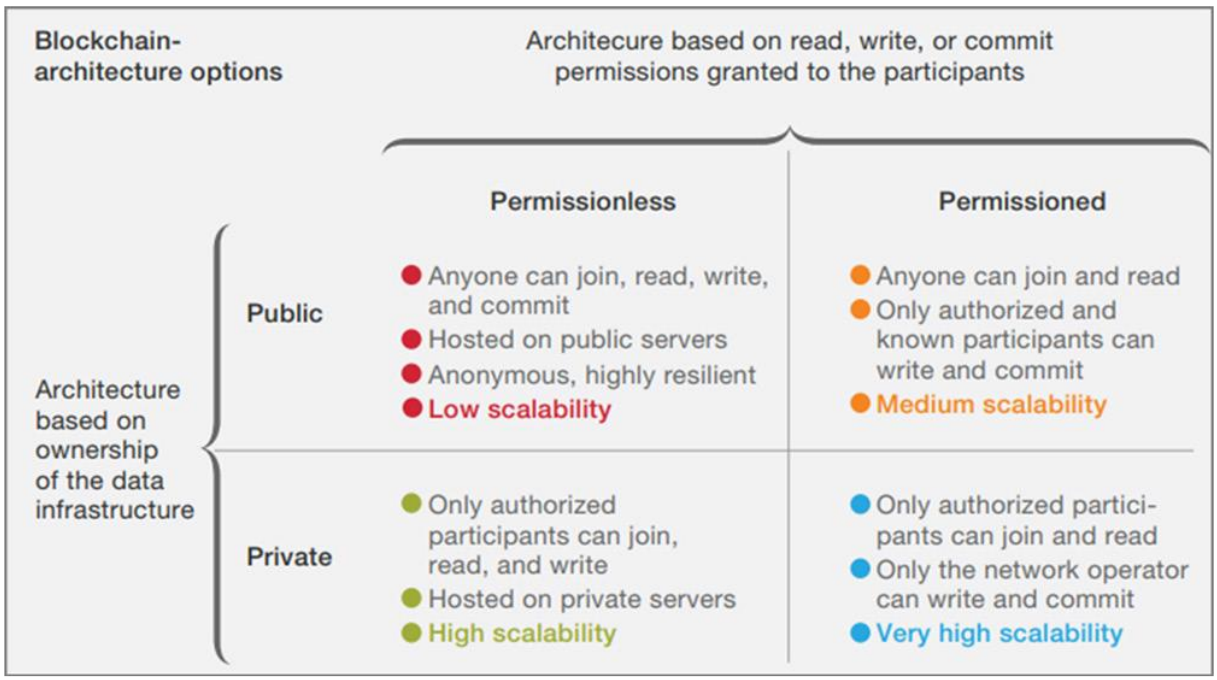

Figure 12 – Blockchain architecture options and differences[3]

(b) According to consensus algorithms, the blockchains are classified as follows:

Proof of work (PoW): This is the algorithm have we briefly described so far. It is essentially about making the creation of blocks hard enough to make it practically impossible for anyone to tamper with the data stored on the block chain. In a PoW blockchain the nodes that create blocks are referred to as "miners" and the block creation is referred to as "mining".

Proof of state (PoS): In this consensus algorithm each node that wants to participate in the creation of a block will have to deposit an amount as insurance that he will "play by the rules". If a node fails to do so and compromises the consistency of the blockchain, the deposit is lost. This way each node that creates blocks has a "stake" in the success of the blockchain. The higher the deposit, the higher the incentive to ensure the blockchain works as expected.

The consensus algorithm selects randomly which node will create each new block taking into account the stake it has in the system. Once selected, the node simply validates the state changes and creates the block without the need to do any additional work as in PoW. The protocol then requires additional validation for the network nodes before accepting the block in the blockchain. In a PoS blockchain the nodes that create blocks are referred to as "validators" or as "forgers" and the block creation is referred to as "minting".

Proof of authority (PoA): This is similar to the PoS consensus algorithm with the difference that in order to become a validator one needs to be accepted by a centralized authority and not a stake on the system. This approach minimizes the energy demands of a blockchain[3].

Proof of elapsed time (PoET): A consensus algorithm that requires participants' identification, which means that it is more common in a permissioned style blockchain than a public one due to efficiency reasons. PoET prevents high resource utilization, energy consumption and operational efficiency[3].

