Politecnico di Torino

Master's Degree in Engineering and Management

Master's Thesis

Trustless contract management: a study on the benefits of blockchain-based smart contracts

Supervisor:
Prof. Danilo Bazzanella

Student:
Antonino Gucciardi

Academic Year 2022/2023

Index

Introduction

Contracts are an essential part of business and legal relationships. They are agreements between two or more persons or parties, that specify mutual rights and obligations and that are enforceable by law. When two businesses want to do business together, a contract outlines the activities undertaken by both organizations and the conditions under which they will carry out their obligations. Contract management is the process of managing contracts creation, execution, and monitoring to make sure that all parties fulfill their obligations. It is a complex activity involving many stakeholders. Effective contract management is essential to maximize operational and financial performance at an organization, while reducing financial risk. By offering a secure, immutable, reliable, and trustworthy platform, blockchain has the potential to revolutionize the way contracts are managed. The decentralized nature of blockchain renders the need for intermediaries unnecessary, reducing delays and costs, and eliminating complexity and trust issues. Smart contracts are a development of the blockchain technology. They are computerized transaction protocols, characterized by immutability and automation, that automatically execute the terms and conditions of a contract. The enforcement of contractual obligations in ensured by unchangeable computer code, allowing the parties to put less trust in one another and only rely on the underlying technology. The aim of this thesis is to explore the potential of smart contracts to improve traditional contract management processes. Chapter 1 introduces the blockchain technology, its main components, the blocks, hash functions, and consensus protocols. Chapter 2 examines the main industry applications of blockchain technology in finance, supply chain management, and the internet of things. Chapter 3 focuses on cryptography, the backbone of blockchain, which makes it a secure and reliable technology. It covers the basics of symmetric and asymmetric key cryptography and digital signatures. Chapter 4 introduces smart contracts and their main features. Special attention is catered towards Ethereum, the most widely used blockchain platform for smart contracts. Chapter 5 provides an in-depth analysis of contract management, covering the definition of contracts, contract organization, and contract administration. Finally, chapter 6 analyzes the impact of smart contracts and blockchain in the field of project and contract management. Their several benefits are listed and thoroughly discussed, while not forgetting the current challenges associated with their adoption. The recognition of smart contracts by the legal community is examined. The final sections of the thesis aim at investigating how blockchain-based frameworks can significantly improve contract management, enforcing contractual terms and conditions, with a special focus on payment automation, surety bonds contracting and dispute resolution. Overall, the thesis provides a comprehensive and in-depth analysis of the potential of smart contracts and blockchain technology in contract management.

1 Blockchain

Blockchain is a digital, decentralized and distributed system (Bashir, 2017), which allows the process of recording transactions and tracking assets in a network. Assets can be tangible (a house, car, cash, land) or intangible (intellectual property, patents, copyrights, branding). On a blockchain network, practically anything of value may be recorded, lowering risks and costs, and increasing efficiency for all parties involved (IBM, 2022). Being a digital ledger, the concept behind blockchain is not at all new. In fact, ledgers have been part of commercial processes since ancient times. New instead is the supporting technology, since it is digital, decentralized, distributed and immutable. A blockchain is actually a database because it is a digital ledger that stores information in data structures called blocks. A database likewise stores information in data structures called tables. However, while a blockchain is a database, a database is not a blockchain (Tabora, 2018). Ledgers may in fact be considered as a subset of databases. In the latter, it is not only possible to add new data, but also to modify and even cancel old ones. In the former, instead, it is only possible to add new data. Typically, the management of a database is entrusted to a known and reliable set of users, who have the power to handle the data stored inside of it at their own discretion. Blockchains are instead open systems whose data everyone can access and is able to consult. Anyone can study the code, develop their own ideas and propose improvements. Since blockchains do not need any form of access control, they are especially valuable in scenarios in which safety and trust are of the utmost importance. The two systems are not mutually exclusive. It is possible to define best use cases for databases and blockchain. For systems that deal with high volume traffic, like retail, a database is still the best solution. Data that does not need to be encrypted or hashed, like the number of items sold by a store at the end of the day, are best recorded on a database. Using a blockchain for something as simple as bookkeeping is expensive and databases result to be more efficient. Other examples for which databases are better suited are data that need continuous updating, like monitoring and sensors; confidential information (non-transparent to the public); financial data from markets that require fast processing; data that does not require verification; standalone applications that store data and relational data. On the other hand, blockchains are ideal for monetary transactions, transfer of value, verification of trusted data (identity, reputation, credibility, integrity, etc.), public key verification, decentralized applications (DApps) and voting systems (Tabora, 2018).

There are several features of blockchain technology that make it a revolutionary and disruptive innovation:

- Decentralization: a network of computers distributed around the world, rather than a central authority, is relied upon to validate and record transactions. This makes it more resistant to censorship and tampering.
- Immutability: Once data have been recorded on the blockchain, they cannot be modified or deleted. This makes it a secure way to store data.
- Transparency: all transactions are visible to anyone with access to the network. This promotes accountability and trust.

Despite these revolutionary features, there is a fundamental challenge that must be addressed in the design of blockchain networks, the scalability trilemma. The scalability trilemma states that blockchains must choose to optimize two of three properties: scalability, security and decentralization.

Security, which derives from the immutability feature, and decentralization have already been defined, scalability refers to the ability of a blockchain network to handle a large number of transactions and is usually measured in transactions per second (TPS). The scalability trilemma highlights the fact that it is unfeasible to achieve all three properties simultaneously in a blockchain network. In practice, a blockchain project must prioritize these properties differently based on its use cases and goals. There have been many attempts to address the scalability trilemma in blockchain technology, including the use of sharding, off-chain transactions, modifying blockchains structure (IOTA), increasing block size limit, and consensus algorithms such as Proof-of-Stake. However, these solutions also come with their own trade-offs and challenges. As an example, the Proof of Stake (PoS) consensus algorithm improves the issue of scalability in the Proof of Work algorithm, but carries the "Centralization Risk", as the concentration of wealth in a small number of validators can lead to centralization, with a small group of individuals getting too much control over the network.

The main components of the structure of a blockchain are:

- Blocks: as functional units of a blockchain, blocks are registries in which data are stored, which may be transactions or entire digital applications. Taking Bitcoin as an example, every bitcoin transfer from a wallet to another gets permanently written over the blocks belonging to the Bitcoin blockchain. It is possible to add new blocks, while it is not allowed to remove or edit previously existing blocks.
- Chain: blocks are connected one after the other thanks to hash functions. Each block hash derives partially from the previous block’s hash. This property makes sure that neither the succession

of the blocks nor the data inside can be altered, giving the characteristics of safety and immutability to the whole blockchain. In fact, if a malicious user wanted to modify the information belonging to a block, it would need to modify those belonging to all following blocks. The further from the end of the chain the more difficult this becomes to pursue.

- Nodes: blockchains are ledgers distributed over nodes belonging to a network. Nodes are essentially computers which store the same copy of the ledger in geographically different places Their presence gets rid of the necessity of a controlling central institution and makes the blockchain a decentralized system. Of course, in order to make a blockchain truly decentralized, it is essential that the quantity and distribution of nodes are such to hinder a coordinated group to gain control of the entire network. They have two main functions: to download and locally store a full copy of the blockchain and to control and validate blocks and transactions held within. There are two main types of nodes: full nodes and SPV (Simplified Payment Verification) or light nodes. Full nodes download and locally store a complete and synchronized copy of the blockchain, which in the case of Bitcoin implies over 430 gigabytes (Ycharts, 2022), and must validate all transactions and blocks. The local copy of the blockchain is constantly updated as new blocks are found and used to extend the chain. They are core clients performing the wallet, miner, full blockchain storage, and network routing functions (Bashir, 2017). Light nodes keep in memory only some data from each block, instead of the entire blockchain, retrieved from full nodes, without which they would not be able to perform. They are unable to validate fresh blocks, but they can superficially validate transactions using the SPV approach (Simplified Payment Verification). SPV nodes use an authentication path, or merkle path, to verify that a specific transaction is included in a block without having to download all of the transactions in the block. Merkle trees are a data structure that is used in blockchain technology to summarize and verify large sets of data. This allows SPV nodes to operate more efficiently, as they only need to download the block headers and a small amount of additional data in order to verify transactions. Unfortunately, the storage and bandwidth requirements of SPV clients still increase linearly with the chain length (Bunz, Kiffer, Luu, & Zamani, 2020). There is a third type of nodes, called miners, who actively participate in the distributed consensus mechanism, by means of a resource-intensive process called mining. Mining is a resource-intensive process by which new blocks are added to the blockchain. It is essential to maintain high security levels and decentralization in the network. The process's high resource requirement is justified by the fact that both the legitimacy of the transactions and the legitimacy of the blocks, specifically that the sequence of the same remains unchanged, that is, that no one has altered the historical data, are equally vital to ensure. A node that wants to tamper with data within blocks and validate malicious transactions needs to have control over at least 51% of the overall network computational power for at least the time necessary to produce a new block (ten minutes in Bitcoin's case). It's practically impossible that an attacker manages to gather and control such computational power in the case of Bitcoin's network. Furthermore, even if it was feasible, it would probably be useless, since Bitcoin's price would be severely affected by the attack, making any double-spending

> attempt worthless. Miners are usually remunerated for their work and efforts by means of newly created cryptocurrencies and are paid transaction fees in return of including transactions in their blocks. Every 210,000 blocks, or roughly every four years (210,000 × 10 min ≃ 4 years), the rate of new bitcoin generation declines by 50%. When bitcoin was initially introduced, the block reward was 50 bitcoins; then in 2012, this was reduced to 25 bitcoins. In July 2016, this was further reduced to 12.5 coins (12 coins) and in 2020 it reached 6,25 BTC. Bitcoin's maximum circulating supply is set at 21 million bitcoins and will be reached in 2140, when the block's reward will be 1 satoshi (10^-8 BTC, minimum bitcoin unit). No new bitcoins can be created after that. Bitcoin miners, however, will still be able to profit from the ecosystem by charging transaction fees.

Scott Stornetta and Stuart Haber can be regarded as the Blockchain's founders because they came up with the concept of a time-linked chain as a solution to the problem of authenticating documents. They were very concerned about ensuring the integrity of digital records. Because retaining records was so important to society and because it was so simple to change digital information covertly, they felt the need to create a reliable infrastructure that could guarantee the accuracy of the records that were kept in. They proposed computationally practical procedures for digital time-stamping of text, audio, picture, and video documents so that it was infeasible for a user either to back-date or to forward-date his document, even with the collusion of a time-stamping service (Haber & Stornetta, 1991). Their solution involved using one-way hash functions, taking requests for document registration (i.e., the document's hash value), grouping the documents into "units" (blocks), building the Merkle tree, and producing a linked chain of hash values. Unfortunately, the solution they had conceived didn’t appeal to the market in the early ‘90s, not because it was outcompeted by any other digital integrity mechanism but because businesses thought that they were doing just fine without any digital integrity mechanism at all.

Satoshi Nakamoto is the pseudonym used by the individual or group of individuals who created Bitcoin, the first recorded application of the blockchain, and authored its original white paper "Bitcoin: A Peer-to-Peer Electronic Cash System" (Nakamoto, 2008). The true identity of Satoshi Nakamoto has never been revealed and remains one of the biggest mysteries in the tech world. In the paper, he described a new electronic cash system, “a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions”, that used a peer-to-peer network to prevent double-spending, a problem that had plagued previous digital currency attempts. To achieve this, he proposed to use a decentralized and distributed ledger able to record all Bitcoin transactions. The blockchain is maintained by a network of computers, called nodes, that work together to validate and record transactions. This system ensures that no single entity has control over the network. Satoshi Nakamoto stresses that the system is tamper-proof “as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes”. One of the main problems with traditional electronic cash systems is the need for a trusted third party, such as a bank, to facilitate transactions. This trust-based model is vulnerable to fraud and can be slow and expensive, as it requires intermediaries to verify and process transactions. In contrast, the Bitcoin system uses a decentralized and distributed ledger, based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Nakamoto also introduced the concept of mining, which is the process of verifying and adding transactions to the blockchain. Miners use powerful computers to solve complex mathematical problems in order to validate transactions and create new blocks. In return, they are rewarded with a certain amount of bitcoins. Satoshi Nakamoto’s invention is also a practical and novel solution to a problem in distributed computing, known as the “Byzantine Generals’ Problem”, that illustrates the challenges of achieving consensus in a distributed system where some of the participants may be faulty or malicious. The Byzantine Generals’ Problem can be illustrated as follows. A group of generals of the Byzantine army are encamped around an enemy city, each commanding their own troops. Geographically these generals are separated and they have to communicate with each other through messengers in order to decide whether to attack the enemy or not. This situation is also complicated by the presence of traitors among the generals. These traitors try to confuse loyal generals, sending them a false information about decisions of other generals. General A plans to attack, but General B, a traitor, sends a message to General C claiming that General A intends to retreat and that he also plans to retreat. Meanwhile, General B sends a different message to General A, falsely claiming that he intends to attack. This deceptive behaviour makes it difficult for loyal generals to make a joint decision, resulting in a weak attack and a likely defeat. The loyal generals must somehow reach a consensus on a course of action despite the possibility of traitorous generals attempting to undermine them. In a centralized system, this problem would not arise, since a central authority would make the decision about whether to attack or retreat, which all the generals would then follow (Lamport, Shostak , & Pease, 1982). In a decentralized framework, it is necessary to devise a consensus method able to take into account the presence of malicious nodes making sure that these are unable to influence the correct functioning of the system. The proof-of-work algorithm conceived by Satoshi Nakamoto makes it possible to overcome the Byzantine generals' problem. In this system, nodes must perform a certain amount of computational work to solve a complex mathematical problem. This ensures that it is difficult for a single node or group of nodes to manipulate the blockchain, because doing so would require a significant amount of computational power: at least 51% of the total computational power of the entire network for the amount of time needed to create a new block. It becomes even more difficult to achieve this feat as the total computational power of the entire network grows, that is as the number of miners increases and, thus, the more decentralized the system gets. The combination of the blockchain and mining makes Bitcoin a decentralized, secure, and transparent system for conducting electronic transactions without the need for intermediaries, such as banks. Since its inception, Bitcoin has grown in popularity and has spawned a whole new industry, known as cryptocurrency. It has also inspired the development of other blockchain-based systems and applications. However, it has also faced its share of controversy and criticism, with some arguing that it is used for illegal activities, such as money laundering and drug trafficking. The large and diverse group of existing blockchains can be divided and classified based on the level of access and control that users have over the network. The main criteria for classification are user access, data transparency and traceability, ability to add new blocks, and the management of the stability and integrity of the blockchain. It is possible to distinguish two main types of blockchains: permissionless or public and permissioned or private. In permissionless or public blockchains everyone can join and participate in the network, execute transactions and validate new blocks without the need for permission. These blockchains are decentralized: there is no central authority controlling access or decision-making over the entire network, which is instead governed by a set of rules and protocols that are agreed upon by all participants. Permissionless blockchains are often transparent, meaning that all transactions and interactions on the network can be seen by anyone. This can increase trust and accountability within the network. One final advantage is their immutability; in fact, once validated and added, transactions cannot be altered and tampered with. Permissionless blockchains are often used for applications that require transparency, such as cryptocurrencies. Their disadvantages mostly lie in scalability and security. Due to the need of consensus among all participants, the process of validating transactions can be slow and less efficient. Security threats, like 51% attacks, can undermine the integrity of the network. In permissioned or private blockchains, a central authority controls and restricts the access to the network. The consensus process can only be achieved by a limited and predefined number of participants. Write access and reading permissions are also controlled by a preselected set of nodes (Guegan, 2017). Permissioned blockchains can be faster and more efficient than permissionless blockchains, as they do not require consensus among all participants. This can make them more scalable and suitable for certain types of applications. They offer more privacy as access to data is restricted to a trusted number of nodes. On the other hand, centralization, limited accessibility and lack of transparency are their main disadvantages. Permissioned blockchains are often used in business and enterprise settings, where there is a need for a secure and controlled environment. Some famous examples are Corda and Hyperledger. There are also hybrid blockchains, which combine elements of both permissioned and permissionless blockchains. These blockchains may have some level of control and access restrictions, but also allow for greater participation and decentralization than purely permissioned blockchains. The are designed to strike a balance between the decentralized nature of public blockchains and the centralized control of private blockchains. Consortium blockchains represent an example. Unlike private blockchains in which the owner has full authority, consortium blockchains preselect a group of nodes or participants, who are given certain privileges over validating the transactions, creating new blocks, and making decisions about the network's governance and management (Sankar, Sindhu, & Sethumadhavan, 2017).

1.1 Blocks in a blockchain

Blockchains are structured as a series of blocks. A block in a blockchain is a container data structure that aggregates a list of transactions that have been recently completed on the network. Blocks are added to the blockchain sequentially starting with the first block, which is called the "genesis block". It is the common ancestor of all the blocks in the blockchain, meaning that starting at any block and following the chain backward in time, the genesis block will eventually be reached. Blocks differ in size and type of information stored depending on which blockchain system they are part of. A common way to represent the structure of the blockchain is to imagine it as a vertical stack of blocks, with the first block serving as the base and subsequent blocks layered on top. Because of this visual representation, the distance of a block from the first block is referred to as its "height," and the most recently added block is often referred to as the "top" or "tip" of the stack. Each block also contains a unique code, called a "hash," that links it to the previous block in the chain, generated using the SHA256 cryptographic hash algorithm on the header of the block. In the header of each block, the "previous block hash" field references the block that came before it, known as the parent block. Essentially, this means that the header of each block includes the hash of its parent block. This creates a chronological record of all the transactions that have taken place on the network. The “previous block hash” field is inside the block header and thereby affects the current block’s hash. When any changes are made to the parent block, the hash of the parent block will also change. This change in the parent's hash will then require a change to the "previous block hash" pointer in the header of the child block. This, in turn, will cause the child block's hash to change, which will require a change in the pointer of the grandchild block, and so on. This cascading effect ensures that once a block has many descendants, it cannot be altered without requiring the recalculation of all subsequent blocks. This process would require a large amount of computation and energy, making it impractical to change a block with a long chain of descendants. As a result, the deep history of the blockchain, represented by the long chain of blocks, becomes immutable, providing a key security feature (Antonopoulos, 2014).

1.1.1 Structure of a block

The block is made of a header, containing metadata, followed by a long list of transactions that make up the bulk of its size. The size of a block on a blockchain can differ, depending on the blockchain in question. Generally, block sizes have a fixed or maximum size to stop the overuse of resources and ensure the blockchain can expand efficiently. As an example, the block size for the Bitcoin blockchain is limited to 1 megabyte (MB). This implies that each block on the Bitcoin blockchain can contain a maximum of 1 MB of data. The average block size for the Bitcoin blockchain is usually around 0.5 MB, though this can change depending on the network activity. Other blockchains may have different block size limits. For example, the Ethereum blockchain has a block size limit of approximately 12 MB, while the Litecoin blockchain has a block size limit of approximately 4 MB. The average block size of a blockchain is determined by a combination of factors, such as the blockchain's specific design, the amount of data recorded, and the amount of network activity. Overall, block size limits play an important role in maintaining the security of a blockchain network. They also have implications for scalability, as increasing block size limits can improve a blockchain network's throughput and transaction speed. The block header is a small piece of data that is included at the beginning of every block in a blockchain. It is typically 80 bytes in size and contains three sets of metadata that are used to identify and validate the block. The first set of metadata in the block header is a reference to a previous block hash, which connects this block to the previous block in the blockchain. The second set of metadata, namely the difficulty, timestamp, and nonce, relate to the mining competition. The difficulty refers to the level of computational effort required to solve the cryptographic puzzle, the timestamp indicates the approximate creation time of the block and the nonce is used to modify the input to the puzzle in order to generate a unique solution. The third element of metadata in the block header is the merkle tree root, which is a data structure employed to summarize all of the transactions housed in the block. The merkle tree root is computed by hashing jointly pairs of transactions and then hashing the ensuing hashes until a single hash, termed the merkle root, is obtained. This allows the transactions in the block to be validated rapidly and expeditiously without the need to process every single transaction.