Due to the way each consensus algorithm is designed, they may be more tailored to the permissionless or the permissioned access control of the blockchain. Table 2 summarizes the combinations that are practically used:

Table 2 – Combinations of access control and consensus algorithms

Consensus algorithm access control	PoW	PoS	PoA
permissionless	x	x	
permissioned		x	x

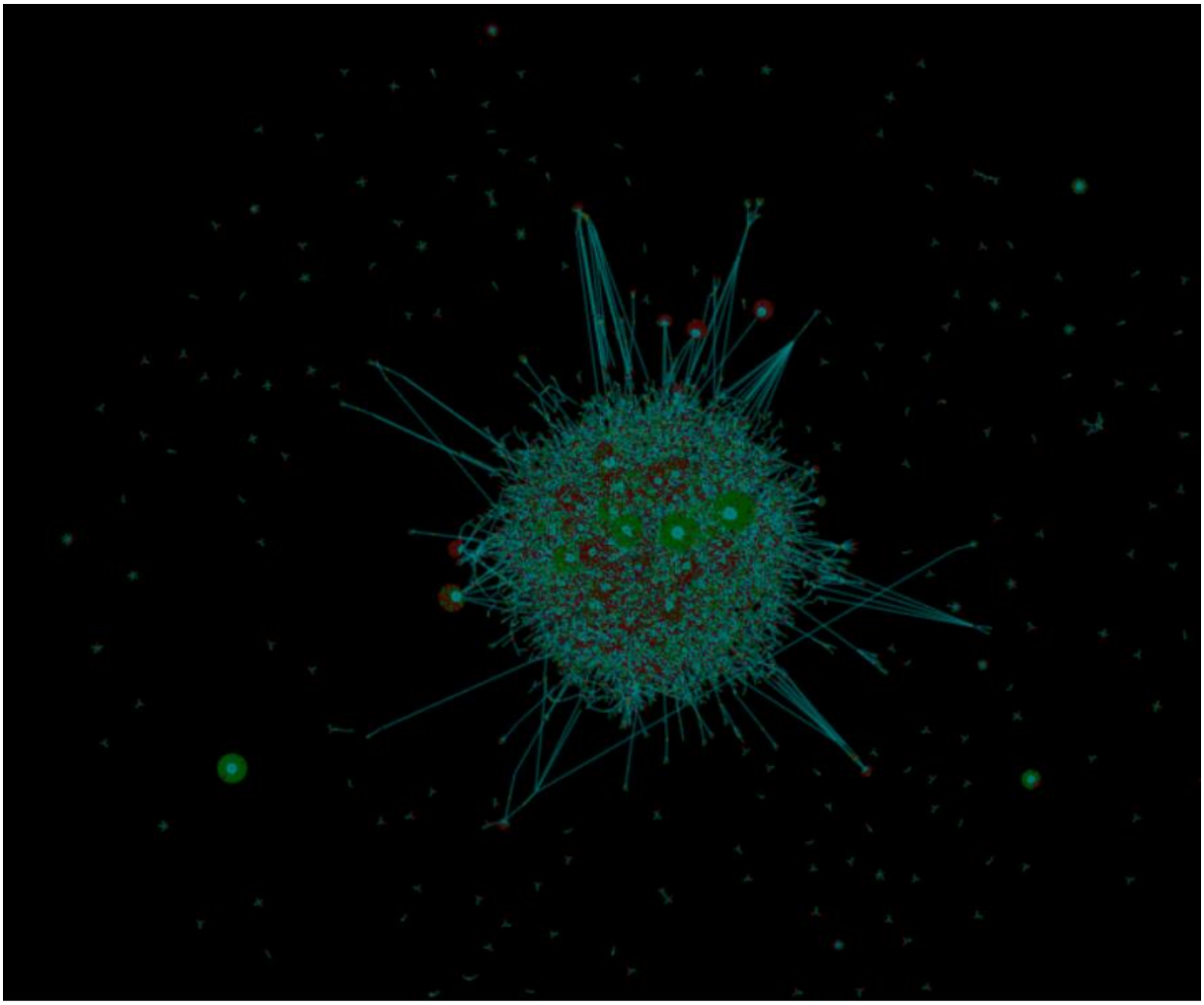

Figure 13 – Bitcoin blockchain network (captured from http://dailyblockchain.github.io)

5.3 Block chain applications

5.3.1 Cryptocurrency

To prevent the network from being corrupted, not only are blockchains decentralized but they often also utilize a cryptocurrency. Cryptocurrencies are a decentralized subset of digital currencies, based on a set of algorithms and protocols that enable a peer-to-peer, cryptographically based payment mechanism, a medium of exchange and a store of value, the best-known example being bitcoin. A cryptocurrency is a digital token that has a market value. A token is a digital item which represents either the right to perform some operation or a physical object of value[1,5].

Cryptocurrencies are traded on exchanges like stocks. Cryptocurrencies work a little differently for each blockchain. Basically, the software pays the hardware to operate. The software is the blockchain

protocol. Well-known blockchain protocols include *Bitcoin*[2]*, Ethereum, Ripple, Hyperledger, and Factom*. The hardware consists of the full nodes that are securing the data in the network[51]. On the other hand, *Kusama*[3] is an emerging cryptocurrency that is based on PoS consensus mechanism. Figure 13 depicts a Bitcoin blockchain network.

A recent bibliometric analysis with regard to blockchain[2] shows that *smart contract* is the hottest topic in the field, followed by the *IoT*, *bitcoin*, *security* and *Ethereum*. The concept of *smart contracts* means *embedding contracts in various valuable and digitally controlled properties*. From a technical perspective, the smart contract can be regarded as a computer program, which can independently execute the provision of the contract[2]. *Bitcoin* is defined as a *digital currency that can be recorded after each transaction on the bitcoin network* and considered as the budding and explosive stage of blockchain technology. Ethereum is a workshop based on state machine transactions written in a Turing-complete language. *Ethereum* is a representation of running smart contracts but it is also associated with cryptocurrency[2].

The applications of blockchain in smart cities and in energy sector (i.e., smart grids; energy contracts, etc.) attract a lower scholars' interest according to the bibliometric findings. The *smart grid* is based on an integrated, high-speed two-way communication network that manages the power through real-time information exchange by an interaction between the power producers and the consumers. The combination of blockchain technology and AI could be used to enhance the utilization of energy from the grid steadily, efficiently, and reliably[2]. *Energy blockchain* is an emerging trend and has been associated blockchain with terms like game theory; consortium blockchain; transactive energy; adaptive aggressiveness strategy; distributed generation; private blockchain; consensus protocol; markets; auctions; and continuous double auction[2]. This association is more likely to prioritize blockchain with ensuring the energy trading (flows and market), instead viewing the energy demands and efficiency of the blockchain technology. More specifically, literature evidence shows a localized P2P electricity trading system using the consortium blockchain (PETCON) method to improve transaction security and privacy protection; edge service framework based on blockchain to assure secure energy trading in the software defined networking (SDN) – enabled vehicle-to-grid (V2G) environment[2].

5.4 Blockchain benefits

Blockchain benefits can be summarized as follows[1]:

1 A distributed ledger shares content across multiple parties. This shared nature makes transactions easily trackable and fully disclosable even in large and complex ecosystems.

2 The physical decentralization of the storage of transaction details is argued to provide security integrated into the design of the technology stack. This feature eliminates the risk of a single point of failure, where one node is critical for the operation of the network and vulnerable for cyber-attacks.

3 New entries are recorded in an append-only manner and linked to the previous transactions. The entries cannot be changed, which safeguards data integrity on the ledger.

4 Transactions are verified via a peer-to-peer consensus mechanism ensuring a common truthful ledger. Centralized parties are no longer needed to assure transaction validity. As a consequence, blockchain shifts power from an intermediary towards the ecosystem. This decentralization of control and power establishes ownership of the nodes and introduces checks and balances ingrained in the technology stack.

---

2 https://decrypt.co/42427/bitcoin-blockchain-grows-to-300-gigabytes-in-size
https://www.statista.com/statistics/647523/worldwide-bitcoin-blockchain-size/

3 https://thousand-validators.kusama.network/#/

5 The combination of a distributed, append-only ledger and a consensus mechanism is argued to present disintermediation: the elimination of middle-men or brokers and remove any middle-men or broker-related transaction costs.

6 Blockchain and energy efficiency

Blockchain is one of the leading technologies in recent times, but at the same time it consumes an extensive amount of energy[6] for computation, storage and synchronization. Blockchain is considered as the secure public chain for transactions, and it assumes that the miners involved in a transaction do not consume much energy (see Figure 14).

During the blockchain process, proof of work (PoW) relies on the network resources consumption for protection from malicious attackers. No intermediaries are involved during this P2P transaction in blockchain, which means that the transactions require a huge amount of hash calculations for achieving the best results. A considerable amount of energy is wasted during these transactions in the form of electricity, which degrades the performance and in this regard the efficiency of blockchain.

Blockchain runs on digital networks in which data transmission is taking place by copying data from one place to another. In cryptocurrencies for instance, a digital coin is copied from one wallet to another and it gives assurance that there must be single coin spending. Distributed voting is another case, where each member in the network can compare versions of the ledger. A user can trace the past history of the system transaction and check their validity; this enables a high level of transparency. This validation process is done by the distributed consensus algorithm. Distributed nodes with collaborative attributes are established by game-theoretic incentives or rewards[6].