Table 1 - The structure of the block header (Antonopoulos, 2014)

Size	**Field**	**Description**
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block
4 bytes	Difficulty Target	The Proof-of-Work algorithm difficulty target for this block
4 bytes	Nonce	A counter used for the Proof-of-Work algorithm

When a node receives a new block from the network, it will validate the block and link it to the existing blockchain by examining the block header and looking for the "previous block hash" field. If the node recognizes the hash as one it already has in its local copy of the blockchain, it will add the new block to the end of the chain, extending the blockchain. A block in a blockchain can be identified in two ways: by its block hash and by its position in the blockchain (also known as its block height). The block hash is a 32-byte cryptographic hash that is calculated by running the block header through the SHA256 algorithm twice. It serves as a unique identifier for the block and can be independently calculated by any node by hashing the block header. The following identifier hash belongs to the genesis block of the Bitcoin's blockchain:

000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

The block hash is not actually included within the block's data structure, but is instead calculated by each node as the block is received from the network. The block height, on the other hand, refers to the position of the block in the blockchain and is not a unique identifier. It may not always identify a single block, as two or more blocks may have the same block height and compete for the same position in the blockchain. The first block ever created is at block height 0 (zero). Each subsequent block added "on top" of that first block is one position higher in the blockchain, like boxes stacked one on top of the other. The block height is not stored within the block and is instead dynamically calculated by each node when the block is received from the network. Both the block hash and block height might be stored in a separate database table as part of the block's metadata, to facilitate indexing and faster retrieval of blocks from disk (Antonopoulos, 2014).

1.1.2 Hash functions

The hash function is a mathematical algorithm that is used to create a unique code, or hash, for each block. Hash functions take in an input, or "message," and produce an output, or "digest," that is a fixed-length and unique to the input string made up of letters and numbers. They are present in numerous protocols, such as digital signatures, integrity verification, message authentication and password protection. It is common, for instance, to store the hash of the users' passwords in internal databases instead of the password itself, so it can verify the authenticity of the user and protect the system in case an attacker gets access to the databases. In the context of blockchain, the input for the hash function is the data contained in the block, including the list of transactions and the hash of the previous block. The output, or hash, is a unique code that is generated based on the data in the block. Because of this character, hash function can provide secure properties like integrality, non-repudiation and so on. It is one of the key techniques to protect personal information, data security and system security. The hash function plays a crucial role in the security and integrity of the blockchain. It ensures that each block is unique and cannot be altered once it has been added to the chain. This is because any change to the data in a block would result in a different hash, which would not match the hash of the previous block and would be rejected by the network. The most representative type of hash function is MD serial, including MD4, MD5, SHA0, SHA1, SHA2, RIPEMD, HAVAL and so on. This kind of hash functions are designed based on MD4 and bring their own improvement to enhance security (Yang, Chen, Zhang, Yu, & Zhang, 2017). MD5 and SHA1 were the two standard hash functions until X. Wang proposed the differential attack in 2004 (Wang, Feng, & Lai, 2004).

The following are the main and most important characteristics of hash functions:

- Equal inputs provide the same outputs;
- Collision resistance property: it is computationally infeasible to find any two distinct inputs x, x’ which hash to the same output, i.e., such that h(x) = h(x’) (Rogaway & Shrimpton, 2004).
- Avalanche Effect property: even a tiny little change of input will cause the tremendous change of output. In order to guarantee that hash function has certain randomness, and make sure the attacker cannot infer input through hash value; hash function should keep

the property of Avalanche Effect (Yang, Chen, Zhang, Yu, & Zhang, 2017).

- Hash functions are designed to be one-way functions, which means that it is computationally infeasible to determine the input data from the hash value. This property is known as "hiding" or "first preimage resistance." In other words, the hiding feature of hash functions refers to their ability to obscure the original input data in such a way that it is difficult or impossible to recover the original data from the hash value. This makes hash functions useful for a variety of applications, such as verifying the integrity of data, generating unique identifiers, and storing passwords in a secure manner.

The hash of each block is generated from the data contained within it and the hash of the previous block, which is why the sequence of the blocks is guaranteed by the cryptographic hash function. In fact, the hash of a given block would not be the same if it were preceded by different types of blocks. Moreover, noticing even a small change to the contents of any block would be very easy as it would induce a clear variation in both the hash of the block and all subsequent hashes.

1.2 Consensus protocols and mining

All network nodes participating in a blockchain agree with a single logical state at every moment. All traditional systems use a central authority to verify and clear all transactions, maintain a master copy of data and ensure that all participants agree on its contents. This central authority is responsible for resolving any conflicts that may arise and ensuring that all participants are working with the same information. The blockchain is not created by a central authority but is assembled independently by every node in the network. Somehow, every node in the network, acting on information transmitted across insecure network connections, can reach the same conclusion and assemble the same copy of the public. In a decentralized system, such as a blockchain, all participants have a copy of the data and any changes to the data must be agreed upon by a majority of participants. This makes it much more difficult for any participant to manipulate the data, as they would need to control a majority of the network in order to do so. Satoshi Nakamoto's main invention is the decentralized mechanism for emergent consensus. It is called emergent because it emerges as a result of the asynchronous interaction of thousands of independent nodes, all following simple rules. Decentralized consensus emerges from the interplay of four processes that occur independently on nodes across the network (Antonopoulos, 2014):

- Independent verification of each transaction, by every full node;
- Independent aggregation of those transactions into new blocks by mining nodes which get then added to the chain;
- Independent verification of the new blocks by every node and assembly into a chain;
- Independent selection, by every node, of the chain with the most cumulative computation.

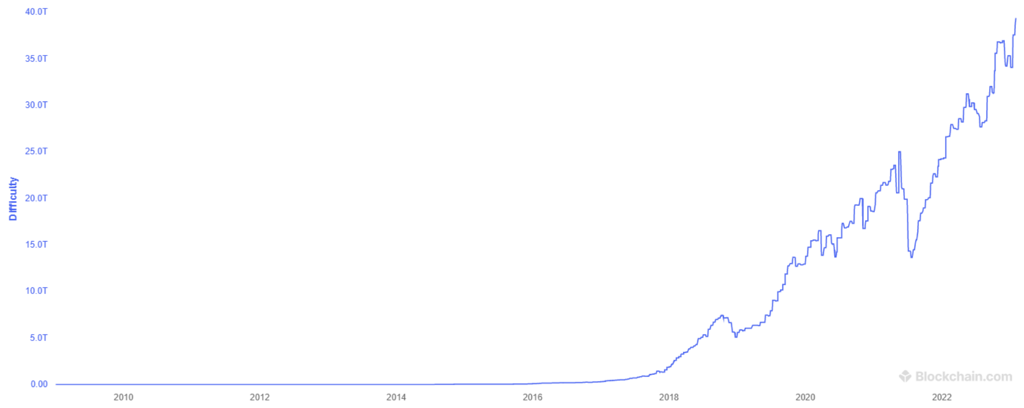
These processes work together to ensure the integrity and security of the network without the need for a central authority or trusted third party. After adding a new block, miners in a blockchain are rewarded for performing the computational work. The block reward consists of two parts: new coins created with each new block, and transaction fees from all the transactions included in the block. The process is called mining because the reward (new coin generation) is designed to simulate diminishing returns, just like mining for precious metals. The reward for mining a block is halved every certain number of blocks. This is called halving; for Bitcoin it occurs every 210,000 blocks and it takes approximately four years. The halving process will go on until the year 2140, when all bitcoin (20.99999998 million) will have been issued. After that date, no new bitcoin will be generated. This decrease in reward is built-in to control the rate at which new coins are created and ultimately to control the total supply of the cryptocurrency. Currently, the second item of the reward, the transaction fees, represents a very small portion of the miner's income, the vast majority coming from the newly minted bitcoin. Transaction fees are paid from users who want to have their transactions included in the next block mined. However, the contribution of transaction fees to the miner's reward will constantly increase due to the halving phenomenon and to the increase of the number of transactions per block. Gradually, the mining reward will be dominated by transaction fees, which will form the primary incentive for miners. After 2140, the amount of new bitcoin in each block drops to zero and bitcoin mining will be incentivized only by transaction fees. Proof of Work and Proof of Stake are the two most notorious consensus mechanisms.

1.2.1 Proof of Work (PoW)

The Proof of Work (PoW) is the consensus protocol currently in use for Bitcoin. Full nodes verify all transactions sent to the network by users. A set of verified transactions are grouped together into a candidate block by miners, which have to solve a complex cryptographic problem requiring huge computational power in order to find a valid block hash. More accurately, they need to perform the double hashing, using the hash function SHA-256, of the header of the block, that consists in the block data, that, in its turn, contains the hash of the previous block, the root of the Merkle tree referred to the block, the timestamp and also the nonce.

Such final hash should be lower than a target number T, the difficulty, set in the network. The cryptographic problem is designed such that its difficulty can be adjusted by the network to control the rate at which blocks are added to the chain, ensuring that the network remains secure and efficient. The difficulty increases as the computational power of the entire network increases. The Bitcoin network has a global block difficulty, which is adjusted periodically as a function of how much hashing power has been deployed by the network of miners to ensure that it takes 10 minutes on average to add a new block to the Bitcoin blockchain (O'Dwyer & Malone, 2014).

Figure 1 - The difficulty trend in Bitcoin network since the beginning (Blockchain.com, 2023)



The difficulty determines how hard it is for a miner to find a valid block hash that meets certain criteria, such as having a certain number of leading zeros. The miners' goal is to find a nonce that, when combined with the other data in the block header, results in a block hash that is numerically less than the target value set by the difficulty. A higher target means it is less difficult to find a hash that is below the target. A lower target means it is more difficult to find a hash below the target. The target T and difficulty D are inversely related:

$$D = \frac{T_{max}}{T}$$

Where the largest possible value of the target T_{max} is $(2^{16} - 1)2^{208} \approx 2^{224}$.