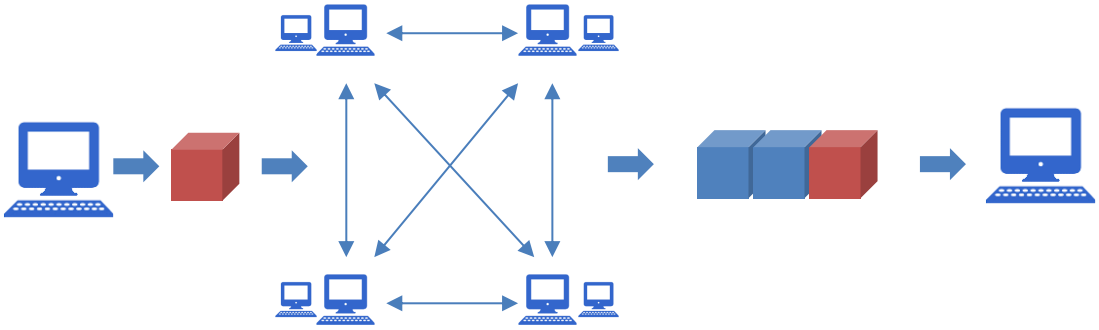

Figure 14 – A blockchain transaction

In the case of bitcoin transactions, a bitcoin's address is generated from the user's public key, through which the user is identified. Transacting parties must know each other's public addresses before the payment transaction. The sender digitally signs and transfers the coin to the receiver via a transaction, which contains the information related to the number of coins traded and the address of the transacting parties with receiver's address in encrypted form. During a transaction (Figure 4, Figure 5) special nodes aggregate the outgoing transactions in the single block and are responsible for the validation process. This process takes an average of 10 minutes for block validation and inclusion in the blockchain. Validator nodes are known as miners and play the most important role in the whole blockchain process: they compete with one another to solve the cryptographic problem and gain the right to add the formatted block in the existing ledger of blockchain transactions. Miners who got the right to add a block in the blockchain will receive the financial reward award in two steps: the first one is a reward that is finalized by the agreement of all the network members which is approximately 12.5 bitcoins nowadays and transaction fees that is offered by transacting parties. A blockchain process is shown in Figure 15.

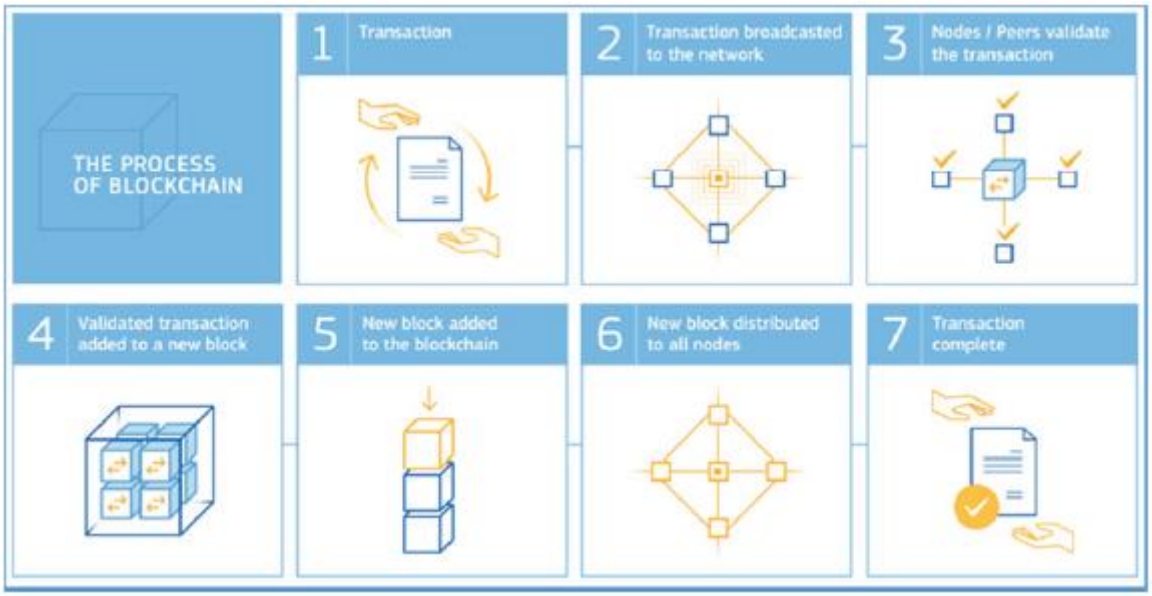

Figure 15 – A blockchain process[3]

Both PoW and PoS methods' usage depends on the circumstances and the transaction size. Random selection is used for large-scale cases and they are able to handle a large number of transactions or blocks in a reasonable time with an extensive number of users or nodes. Multiple chains can also be formed by multiple nodes using a lottery-based approach, but these must be consolidated before the transaction's completion and affect the transaction's speed recorded in a blockchain. Contrary to a lottery-based approach, voting approaches are faster to complete but at the same time, they get slow when reaching for consensus of the large number of nodes in the network. This occurs because each node has to exchange information with others and causes multiple voting until an agreement is achieved. This forms a trade-off between scalability and completion speed. Several approaches are proposed to enhance the speed and scalability of the blockchain process such as shading, sidechains, utilization of payment channels and parallel processing.

What do all these processes mean in terms of energy?

The initial fuel consumed for processing these financial transactions is electricity[6]. It is estimated that 2.55 GW of electricity was consumed up to quite recently and shortly this figure will rise to 7.67 GW for processing financial transactions which is equivalent to countries like Ireland (3.1 GW) and Austria (8.2 GW). Economic models also announced that we are approaching the latter number. As was explained earlier, in the blockchain, the first solution to timestamping transactions is hashing in which PoW is achieved by a hashing perform with SHA-256 algorithms and the hash starts with a specific number of zero bits. Attempts to find such a hash made every second can be called a hash rate[4]. Once a node achieves a hash that satisfies the required number of zero bits, it transmits the block on to the rest of the network where it was working. Hash rate cannot be calculated directly but it is possible to derive this from the actual time required to mine new blocks for the blockchain. According to a report in mid-March 2018, there were about 26 quintillion hashing operations performed every second by the bitcoin network non-stop. The bitcoin network is processing at 2–3 transactions per second which is almost 200,000 transactions per day, this means hash calculation to process transaction will be 8.7 quintillions to 1 at best (Nair et al., InPress).