Every time that the computed hash is not less than the target, the miner will modify the nonce (usually just incrementing it by one) and try again. It is possible to estimate the amount of work that it takes to succeed from the difficulty imposed by the target. When the algorithm is a based on a deterministic function such as SHA256, the input itself constitutes proof that a certain amount of work was done to find the correct nonce to produce a result below the target, hence the name Proof-of-Work. Once a miner finds a valid block hash, the information is then transmitted to the other nodes in the network, which then verify the block to ensure that the block hash is indeed below the target value. This ensures that only valid blocks are propagated on the network. As the block ripples out across the network, each node adds it to its own copy of the blockchain. The process of adding a block to the chain is referred to as consensus. As mining nodes receive and validate the block, they abandon their efforts to find a block at the same height and immediately start computing the next block in the chain, using the previous block as the "parent". The miner who added the valid block to the chain gets the reward in cryptocurrency in terms of newly issued coins and transaction fees. Miners who act dishonestly have their blocks rejected, waste the effort expended to find a Proof-of-Work solution, thus incurring the cost of electricity without any reward. The entire outlined process ensures that the state of the blockchain is secure and tamper-proof. It makes it very difficult for any miner or group of miners to manipulate the state of the blockchain, as they would need to control the majority of the network's computational power. In fact, in order to change a block in the blockchain, a miner must not only redo the work to find the right hash for that block but also for all subsequent blocks in the chain. In theory a group of miners controlling 51% of the network's computational power would have the ability to manipulate the blockchain by adding blocks faster than the rest of the network. However, this is very difficult to happen in practice both because for networks with a very high total hash rate, like Bitcoin, it's unfeasible to amass or control the hardware needed to generate similar power, and because the network is constantly growing and adding new miners. Additionally, even if a group of miners managed to control 51% of the network's computational power, the PoW consensus protocol ensures that the network will always follow the longest chain. So, the group of malicious miners should be able to manipulate a block and recalculate the solution for all subsequent blocks before the rest of the network manages to find the solution for the latest block. The Proof of Work (PoW) consensus protocol was originally proposed by Dwork and Naor as a solution to control spam mail (Dwork & Naor, 1993). The idea behind it was that, in order to send an email, the sender would have to solve a cryptographic problem, which would take only a few seconds for a single email but would prevent spammers from sending large numbers of emails quickly. The concept of using hashing to solve a cryptographic problem by varying a nonce to reach a target was first introduced years later by Back in Hashcash (Back, 2002) in the same application studied by Dwork and Naor. However, it was only in Bitcoin that the PoW was properly introduced and used as a competition amongst miners. The refinement of Hashcash led to its use as a means of competition amongst miners, where each miner competes to solve the cryptographic problem and add a block to the blockchain.

The Proof of Work presents some serious drawbacks that pose a threat to its continuity (Sriman, Ganesh Kumar, & Shamili, 2021):

- Vulnerability to the 51% attack, namely the attacks carried out by a group of users or pools that, if it possesses more than half of the total computing power of the network, gets the power over the blockchain and can validate any transactions and append any block.
- Energy consumption: in order to solve the mathematical crypto puzzle, miners consume high computational power. This results in wastage of resources like hardware, space, money, and energy. The carbon footprint and the electrical energy consumption can be compared respectively to Denmark's and Austria's. The high energy consumption of PoW is not only economically unsustainable, but also has a significant environmental impact, as the energy is often generated from non-renewable sources.
- Time consumption: the process of finding the correct nonce is time consuming. It takes a minimum of 10–60 min for the confirmation of any of the transactions in the Bitcoin blockchain network.
- Scalability: slow transaction processing times and high transaction fees can limit the scalability of blockchains.

1.2.2 Proof of Stake (PoS)

Proof of Work is the most prominent consensus mechanism, but mainly due to its environmental unsustainability the Proof of Stake consensus protocol is currently gaining ground. PoS was proposed in 2011, as an alternate consensus protocol, which was later used by the crypto currency Peer coin (also known as PPcoin) in 2012, in order to eliminate the competitive approach of the PoW consensus protocol consuming a high amount of energy. While in PoW miners solve a cryptographic puzzle with the help of high computing resources, the users who intend to validate blocks in a blockchain ruled by Proof of Stake should stake or lock up as collateral the amount of cryptocurrency they hold, in a way similar to a security deposit. Proof of Stake is a protocol used to achieve distributed consensus in which voting power is proportional to the amount of cryptocurrencies, defined as stakes, held by validators (the equivalent of miners in PoW). The more coins it holds at stake, the higher is the probability that the user gets chosen as the validator of a new block. This protocol, like PoW, is fair towards validators: validators are chosen randomly, but their chance of being selected is proportional to their stake. A validator who owns 10% of the total tokens will get the right to create a new block 10% of the time. When a user is picked as validator, it builds a block of transactions and then the other network participants need to check if the new block is indeed valid. Compared to Proof of Work, Proof of Stake shows the following advantages:

- It does not require a lot of energy since it does not need the solving of any complex mathematical problem. Hence, economic costs and environmental burden are kept at bay.
- If a user attempts to include invalid transactions, he can be penalized by destroying his stake. This works as a prevention mechanism towards malicious behaviours.
- Even though theoretically Proof of Stake is also susceptible to the 51% attack, the probability of this happening is quite low. First of all, a malicious user willing to perform such attack should own more than half of the token circulating supply, which is quite costly, even more so considering that trying to acquire a large portion of the network's staked tokens would greatly increase the price of the token. Secondly, if a token holder detains more than half of the coins, it would be irrational to carry attacks to the

network. It would cause a fallout on trust in that blockchain and a drastic drop in the value of the same token he holds.

The main issue in proof of Stake (PoS) lies in reduced decentralization, as those who hold a large amount of the network's cryptocurrency are more likely to be selected as validators. Yenatfanta Shifferaw and Surafel Lemma discuss the limitation of the Proof of Stake algorithm in blockchains highlighting the less decentralized blockchain gap. If a node owns some amount of stake in the network, his stake represents its voting power in the network. Given that a large portion of the network's stake is concentrated in the hands of a few nodes, those nodes exhibit more authority in the network and can influence the networks consensus. This concentration of wealth among a smaller group of individuals could easily lead to less decentralization of the network, where a smaller number of validators control the majority of the validation power. Furthermore, centralization can lead to a lack of diversity in decision-making, as the validators may have similar interests or motivations (Shifferaw & Lemma, 2021). Therefore, PoS also carries its own set of challenges and trade-offs. These trade-offs must be carefully considered and analyzed when choosing the consensus protocol for a blockchain, as they can impact the security, efficiency, and decentralization of the network. In some cases, a combination of different consensus mechanisms or a slight modification in the traditional ones may be the best solution. Larimer proposed a consensus algorithm based on stake voting, called DPoS (Larimer, 2014). In this variation of PoS stakeholders elect a small group of nodes, called witnesses or delegates, to validate transactions and produce blocks. They are expected to also give some of the rewards to their voters. This allows for a faster process and higher scalability while still allowing a good degree of decentralization. A hybrid between PoW and PoS combines the strengths of both Proof of Work and Proof of Stake. The PoW component grants for security and the PoS component for efficiency and energy savings.

Table 2 - Comparison between Proof of Work and Proof of Stake

Property	**Proof of Work (PoW)**	**Proof of Stake (PoS)**
Validation Method	Miner solves cryptographic problems to validate transactions and create new blocks	Validators get selected based on the amount of cryptocurrency they hold and are willing to "stake"
Resource Consumption	High energy consumption due to intensive computational power required	Low energy consumption as no intensive computation is required
Scalability	Scalability is limited by the time required to find the solution	Scalability is improved
Centralization	Potential for centralization of computational power (low risk)	Potential for centralization of wealth (high risk)
51% Attack	Possible if a group of miners controls 51% of the network's computational power	Possible if a group of validators controls 51% of the network's staked tokens

1.2.3 Other consensus protocols

Although Proof of Work and Proof of Stake are the two most notorious consensus mechanisms, different types of consensus mechanisms exist and can be combined or modified to create a more effective solution for a specific blockchain. Each of these mechanisms brings its own advantages and disadvantages, and may be better suited for specific use cases depending on the requirements of the blockchain.

- **Proof of Activity:** Bentov et al. proposed a new protocol that builds upon the Bitcoin protocol by combining its Proof of Work component with a Proof of Stake type of system (Bentov, Lee, Mizrahi, & Rosenfeld, 2014). They argue that the cost of a possible attack would be much higher with the PoA protocol compared to Bitcoin's pure PoW protocol. Furthermore, the PoA protocol is likely to accomplish other beneficial properties, namely an improved network topology, incentives for maintaining full online nodes, low transaction fees, and a more efficient energy usage. The block creation process in PoA involves several steps. The philosophy of this method is giving awards both to stakeholders and miners. Miners generate empty block headers with hash of previous block, public address, height, and nonce. A miner broadcasts the block header once it meets the current difficulty target. The network determines a fixed number N of stakeholders using the hash of the block header. The first N-1 stakeholders validate the block and sign it with their private key, while the Nth stakeholder includes transactions and all signatures. The Nth stakeholder broadcasts the wrapped block, which is considered a legitimate extension of the blockchain. The fees from the transactions are shared between the miner and the N stakeholders. To perform an attack on the network, a malicious user would need to have both a significant percentage of computing power and of coins held, making an attack less likely compared to PoW.

- **Proof of burn:** it has been used as a mechanism to destroy cryptocurrency in a verifiable manner. The process consists of two functions (Karantias, Kiayias, & Zindros, 2019): the first is a function that generates a cryptocurrency address and when a user sends funds to this address, the funds are destroyed, and the second is a verification function that checks that an address is unspendable. The user burns coins in the source blockchain and subsequently creates a proof-of-burn, a short string proving that the burn took place, which she then submits to the destination blockchain to be rewarded with a corresponding amount. The more coins burned, the higher the probability of the user be selected to append a block.
- **Proof of deposit:** the validation of transactions and creation of new blocks are determined by the amount of funds the users have deposited in the network. The core idea behind this scheme is that newly minted blocks by miners are made un-spendable for a certain period of time. More precisely the coins get locked for a set number of blocks during the mining operation. The scheme works by allowing miners to perform mining at the cost of freezing a certain number of coins for some time (Bashir, 2017). This deposit acts as collateral and incentivizes the validators to act honestly, as their deposit can be penalized if they engage in malicious behaviour.
- **Proof of Reputation:** this consensus mechanism uses a node's reputation, rather than computational power, as the factor

determining the probability of a user to be selected as validator. The idea is that nodes that have a long history of trustworthy behaviour will be more likely to validate blocks correctly and keep the network secure. A protocol in which this consensus mechanism is used is RepuCoin (Yu, Kozhaya, Decouchant, & Esteves-Verissimo, 2018). In particular, RepuCoin defines a miner's power by its 'reputation', as a function of its work integrated over the time of the entire blockchain, rather than through instantaneous computing power, which can be obtained relatively quickly and/or temporarily.

- **Proof of Coin Age:** In a Proof of Coin Age system, validators are selected based on the age of their coins, i.e., the length of time they have been holding the coins. This mechanism favours users that have been in the network for longer. Since it proves that they are more invested in the network, and therefore, more trustworthy as validators.

1.2.4 Forks

In a blockchain forks may happen when nodes enter in disagreement with its status. Due to the decentralized data structure of a blockchain, blocks might arrive at different nodes at different times causing each node to create its own version of the chain. Whenever two miners solve the Proof-of-Work algorithm at approximately the same time, they broadcast their own winning block to the neighbouring nodes who propagate it across the network. This can result in two or more competing versions of the blockchain that run parallel to each other. Some nodes regard as valid one branch and the others the other one. These temporary inconsistencies between different version of the blockchain are resolved when the network eventually reaches consensus on one branch, and the blocks on the other

branch are discarded as invalid, its transactions sent back to the transaction pool waiting to be validated again. The moment one of the blockchains adds the next block before the other, all nodes join the chain of blocks that represents the most Proof-of-Work, that is the longest chain. The reconvergence ensures that there is only one authoritative version of the blockchain, preserving its integrity and consistency.
Other cases of forks occur when there is a change in the rules that govern a blockchain network's consensus mechanism. Consensus rules determine the validity of transactions and blocks in the blockchain and are responsible for the convergence of all local perspectives into a single consistent blockchain across the entire network. It's important to note that the consensus rules are not permanent and can change over time in order to accommodate new features, improvements, or bug fixes. Updating the consensus rules in a blockchain network, however, is a complex process that requires coordination between all participants. If the network does not reconverge onto a single chain after a consensus rule change and part of the network is operating under a different set of consensus rules from the rest of the network, this may either cause soft forks or hard forks. A soft fork is a backward-compatible change to the consensus rules that allows nonupgraded clients to continue to participate in the network. In order for this to happen the new rules must be a subset of the old rules, and transactions and blocks created under the new rules must also be valid under the old rules. It is not mandatory that all nodes adapt to the new rules, in fact new blocks are recognized as valid both by nodes that comply with the new rules and by those not yet updated. In the case a nonupgraded node validates a new block which complies with the old rules but not with the new ones, a temporary fork is created. However, the alternative chain will soon die out because it will not reach consensus from the upgraded nodes. In summary, implementing changes to the network while

maintaining compatibility with older software versions gives raise to soft forks. Hard forks are non-backward-compatible changes to the consensus rules that hinder nonupgraded clients to continue to participate in the network. Nodes deciding not to upgrade to the new consensus rules get excluded from the consensus process and are forced onto a separate chain at the moment of the hard fork. The new consensus rules represent an extension of the old ones. Blocks that were previously considered invalid will be accepted, and nodes that don't upgrade to the new protocol won't be able to recognize them as valid. Once a miner using the new rules mines a block, the chain will fork diverging in two separate chains. New miners will mine on top of the new block, while old miners will mine a separate chain based on the old rules. If the community does not move on the new version in its entirety, a fracture of the community can occur, as happened for example between Bitcoin and Bitcoin Cash[1], and between Ethereum and Ethereum Classic[2]. In this case the two derived blockchains maintain the history of the transactions prior to the moment of the fork, and users can find themselves with the same amount of two different cryptocurrencies on the two different blockchains.

[1] The hard fork was caused by a disagreement over the measures to adopt to overcome Bitcoin's scalability issue (Nyffenegger, 2018). A party proposed the adoption of the Segregated Witness (SegWit) upgrade, the other party pushed for an alternative plan that would increase the block size limit to eight megabytes. This lead to the creation of Bitcoin Cash.

[2] In June 2016, a vulnerability in the code of the DAO, decentralized autonomous organization on the Ethereum network, was exploited, leading to the loss of approximately 3.6 million Ether. One part of the community decided to perform a hard fork and reverse the attack, while the other did not want to invalidate the principle of immutability of the blockchain. The result was a hard fork that created two blockchains: Ethereum, the forking blockchain, and Ethereum Classic (the original blockchain (Siegel, 2018).

2 Blockchain industry applications

Blockchains have the potential to be applied in a plethora of other applications beyond cryptocurrencies, as it encapsulates unique properties including decentralization, security, transparency and anti-tampering. Some of the most relevant fields where blockchains are being applied or have the potential to be applied include finance, supply chain management, project management, Internet of Things, identity management, healthcare, government, education, real estate, art and media and so on. This chapter will analyse some of them in more detail.

2.1 Finance

Blockchain may provide a potentially attractive alternative way to organize the financial system by eliminating the need for centralized trusted intermediaries such as central counter parties (CCPs), central securities depositories (CSDs), the Society for Worldwide Interbank Financial Telecommunication (SWIFT), CLS Bank, and so on. These intermediaries have been traditionally trusted to ensure the smooth functioning of the financial system and it was assumed that they were too big to fail (TBTF), so that the government would step in and bail them out if necessary. However, their failures during the Global Financial Crisis of 2007–2008 and instances of hacking of the computers of large financial institutions shattered these assumptions. While cryptocurrencies have been able to quickly gain acceptance due to their technological complexity, even a decade after the launch of Bitcoin, only a few other potential applications of blockchain in the financial sector have been implemented even though technologically feasible, all because of legal, regulatory, institutional, and commercial barriers (Varma, 2019). Blockchain technology was first used for digital currencies like Bitcoin, offering a fully decentralized issuance of currency, and traceable payments. Bitcoin's success has led to the creation of many other digital currencies that are based on similar technology. Currently there are over 600 different digital currencies that use blockchain technology as their underlying technology layer (Best, 2022). Blockchain technology can be used to create decentralized exchanges, which allow users to buy, sell, and trade a variety of digital currencies, including Bitcoin, Ethereum, and Litecoin. Decentralized exchanges operate without a central authority or intermediary, using smart contracts to facilitate transactions between users. Uniswap, Kyber Network, Binance DEX and IDEX are examples of decentralized exchanges. Companies such as Binace, Coinbase, Crypto.com are instead examples of centralized digital currency exchanges. In a paper covering the topic of the financial sector applications of blockchain technology beyond cryptocurrencies, Ariana Polyviou, Pantelis Velanas and John Soldatos discuss five use cases in which blockchain technology is expected to have a significant impact on the financial sector in the future (Polyviou, Velanas, & Soldatos, 2019). All financial organizations initiate KYC and KYB processes whenever they onboard a customer to verify and identify new customers according to national and international regulations. Customer documentation can be centrally maintained by an authority, but this solution is vulnerable to cyber-attacks and data breaches. Blockchain technology can improve the KYC process by securing customer data in a distributed ledger, enabling financial organizations to access up-to-date customer information at all times. Blockchain solutions offer advantages such as decentralization, improved privacy control, and immutability.

Banks are seeking new approaches to credit scoring for SMEs beyond traditional finance and accounting data (e.g., P&L balance sheets).

Blockchain technology can enable the secure sharing of credit scoring information from multiple parties, such as banks and credit risk assessment organizations, to improve the accuracy of credit risk assessments and facilitate lending decisions as more banks collaborate on the blockchain.

Blockchain technology can improve customer profiling and product personalization in the financial industry by enabling the secure sharing of data across institutions. This can lead to the development of more tailored asset management recommendations and retail banking products. Blockchain infrastructures can also serve as the basis for personal data markets, where customers can exchange access to their data for incentives from financial institutions.

The insurance sector is very closely affiliated to the finance sector. Insurance claims management is a lengthy and tedious process, which blockchain technology can streamline by integrating all stakeholders on a distributed ledger infrastructure and implementing smart contracts for checks, verifications, calculation and validation of the amount to be paid. The process can also be enhanced with the inclusion of multimedia evidence and driver performance assessment scores based on vehicle data like acceleration, steering drive, speed and brake patterns.

Finally, the financial services industry is a primary target for cyber criminals. By enabling trustful sharing of security information through a distributed ledger, financial institutions can collaborate and better protect their critical infrastructures from cyber-attacks.

2.2 Supply chain management

A supply chain is a system of organizations, people, activities, information and resources involved in transforming natural resources and raw materials into a finished product for delivery to the end customer. Stock and Boyer, after examining 166 definitions of CSM from the literature, presented an encompassing definition for SCM as: “The management of a network of relationships within a firm and between interdependent organizations and business units consisting of material suppliers, purchasing, production facilities, logistics, marketing, and related systems that facilitate the forward and reverse flow of materials, services, finances and information from the original producer to final customer with the benefits of adding value, maximizing profitability through efficiencies, and achieving customer satisfaction (Stock & Boyer, 2009)”

Between the market expansion, the growth in suppliers’ relationships, the rising consumer demand, and the growing number of intermediaries between manufacturers and end consumers, supply chain management has become more complex and new challenges have emerged. There are billions of products being produced globally every day through complex supply chains that span the entire world. However, there is often lack of knowledge about the origins, processing, or shipping journey of products. Key supply chain management objectives such as cost, quality, speed, dependability, risk reduction, sustainability and flexibility need to be addressed in an efficient manner (Kshetri, 2018).

According to Abeyratne and Monfared (2016), the primary challenge in the supply chain is traceability and data management. End-to-end supply chain transparency and visibility can help track the flow of products from raw materials to manufacturing, testing, and finished goods, enabling new types of analytics for operations, risk, and sustainability. Transparency in supply chains can help address the negative consequences of the manufacturing industry, such as environmental damage, waste, unethical labor practices, and counterfeit products. With this goal, sustainability standards and certifications have become important tools. However, they might be subject to corruption and verifying the claims made by these certifications can be costly since it requires thorough auditing. Traceability and data management are frequently centralized and managed by non-profit, governmental entities, or other third parties through centralized information depositories. Relying on a single organization to broker sensitive and valuable information requires a high level of trust from all parties in the supply chain, since it could, through the possession of these data, harm or extort other organizations if biased. Additionally, this centralized approach creates a single point of failure that leaves the entire system at risk due to hacking or corruption. In summary, centralized supply chain management systems expose the supply chain to corruption, fraud, and tampering. Abeyratne and Monfared conclude that the blockchain technology can potentially improve the transparency and traceability issues within the manufacturing supply chain through the use of immutable record of data, distributed storage, and controlled user accesses.