Determining the exact value for the energy consumption of a multitude of open, distributed networks is a hard task because the precise number of participants, the properties of their hardware, and the effort which they put into mining are unknown. Fortunately, however, one can obtain good estimates

4 https://www.blockchain.com/charts/hash-rate

for a lower and an upper boundary of the energy consumption of any PoW blockchain[8]. Since both the difficulty of the cryptographic puzzles and the frequency at which solutions are found are easily observable, one can calculate the expected value of the minimum frequency of calculations ("hash-rate") needed to solve the puzzles. This gives a lower boundary of the energy consumption of an arbitrary PoW blockchain:

$$total\ power\ consumption \geq total\ hash\ rate \times min\ energy\ per\ hash \quad (1)$$

The formula (2) estimate indicates the lower boundary, reflecting the likelihood that more solutions are found than disseminated, that further computations in addition to mining are being carried out, and that not every miner has the most energy-efficient hardware[8]. Mining hardware is in general blockchain-dependent because the algorithms used for hashing can differ. For example, Bitcoin uses SHA256, for which very efficient application-specific integrated circuits (ASICS) exist, i.e., chips that are highly optimized for computing hash values and, thus, for solving the puzzles. On the other hand, Ethereum was designed to prevent the use of highly specific mining hardware, so general-purpose graphic processing units (GPUs) can be used for mining[8].

One can also determine an upper boundary for the energy requirement of the mining process for a PoW blockchain, assuming honest and rational miners whose utility from mining is solely financial profit: Participation in the mining process is only profitable as long as the expected revenue from mining is higher than the associated costs[8]:

$$mining\ rewards + transaction\ fees = tot.\ mining\ revenue$$

$$\geq tot.\ mining\ costs$$

$$\geq tot.\ energy\ consumption \times min.electricity\ price.$$

The total power consumption can be calculated by formula (2):

$$\text{total power consumption} = \frac{\text{block reward} \times \text{coin price} + \text{transaction fees}}{\text{avg.blocktime} \times \text{min.electricity price}} \quad (2)$$

The *block reward* (i.e., the number of cryptocurrency coins one receives for solving a puzzle), the *price of a coin*, and current *transaction fees* are, publicly known for every PoW cryptocurrency, the only sensitive number which has to be estimated is the *minimum electricity price*.

The use of formula (1) with data from collectors[5] returns an amount of approximately 125 TWh per year for the energy consumption of Bitcoin, using data from Coinmarketcap for 2020-02-05[8]. To validate a single block in today's cryptocurrencies, every node must typically download up to a few Megabytes of data and perform as many as several thousand hash computations, as well as a comparable number of corresponding computations and database operations. For example, in a 1 MB block used in Bitcoin, there can only be a maximum of around 2000 transactions. These are the leaves of the Merkle tree and, therefore, give a total of 4000 hash value computations and a similar number of corresponding database manipulations and signature checks. By comparison, finding a single block currently involves around 1023 hash computations to solve a puzzle in Bitcoin, around 1020 hash computations for Bitcoin Cash and Bitcoin SV, and around 1015 hash computations for Ethereum and Litecoin[8].

It is important to emphasize that further increasing the energy efficiency of mining hardware would not reduce a PoW blockchain's energy requirements in the long term: To keep the average time for solving a puzzle constant, and, hence, to ensure the security and constant functionality of the network, the difficulty of the cryptographic puzzles is periodically adapted to the total computing power of the network[8].

5 https://www.blockchain.com/charts/hash-rate

In contrast, in the PoS consensus mechanism the weight of a participant's vote is not tied to the scarce resource of computing power, but to the scarce resource of capital. More precisely, there is a random mechanism (there are no truly random number generators for classical computers, but, as a first approximation, this heuristics provides a good indication. The pseudo-randomness typically comes from a subset of the previous blocks) that determines who is allowed to build and attach the next block. The advantage of PoS is that it does not involve any computationally intensive steps such as solving the cryptographic puzzles in PoW. The computational complexity of PoS consensus is low and, typically, insensitive to network size. It is, therefore, very energy-efficient for large-scale systems[8].

On the other hand, the more secure these PoA consensus mechanisms are, the greater their complexity and, therefore, the greater their energy consumption. For example, practical Byzantine fault tolerance (PBFT) consensus overhead scales at least quadratically with respect to the number of nodes in the network and is hence, by contrast to PoW and PoS, highly sensitive on the network size. This, in turn, correlates with the energy consumption associated with consensus[8].