Combining theoretical and real-world application studies, Rita Azzi, Rima Kilany Chamoun and Maria Sokhn (2019) describe how the blockchain can be integrated into the supply chain architecture to create a reliable, transparent, authentic and secure system. Tracking systems have evolved from paperwork to Internet of things (IoT) hardware and sensors, and their main components are the tag (e.g. RFID and QR codes), the tracer (substance providing information about the quality of a product) and the sensor (a device that detects environmental changes). However, tracking devices are sometimes compromised and subject to cloning. Cloned tags on counterfeit products can mislead the consumers and endanger the consumers' safety especially in a medical or food industry. For this reason, according to Toyoda et al. standard track and trace methods cannot guarantee that products in retail stores are genuine (Toyoda, Mathiopoulos, Sasase, & Ohtsuki, 2017). According to data from the Global Trade in Fakes report by the OECD and EUIPO, trading with counterfeit goods amounted to roughly $449 billion in 2019, equivalent to 2.5 percent of the world trade and roughly six percent of imports into the European Union (Zandt, 2022). RFID tags can be used to store information about a product, such as its origin, quality, and location, and this information can be transferred to a blockchain-based product ownership management system. The RFID reader can then be used to access and verify the information stored on the blockchain to create a reliable, transparent and secure decentralized platform, where all supply chain actors can interact.

Traditional enterprise resource planning (ERP) technology has limitations in terms of transparency, flexibility, data accessibility, and advanced decision-making. A cloud-based NetMES system may solve some of these problems, but being a centralized virtual database replacing a centralized physical database, it remains a single point of failure entity. It’s not the case with the distributed ledger where a hacker cannot take advantage of a vulnerable point; if one node fails, the remaining nodes will not be affected.

The main challenges that may be blocking the adoption of blockchain in the supply chain include the high cost of implementation and the scalability issue. In fact, the cost of the implementation of a RFID tag system is quite high. Transaction’s rate in blockchains is usually limited (seven transactions per second in the Bitcoin network), which may not be feasible for large-scale supply chain operations.

2.3 Internet of things

The term "Internet of Things" (IoT) was coined by British technology pioneer Kevin Ashton in 1999 to describe a system in which objects in the physical world could be connected to the Internet by sensors. The concept of connecting devices and networks to monitor and control objects has been around for decades, but early solutions were based on closed networks and proprietary standards, rather than Internet Protocol (IP)-based networks and Internet standards. The idea of using IP to connect devices other than computers to the Internet is not new, examples are a soda machine at Carnegie Mellon University in the US and a coffee pot in the Trojan Room at the University of Cambridge in the UK, which remained connected to the Internet until 2001 (Rose, Eldridge, & Chapin, 2015). The Internet of Things (IoT) refers to the network of physical devices, vehicles, buildings, and other items embedded with electronics, software, sensors, and connectivity which enable these objects to connect and exchange data. A *smart object*, which is the building block of the Internet of Things, is just another name for an embedded system that is connected to the Internet. These devices range from everyday household items such as smart thermostats and appliances, to industrial equipment, medical devices, and even entire buildings. Overall, the Internet of Things has the potential to greatly improve efficiency and productivity in a wide range of industries, as well as improve the quality of life for individuals through the use of smart home technology. However, it also brings security and privacy concerns which need to be taken into account. Examples can be given to clarify possible use cases: the home heating system is activated only if the temperatures predicted by the weather forecasts are below a certain threshold; alarm clocks go off early if heavy traffic is detected; the car decelerates if it comes to a stretch of road where the maximum allowable speed is lower than the one the car is speeding at or advises the driver to stop if it senses his fatigue; the navigator advises a different route if there is a lot of traffic or there has been an accident along the usual route; a wearable device for elderly care that can monitor heart activity and vital parameters activates prompt action in case of an emergency. Blockchain technology has been growing at an astounding pace over the past years. As reported by Statista, the number of Internet of Things (IoT) devices worldwide is forecast to almost triple from 9.7 billion in 2020 to more than 29 billion IoT devices in 2030. By 2030, China is expected to have the largest number of IoT devices, with approximately 5 billion consumer devices. The most important use case for IoT devices in the consumer segment are consumer internet & media devices such as smartphones, where the number of IoT devices is forecast to grow to more than 17 billion by 2030. Other use cases with more than one billion IoT devices by 2030 are connected (autonomous) vehicles, IT infrastructure, asset tracking & monitoring, and smart grid (Vailshery, Number of IoT connected devices worldwide 2019-2021, with forecasts to 2030, 2022). The total Internet of Things (IoT) market in 2021 was worth around 181 billion U.S. dollars and is set to rise to more than 622 billion U.S. dollars in 2030. The consumer sector continues to dominate and is forecast to generate 183 billion U.S. dollars in revenue by 2030 (Vailshery, IoT global revenue 2020-2030, by vertical, 2022). Overall, while the Internet of Things has the potential to bring significant benefits, there are several key issues that must be addressed in order for it to continue to grow and be successful. The current centralized architecture introduces numerous challenges involving a single point of failure, security, privacy, transparency, interoperability, scalability and cost (Atlam, Azad, Alzahrani, & Wills, 2020). A single point of failure implies that if the central server goes down, all associated IoT applications and services will also be affected in availability and quality. According to Ashok Kumar Reddy Nadikattu (2018), security is a concern since a centralized server, which stores data from various IoT devices in one location, makes it an attractive target for attackers. The fact that IoT devices are connected to the internet makes them vulnerable to cyber-attacks, and as more devices connect, there is less control over the data being collected and shared. Furthermore, maintaining data privacy is questionable because IoT devices generate a high amount of data, creating entry points for hackers to access sensitive information belonging to the organization and clients. There is no certainty that personal data is used responsibly and secured by a third-party provider. IoT scalability raises question marks. As the number of IoT devices grows, it can be challenging for data centres and IoT networks to manage and process the large amounts of data they generate. Blockchain technology brings the opportunities in addressing the challenges of IoT. The integration of blockchain technology with IoT has been labelled as blockchain of things, BCoT (Dai, Zheng, & Zhang, 2019). Atlam, Azad, Alzahrani and Wills (2020) affirm that moving the IoT into one of the distributed ledger technologies may be the correct choice, and among them they propose the blockchain. In fact, the use of decentralized and distributed attributes of blockchain technology can address the concerns related to security and to a single point of failure associated with the centralized IoT architecture, as there is no need for a central server to control IoT devices and their communications with each other. Furthermore, blockchain delivers improved security and privacy through the use of cryptography, hash functions, and timestamps. Additionally, the blockchain's tamper-proof and immutable ledger safeguards data against harmful attacks such that data change can only be stored in the ledger if the majority of contributing users accept it. Blockchain technologies can be applied in different areas where IoT applications are involved like sensing, data storage, identity management, smart living applications, intelligent transportation systems, wearables, supply chain management, and so on (Fernández-Caramés & Fraga-Lamas, 2018). Han Donhee, Hongjin Kim, and Juwook Jang (2017) propose a Blockchain based smart door lock system to deal with the issue that data sent and received by existing Smart Door Lock system are vulnerable to forgery and hacking. The proposed system does not require a central server to control IoT devices and their communications with each other, and it delivers better security and privacy through the use of sophisticated cryptography algorithms, hash functions and timestamps. In the paper *Thing-to-thing electricity micro payments using blockchain technology* (Lundqvist, De Blanche, & Andersson, 2017) propose an application of a blockchain to IoT or to the Internet of Energy (IoE). The paper discusses the use of blockchain technology in facilitating micro payments for electricity consumption between connected devices. The paper describes a proof-of-concept implementation of a smart cable that connects to a smart socket and without any human interaction pays for electricity. The authors identify several obstacles for the widespread use of traditional cryptocurrency such as Bitcoin in thing-to-thing payments, such as high transaction fees for micro-transactions. To address this issue, the researchers present a single-fee micro-payment protocol that aggregates several small payments into a larger transaction, thus reducing the impact of transaction fees. The proof-of-concept demonstrates that trustless, autonomous, and ubiquitous thing-to-thing micro-payments is possible using blockchain technology. Finally, a use-case of blockchains in the pharma supply-chain can be discussed (Bocek, Rodrigues, Strasser, & Stiller, 2017). The authors present a traceability application from a start-up called modum.io which uses IoT sensor devices leveraging blockchain technology to ensure data immutability and public accessibility of temperature records, while also reducing operational costs in the pharmaceutical supply-chain. This verification is critical for the transport of medical products in order to ensure their quality and environmental conditions (i.e., their temperature and relative humidity) due to the many complex and strict environmental control regulations that the medical industry has. The sensor devices monitor the temperature of each parcel during the shipment to fully ensure GDP regulations. All the data collected by the sensors are then transferred to the blockchain, where a smart contract assesses it against the product attributes.

3 Cryptography

The word "cryptography" comes from the Greek words "kryptós," which means "hidden," and "gráphein," which means "to write." Therefore, cryptography is defined as the science of secret writing and refers to the practice of writing in code using mathematical algorithms and techniques to transform a message in a form that cannot be intercepted and modified ensuring its confidentiality, integrity, and authenticity. The practice of cryptography has a long history dating back to ancient civilizations, where it was used to protect sensitive information and communications from being understood by unauthorized individuals. For a long time, it was considered as an art, and only became a science in the 20th century. It was the massive use of computers that democratized its use. The following are the main problems existing within communication and its main aims (Agrawal & Mishra, 2012):

- Confidentiality: ensuring that information is only accessed by authorized parties and not by anyone else
- Authentication: verifying the identity of the sender of a message to ensure that it is not being sent from a false identity.
- Integrity: ensuring that no one apart from authorized parties, sender and receiver, can modify the message.
- Non-repudiation: Ensuring that neither the sender nor the receiver of a message can deny the transmission.

Cryptography encompasses both ciphering, the process of encrypting information to protect its confidentiality, and deciphering, the process of decrypting it to restore its original form. Ciphering converts readable information, called plaintext[3], into unreadable ciphertext[4] by means of a mathematical algorithm and a secret key. Deciphering converts ciphertext back into the original plaintext form using the same key that was used to encrypt it, restoring the original, readable form of the information. Based on the key type, cryptography can be classified into two groups: symmetric (or private) key and asymmetric (or public) key cryptography.

3.1 Symmetric key cryptography

In symmetric key cryptography, also known as private key cryptography, the same key is used for both the ciphering (encryption) and deciphering (decryption) processes. Therefore, a pair of users intending to communicate between themselves must share the same secret key. This is a fast and efficient system, which however shows some criticalities. Firstly, it requires the exchange of the secret key between sender and receiver. This can be a security risk, as the key can potentially be intercepted or stolen during the exchange process. A physical key exchange would eliminate such risk, however it is not always possible if users are many and distant from each other, as in most modern scenarios. Secondly, symmetric key cryptography holds a scalability issue. Since every pair of users must share a unique key, the number of keys necessary to allow the safe exchange of messages between *n* users grows proportionally to the square of *n*:

$$\binom{n}{2} = \frac{n!}{2!\,(n-2)!} = \frac{n(n-1)}{2} \sim \frac{n^2}{2}$$

[3] The original message that the person wishes to communicate with the other is defined as plaintext. Anna wants to send the message “Hi Bob, how are you?” to Bob, in this example “Hi Bob, how are you?” is the plaintext.

[4] The ciphertext is the decrypted text which can only be accessed by someone who has the correct decryption key or algorithm. For example, “Ajd672#@91ukl8*^5%” is a ciphertext produced.

This can become impractical for large numbers of users, as the number of keys required can become very large.

Thirdly, most of the symmetric key algorithms contain a large number of rounds, which can lead to longer processing times and slower performance. Another disadvantage is that they do not provide data origin authentication and data integrity protection. In other words, the recipient can neither authenticate the sender nor verify that the decrypted message is the same as the original message and finally he cannot provide digital signatures which means that they cannot be used to provide non-repudiable evidence of the authenticity of a message (Joseph, Krishna, & Arun, 2015)

There are various symmetric key algorithms including enigma (involves shifting letters), the one-time pad, Data Encryption Standard (DES), 2DES, 3DES and the Advanced Encryption Standard (AES) (Lin, 2010). Symmetric encryption algorithms are almost 1000 times faster than asymmetric algorithms because they require less processing power for computations (Hardjono & Dondeti, 2005).

3.2 Asymmetric key cryptography

In asymmetric key cryptography, also known as public key cryptography, two different keys are used for the ciphering and deciphering processes. It is called "asymmetric" because the keys used for encryption and decryption are not the same. A key is a numeric or alpha numeric text or may be a special symbol, usually represented in the hexadecimal numeral system. One key, known as the public key, is used for encryption, and the other key, known as the private key, is used for decryption. The public key is made widely available to anyone wishing to send an encrypted message since the sender must know the recipient's public key so that he can encrypt the message. The private key is instead kept secret and only known by the owner, so that he is the only one able to decrypt the received message using his own private key.

When A wants to send a message to B, the following steps are involved:

- It is of the utmost importance that A and B know the public key of each other, but private keys are kept secret;
- A encrypts a plaintext message for B using B's public key;
- The newly generated ciphertext gets transmitted to B;
- B receives the ciphertext and decrypts it using his own private key;
- B can now read the plaintext message.

Every user needs to generate autonomously his pair of keys. The private key gets generated randomly and must remain secret. The public key derives mathematically from the private key and can be shared with anyone. Public key cryptography is based on the idea of a one-way function, which is a function that uses a relatively small amount of computing power but whose inverse function is extremely expensive to compute, so that an attacker is not able to derive the original plaintext from the transmitted cipher text within a reasonable time frame. This means that it is easy to compute the public key from the private key, but it is computationally infeasible to determine the private key from the public key. Asymmetric key cryptography solves the two main issues regarding symmetric key cryptography. Firstly, it is no longer needed that the sender and the receiver of a message safely share a secret key in advance, since public key cryptography is not vulnerable to attacks where a malicious user intercepts the secret key and uses it to decrypt the message. In fact, the latter can only be decrypted by means of the private key, which remains secret and only known by the receiver. Thus, asymmetric key cryptography is more secure than symmetric key cryptography. Secondly, this system requires the generation of fewer keys overall, as all users can use the same public key to send messages to a particular user. In a symmetric key system, to do so, each user would have to use a different key for every recipient. It is important to note that in an asymmetric system, since each user has two keys, the number of keys necessary to maintain the system still increases linearly with the number of users in the network. The total number of keys is, in fact, equal to 2*n,* which in case of high number of users in the system turns out to be much less than the approximately $\frac{n^2}{2}$ needed in a symmetric system (Bazzanella, 2021).

However, there are also some aspects for which public key cryptography may be more undesirable compared to private key cryptography (Ketu File white papers, 2004):

- A symmetric encryption algorithm, like AES, runs faster compared to asymmetric key algorithms. Also, it may require fewer CPU cycles and less memory to encrypt and decrypt compared to RSA (asymmetric).
- Some asymmetric encryption systems may have key recovery built in, which raises concerns about who has access to this key recovery.
- Asymmetric encryption schemes are based on the multiplication of two large prime numbers. Cracking a message encrypted with this scheme involves factorizing the original two numbers. Advances in mathematical techniques and CPUs, including those dedicated to cracking, continue to be made. These efforts have threatened the confidence in RSA and other public key encryption systems.
- When using asymmetric encryption, a sender must request the recipient's public key and use it to encrypt the message. However,

if the public key is intercepted by an attacker, they can create a message and send it as if it came from the intended party.

- A direct consequence of the last bullet point is that asymmetric encryption does not inherently allow for non-repudiation, the ability to provide evidence that a message was sent or received by a specific party.

Elliptic Curve, Diffie-Hellman (DH), DSA and RSA are the most notorious examples of asymmetric algorithms. NTRU is one of the most recently developed public key algorithms. It is the only algorithm that is resistant to quantum algorithm attacks, thus making it significantly more secure than other algorithms, but it has not experienced widespread implementation. Despite the advent of many new public key algorithms, RSA continues to have the highest popularity in implementation, with a popularity of 43% (Gaithuru, Bakhtiari, Salleh, & Muteb, 2015).

3.3 Digital signatures

A digital signature is a string of ones and zeroes generated by using an algorithm useful for verifying the authenticity and validation of digital messages or documents. Traditionally, paper documents are validated and certified by written signatures, which work fairly well as a means of providing authenticity. For electronic documents, a similar mechanism is necessary. Digital signatures may be regarded as the cryptographic analogue of handwritten signatures and are increasingly being accepted as legally binding in many countries. In fact, they are getting used for certifying contracts or notarizing documents, for authentication of individuals or corporations, and as components of more complex protocols. In addition, digital signatures enable the secure distribution and transmission of public keys, which are an essential component of public-key cryptography (Katz, 2010). By providing a secure and tamper-proof way to distribute public keys and verify their authenticity, digital signatures prevent the risk that an attacker could impersonate the owner of the key and alter messages or transactions after having intercepted them. In order to provide a secure way to verify the authenticity and integrity of digital messages, digital signatures should have the following properties (Subramanya & Yi, 2006): it should be a unique pattern of zeroes and ones that depends on the specific message or document being signed, that is the digital signature should be different for different documents; it must use some information that is unique to the sender, such as the private key, to prevent both forgery - only the true owner of the private key can create a valid digital signature - and denial - the sender cannot later deny having signed the message or document; it must be easy to produce and include in messages; the recipient should recognize it with effortlessness; it must be computationally infeasible to forge a digital signature either by constructing a new message for an existing digital signature or creating a fraudulent digital signature for a given message; keeping copies of the digital signature in storage for later use in arbitrating possible disputes should be straightforward. Digital signatures are created combining hashing and public key cryptography. The process of creating and verifying a digital signature can be divided into two main steps: signing and encryption, and decryption and verification (Nist, 1992).

1. Signing and encryption:

   A hash function is used in the signature generation process to obtain a condensed version of the message, also known as a message digest. The signature function uses the message digest and the sender's private key to generate the digital signature. A very simple form of the digital signature is obtained by encrypting the message digest using the sender's private key. The digital signature is sent to the intended recipient along with the original message.

2. Decryption and verification:
   The recipient of the message and signature decrypts the signature by using the sender's public key to recover the original message digest. The received message is subjected to the same hash function to which the original message was subjected. The resulting message digest is compared with the one recovered from the signature. If the two hashes match, it means that the message or document has not been altered and is authentic. If the hashes do not match, it means that the message or document has been tampered with and is not authentic. The same hash function must be used in the verification process.

Digital signatures serve four main purposes (Kaur & Kaur, 2012):

- Privacy or confidentiality: a message or a transaction between two parties cannot be viewed or interfered with by a third party.
- Authentication: since a private key is associated to one specific user, a valid signature demonstrates unambiguously that a message or transaction was sent by its owner. This helps to establish the identity of the sender and ensure that the message or document is genuine.
- Nonrepudiation: who signs a message or document cannot later deny having sent it. Thus, when the message is sent to the receiver, the receiver can prove that the alleged sender in fact sent the message.
- Integrity: Digital signatures help to ensure the integrity of a message or document since it proves that a transaction (or specific

parts of a transaction) has not and cannot be modified by anyone after it has been signed. This is done by computing a hash of the message or document and comparing it to the decrypted hash included in the digital signature. If the two hashes match, it means that the message or document has not been modified.