Finally, the PoEA consensus mechanisms are more energy efficient, since they intend to establish trusted random number generators through secure hardware modules. As for PoS and PoA, these further concepts typically do not involve a cryptographic puzzle, except for some concepts which try to establish some kind of useful proof of work (PoW) which solves puzzles that are in some way meaningful for business or science[8].

As was explained above, an important factor for PoW energy efficiency are the mining machines. To measure the electricity consumed by the blockchain mining machines which perform hash calculation is a very big challenge. Although we can calculate the total computational power that is not enough to calculate the power usage required by the underlying machines. An amount of 14 tera hashes per second of hash rate is generated by single Autominer S9 which runs on 1372 W, which is almost more than PlayStation-3 devices running on 40 MW. It is next to impossible to calculate the exact number of connected devices, bitcoin networks have more than 10,000 connected nodes and these single nodes may also consist of multiple machines. As Table 3 shows the electricity consumption required by some of the machines in the bitcoin application generate the energy efficiency of these devices (Nair et al., InPress).

Cooling requirements for blockchain is another aspect that requires energy. According to a study by Hileman and Rauchs that took place in 2017 with 48 miners[6], 11 of these devices were developed for large mining operations and these contributed more than half of the global bitcoin hash rate. These machines generate a huge amount of heat, so additional energy demand was generated for cooling. The blockchain process in general requires cooling technology in indoor operations with a lack of power usage effectiveness (PUE).

Storage of data on blockchain also consumes a lot of energy which indicates that those who want to transact directly on a blockchain would have a high energy cost.

Table 3 – Machines based on ASICS miners[6]

Device/Miner	Hashrate (TH/s)	Energy use (W)	Energy efficiency (J/GH)
Antminer S9	14	1372	0.098
AvalonMiner 821	11	1200	0.109
Bitfury B8 Black	55	5600	0.11
Antminer T9	12.5	1576	0.126
Antminer T9+	10.5	1332	0.127
Bitfury B8	47	6400	0.13
AvalonMiner 761	8.8	1320	0.15
AvalonMiner 741	7.3	1150	0.16

Antminer V9	4	1027	0.257
Antminer S7	4.73	1293	0.273

Energy efficiency in blockchain can be performed with alternative calibrations (Nair et al., InPress):

1 Specializing the data centre – Clouds are predominantly used in the blockchain process, recently GPU and field-programmable gate array (FPGA) based clouds have done a significant job when dealing with intensive workloads by improving the power and performance.

2 Resource-efficient mining – This method is also proposed to minimize the energy wastage during the blockchain process. This approach is based on trusted hardware by Intel.

3 Software guard extension (SGX): This assures security as much as is provided by proof of work but also borrows the partial decentralized trust model to be incorporated into SGX to achieve trust as given by proof of work. In this the basic idea implemented is proof of useful work (PoUW), involving miners which provides trustworthy reporting on central processing unit (CPU) cycles.

4 Transfer of proof: Instead of miners battling for block hashing rights, contrary to its network is giving block adding rights to forgers depending on their capacity of holding the blockchain (i.e., in Ethereum). This approach is based on public blockchain which would slash the energy consumption to a great extent.

5 Sawtooth blockchain software: Intel proposed a novel energy saving blockchain system that incorporates the security features into the chipmaker's CPU.

6 Side chains: This method has evolved over Bitcoin and Ethereum networks using proof of authority (PoA) that allows preselected nodes to run a chain, consuming the same energy as that of light bulb or 78 W.

Long term energy efficiency of public blockchains

Although it is very difficult to calculate the overall energy efficiency of the public blockchains, a market dynamics approach may provide us with insights on the long-term overall efficiency of a public PoW blockchain network.

The primary incentive for a miner to join the network is the financial profit. This is a function of the price of the blockchain assets minus the mining cost. The mining cost is primarily related to the efficiency of the mining equipment and the price of the electricity it consumes.

As the price of the blockchain assets are traditionally extremely volatile, when their price drops the miners that own equipment of low efficiency or use electricity with price will no longer have a financial gain from mining. The miners who have invested in mining equipment with high efficiency and/or low-price electricity will be able to operate even when the blockchain asset are valuated at a lower price.

These dynamics create incentives for the miners to invest in highly efficient mining equipment which in turn drive the R&D companies in this area to innovate. It also incentivizes the miners to seek low-cost energy which in many cases leads them to cleaner, renewable energy sources.

What other energy implications come from blockchain?

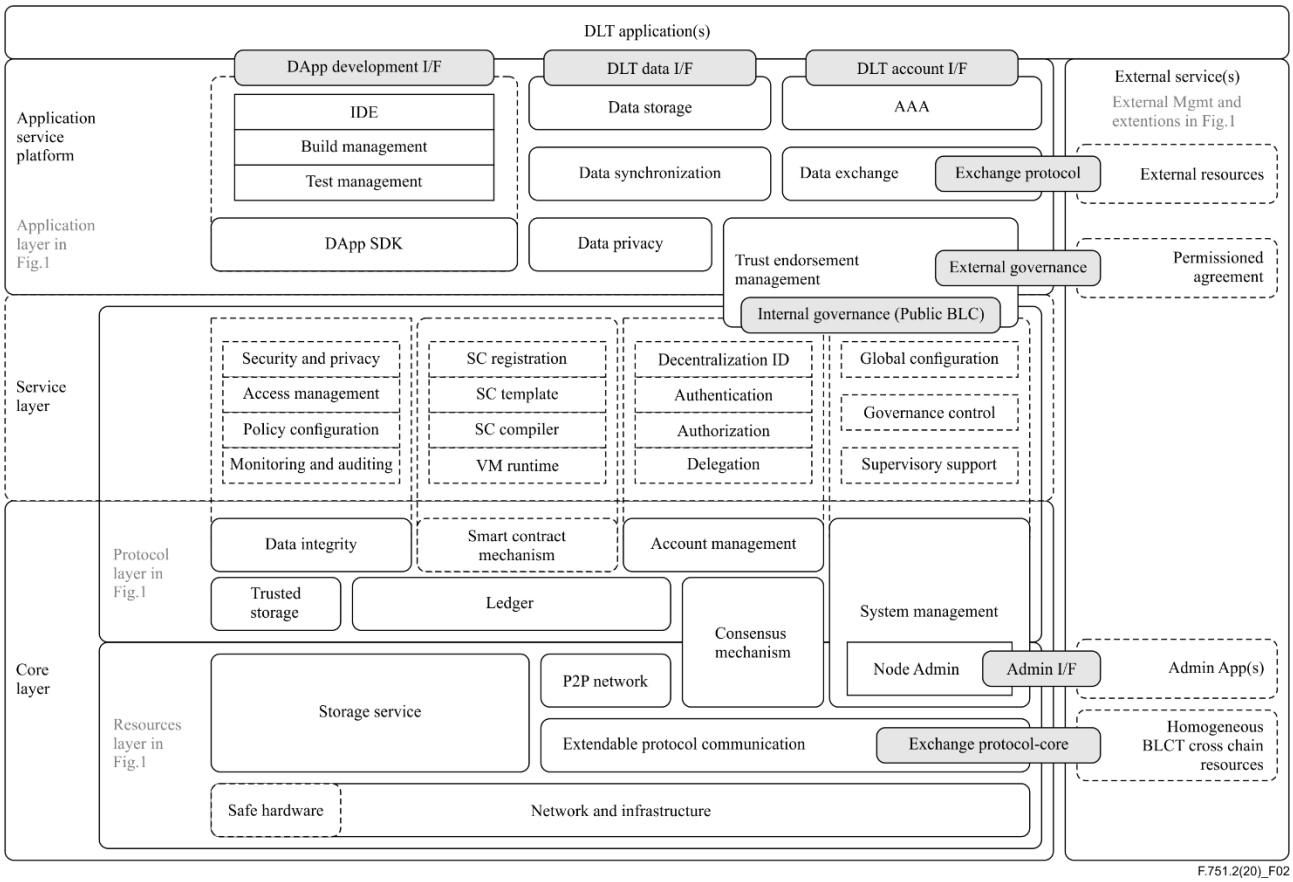

Figure 16 – Schematic diagram of the detailed architecture (TREC-751.2)

This document's focus has been on the core layer of blockchain, see Figure 16. However, the overall architecture generates additional energy demands.

For instance, if a blockchain provides a smart contract mechanism this requires additional energy amounts for each transactions' execution. Without going into too much detail, a smart contract is essentially an application that is deployed on the blockchain. When invoked, each node of the blockchain network executes the code in order to produce/verify a new block. For more details on the rational and the smart contract mechanism, T.REC F.751.1[12] is good starting point.
Depending on the program's complexity a smart contract may have an exponential impact on the energy consumption of the blockchain as the number of nodes increases. Therefore, the design of the blockchains that provide this functionality also provides a control mechanism.
When a smart contract is deployed on the blockchain, the miners are rewarded additionally every time it executes. In Ethereum for example, each operation that a smart contract executes carries a cost which the agent that requested the execution needs to pay. In Ethereum terms, this is called "gas"; the higher the complexity of the contract, the more gas it requires to be executed. Gas needs to be bought using the native cryptocurrency of the Ethereum blockchain, Ether. The higher the Ether price, the more expensive the contract execution. This mechanism provides a way of ensuring that highly complex smart contracts will not be executed on the blockchain as it will be prohibitively expensive. We can however extend formula (1) to take into account this additional source of energy consumption with the following formula (3):

$$\textit{total power consumption} \geq \textit{energy for transaction execution} + \textit{energy for block production} \; (= \textit{total hash rate} \times \textit{min energy per hash}) \quad (3)$$

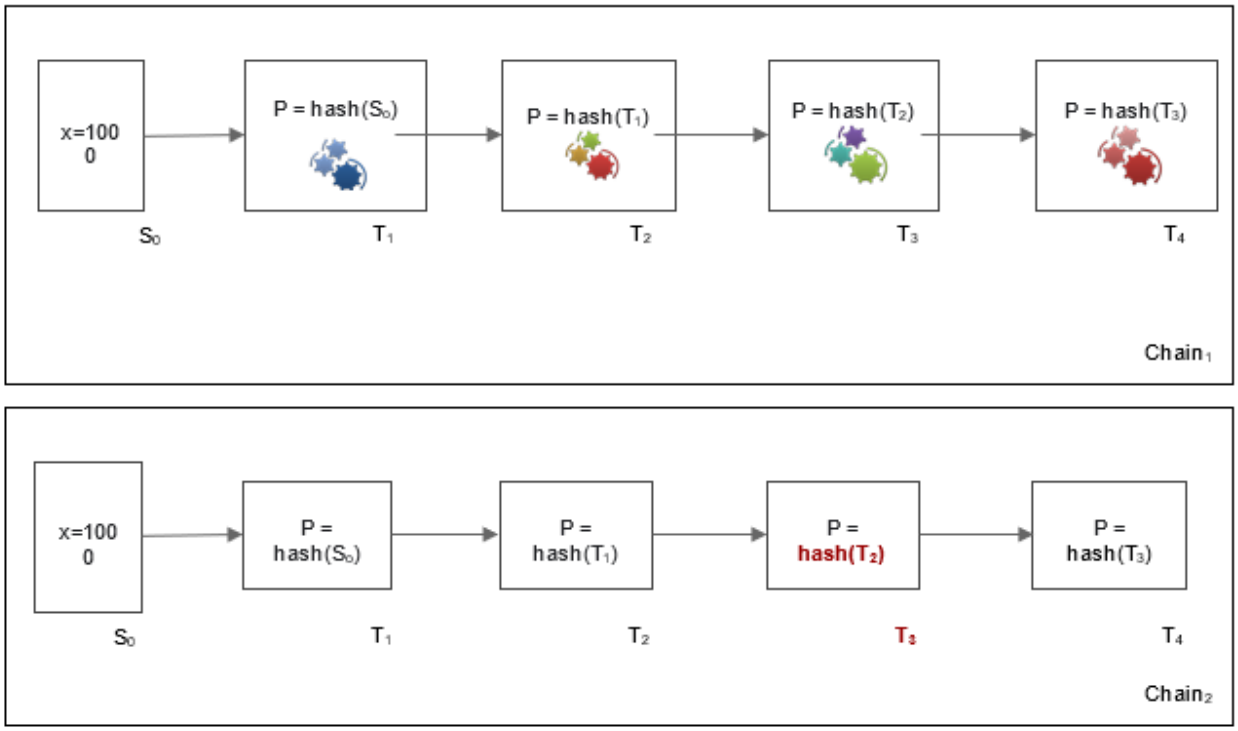

Figure 17 – Schematic diagram of the detailed architecture (TREC-751.2)

Figure 17 demonstrates how the primary chain (shown in Figure 3) changes when a smart contract or other service is installed on the blockchain. Each of the "gear icon set" in the blocks depicts an instance of such an algorithm, which runs and operates during each block's execution. Such an algorithm can also emerge (it is depicted with different colours to indicate different versions of the same algorithm) and in this regard it can be realized that the overall energy demand emerges too.

7 Conclusions

The above findings support the purpose of this document, which is twofold: to identify the energy demand sources and to model this energy demand, in order to calibrate its efficiency. In this regard, this document returned useful findings for a policy maker who has to deal with blockchain implementations. More specifically, the following decisions need to be made:

1) **Choose the level of trust**: as long as trust decreases, the energy demand increases and cost increases too. Literature evidence showed that PoA has minimum energy demand; PoW: has the maximum energy demand; and PoS is in between these choices.

2) **Transaction timeslot**: plays crucial role and is a critical parameter that affects the energy performance of a blockchain, since it controls the computational power for solving a blockchain puzzle (in Bitcoin this timeslot is approximately 10 minutes). It is important to realize that this timeslot definition affects the energy demand of all blockchains.

3) **PoS is a medium choice in terms of energy efficiency**: PoS energy demand is affected by the number of validators that are defined for a network. Kusama is a real case PoS case, with specific computational power demand rules for becoming a validator.

4) **The choice of the devices affects the energy performance**: Table 3 contains representative miners, with their energy performance. This table changes over time and needs to be updated.

Moreover, formulas (1) and (2) explain how energy demand can be calculated in PoW cases and return an estimation, which can be considered and calibrated when needed for a new PoW blockchain deployment and in this regard, it can become a reference for future implementations. Formula (2)

does not estimate the energy demand directly but, it can justify whether a cryptocurrency's value is really worth it in terms of energy consumption and its corresponding environmental impact.

Finally, some future thoughts for this document have to do with testing of these suggestions and with blockchain evolution. Testing can be performed with real case installations, with specific architectures and users, and the measurement of the energy performance. With regard to the future of blockchain, trends such as the new blockchain of blockchains (Polkadot[6]) emerge (Kusama is a testing case), which need to be investigated further with regard to energy demands and efficiency.

---

6 https://polkadot.network/