4 Smart contracts

A contract is a legally binding agreement between two or more parties. Written documents that are signed and executed by the parties involved, and enforceable through the legal system, are the traditional way to formalize the "meeting of the minds". In the 1990s, computer scientist and legal scholar Nick Szabo suggested that the digital revolution would make possible new ways to formalize relationships between parties. He coined the term smart contract and defined it both as “a computerized transaction protocol that executes the terms of a contract” and as “set of promises, specified in digital form, including protocols within which the parties perform on the other promises”. He also laid out the general objective of smart contracts: to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, minimize the need for trusted intermediaries, and lower fraud loss, arbitration and enforcement costs, and other transaction costs (Szabo, 1996). So, compared to the traditional ones, smart contracts guarantee both a higher level of security and a reduction in transaction costs, as intermediaries especially in the form of centralized authorities, often turn out to be inefficient, slow and expensive. He decided to call these new contracts "smart", because they are far more functional than their inanimate paper-based ancestors. Since then, the concept of smart contracts has evolved, especially after the introduction of decentralized blockchain platforms with the invention of Bitcoin in 2009. In their book titled *Mastering ethereum: building smart contracts and dapps,* Andreas Antonopoulos and Gavin Wood suggest to use the term smart contracts to refer to immutable computer programs that run deterministically in the context of an Ethereum Virtual Machine as part of the Ethereum network protocol - i.e., on the decentralized Ethereum world computer (Antonopoulos & Wood, 2018). In particular, in this definition they emphasize on the immutable and deterministic nature of smart contracts. Just like in blockchain technology, immutability is one of the key features of smart contracts. It means that once deployed, the code of a smart contract and its stored data cannot be altered. Unlike with traditional software, the only way to modify a smart contract is to deploy a new instance. This makes the contract tamper-proof and provides a high degree of security and trust. It is important to note that while the code of a smart contract cannot be changed, it is still possible to modify its behaviour by deploying a new version of the contract with the desired changes. This ensures that the immutability of the contract is maintained, while still allowing for updates and improvements to be made over time. The fact that a smart contract runs deterministically in the context of an Ethereum Virtual Machine assures that the outcome of its execution is the same for everyone who runs it being only determined by its code, the input data provided by the transaction that initiates its execution, and the current state of the Ethereum blockchain. The rules and conditions set forth in the contract are automatically enforced by the network, ensuring that the contract is executed as intended and that the outcome is fair and predictable. Traditional contracts allow parties to have discretion over whether to fulfil the obligations of the contract, whether to implement the contract only partially (by leaving out some obligations) or whether to breach the contract and pay instead for damages or compensation. On the other hand, with smart contracts, parties have no choice but to fulfill the terms of the agreement because they have been encoded, written into the code. It cannot be breached unless one actually manages to break into the code (De Filippi, 2015). To summarize, a smart contract is a transparent (its code is open source; anyone can examine it) and self-enforcing software with all the characteristics of a real-world contract, which is stored and replicated on a blockchain network and can be programmed to automatically trigger actions based on the specific contractual conditions being met, eliminating the counterparty risk and the need for trust in intermediaries. In fact, there is no need to rely on central authorities, their role is replaced by the consensus of the network. A smart contract can be seen as an IFTTT (If This Then That) application that reacts to certain events, usually in the form of transactions. Based on their popularity in the developing community and level of technical maturity, Ethereum, Hyperledger Fabric, Corda, Stellar, Rootstock, and EOS are recognized as the major smart contract development platforms (Zheng, Xie, Dai, Chen, & Chen, 2020). Here are a few examples of how smart contracts can be used in real-world cases:

- Managing real estate transactions, from offer to closing, automatically executing the transfer of ownership and handling payments once the agreed-upon conditions are met.
- Home food delivery: a smart contract may grant a discount on the order based on how long the delivery took (for example a 50% discount if delivery took over 40 minutes). Discount conditions and a timestamp identifying the time at which the order was made get saved on the smart contract. When a customer places an order, the full amount is blocked within the smart contract and, once the delivery is complete, the actual price to be paid is established based on the delivery time.
- Smart contracts can be used to automate the claims process for insurance policies, ensuring that claims are processed efficiently and fairly, without the need for intermediaries.

There are some downsides associated to smart contracts. The decentralized and autonomous nature of smart contracts means that they operate independently of legal systems and are not subject to the same legal protections and safeguards that traditional contracts are. For instance, there are many situations in contract law that might either invalidate the contract (if it was agreed to under undue influence, for example) or limit its enforceability (to the extent that it goes against the interests of consumers). But smart contracts operate within their own closed technological framework, relying on code to enforce the terms of the agreement and ensure compliance, which does not necessarily incorporate these legal safeguards. In this sense, smart contracts could effectively bypass the legal framework of contract law (De Filippi, 2015). According to Nick Szabo, most of the contractual disputes involve unforeseen events. Smart contracts are, however, not designed to include all life events in the code which may influence the performance of contract and give just ground for refusal under the traditional contract law. It may be possible to include force majeure events in the code, but it’s unreasonable to assume to be able to include all eventualities. As the performance of contract takes place automatically under the code, the code must either account for certain real-world events and circumstances (e.g., force majeure events) to follow current contract law principles or leave a cap to the code, which makes it insufficient (Kerikmäe & Rull, 2016). Furthermore, writing reliable smart contract code is difficult even for those with a strong background in computer science and software engineering, this may make them vulnerable to hacking due to the possibility of bugs and errors in the code. Scalability, too, remains an issue with smart contracts. As the number of users and transactions on a blockchain grows, it can become difficult for smart contracts to handle the increased load and maintain acceptable performance. Finally, up to this moment, smart contracts lack legal recognition in many jurisdictions, where they are not recognized as legally binding agreements and therefore it’s not possible to enforce them in a court of law.

4.1 Tokens

The etymology of the word "token" comes from the Middle English word “tācen”, meaning a sign or symbol. In the general sense, the word "token" refers to a physical object, mainly privately issued special-purpose coin-like items, that is used as a symbol to represent something else, like laundry tokens and arcade game tokens. They usually have insignificant intrinsic value. Physical tokens are often specific to a certain business, organization, or location, and they typically have only one function, such as accessing a restricted area or paying for a particular service. In the blockchain realm, a token is a unit of digital representation for a certain asset or utility created by combining the potential of a smart contract with a cryptocurrency. Differently from the physical version of tokens, blockchain tokens do not have the “insignificant intrinsic value” restriction, as they can be traded for each other or for other assets, just like cryptocurrencies, on global liquid markets and usually serve multiple functions. In February 2018, the Swiss Financial Market Supervisory Authority (FINMA), the Swiss government body responsible for financial regulation, released ICO guidelines and divided the nature of tokens into three categories in the attempt to regulate ICO tokens depending on their nature (FINMA, 2018):

- Payment tokens are digital private currencies with no further functions or links to other development projects. They are designed to function as a medium of exchange, much like traditional fiat currency or other forms of digital currency, and may become accepted as a means of payment over time.

- Utility tokens are tokens which are intended to provide digital access to an application or service within a particular blockchain platform or ecosystem. Unlike payment tokens, which are designed to function as a medium of exchange, utility tokens do not have a fixed value and are not designed to be used as a store of value. Instead, their value is tied to the perceived usefulness and demand for the underlying product or service they provide access to.
- Asset or equity tokens represent assets such as participations in real physical companies or an entitlement to dividends or interest payments. In terms of their economic function, the tokens are analogous to traditional stocks, bonds and derivatives. They provide holders with a share of the profits, revenues, or other financial benefits generated by the underlying asset. Equity tokens may be as limited as nonvoting shares, providing only the right to receive dividends or profits, or as expansive as voting shares in a decentralized autonomous organization allowing holders to participate in governance and management decisions.

A further distinction can be made between fungible and non-fungible tokens (NFTs). Fungibility is the ability of a good or asset to be interchanged with other individual goods or assets of the same type. Cash or other forms of currency are fungible, as one dollar bill is interchangeable with another dollar bill. Tokens are fungible when we can substitute any single unit of the token for another without any difference in its value or function. Non-fungible tokens (NFTs) represent a unique tangible or intangible item and therefore are not interchangeable on a one-to-one basis. NFTs are often used to represent ownership of a digital asset, such as a piece of artwork, a collectible item, or a piece of virtual real estate. Each NFT is cryptographically verified to be one-of-a-kind, and its ownership is recorded on the blockchain.

4.2 Oracles

The term oracle derives from the Latin verb *ōrare,* meaning “to speak”. The word comes from Greek mythology, where it referred to the agency or medium, usually a priest or priestess, through whom a deity would confer visions of the future. In the context of blockchains, oracles are entities that provide external data and information, such as the result of a football game, exchange rates or the price of a token, to smart contracts. The latter need access to extrinsic information, in order to automate the execution of transactions based on the specified conditions. Oracles bridge the gap between the blockchain and the outside world, providing the necessary information and data to trigger the execution of a smart contract. If a person A was to bet a person B that it is going to rain all days of the following week, the bet amount would get locked in a smart contract, the oracle would provide weather information for each of the following days of the week, and based on these data, the smart contract would deliver the funds to the winner of the bet. There are two main types of oracles in the world of blockchain: software oracles and hardware oracles. Software oracles retrieve and deliver data from digital sources such as websites, servers, or databases, while hardware oracles retrieve and deliver data from the physical world. Types of data that software oracles provide are exchange rates, price fluctuations, and so on. Examples of data that hardware oracles provide are information from camera motion sensors, radio frequency identification (RFID) sensors, thermometers, and so on. Oracles can be centralized or decentralized. Centralized oracles relay on a single entity as the sole provider of data for a smart contract. They require contract participants to place a significant amount of trust in one single entity. The issue is that they can be vulnerable to manipulation, hacking and tampering, thus, threatening the security of a smart contract: if a centralzed oracle is compromised, so is the smart contract. Since smart contracts and decentralized applications rely on external data to trigger their execution, it is important to have a trusted and secure source of information. This is known as the "oracle problem". Decentralized oracles try to overcome the oracle problem in the same way a blockchain network achieves security and reliability: by distributing trust among many network participants. They allow multiple participants to provide data to the network, which can also be incentivized to provide accurate information as they risk losing their reputation and rewards if they provide false or unreliable data. This provides a secure and trustworthy way of accessing external data in blockchain applications, as the data is validated by the consensus of the network participants, rather than relying on a single, centralized entity. Once the data has been validated, it can be used to trigger smart contracts. Blockchain projects that are working to develop (or have developed) decentralized oracles are Chainlink, Band Protocol, Augur, and MakerDAO. ChainLink has an on-chain component consisting of three main contracts - a reputation contract, an order-matching contract, and an aggregating contract - and an off-chain registry of data providers. The reputation contract keeps track of oracle-service-provider performance metrics. The order-matching smart contract matches incoming data requests from smart contracts with available oracles. It is responsible for ensuring that the most suitable oracle is selected based on factors such as reputation, fees, and the type of data required. It then finalizes a service-level agreement, which includes query parameters and the number of oracles required. The aggregating contract collects the oracle providers’ responses, calculates the final collective result of the ChainLink query and finally feeds oracle provider metrics back into the reputation contract. The off-chain registry of data providers is a list of all the oracles that are available on the network. In summary, the three key smart contracts and the off-chain registry work together to provide a secure, reliable, and efficient solution for connecting smart contracts to real-world data and events (Ellis, Juels, & Nazarov, 2017).

4.3 Decentralized applications (DApps)

Decentralized Applications (DApps) are a new class of software applications that run on a decentralized network, typically built on blockchain technology, free from control and interference by any single authority. Standard web applications rely on centralized servers and data storage owned and operated by an organization, giving it full authority over the app and its workings. Users interact with the app by downloading a copy and then sending and receiving data back and forth from the company's server. Smart contracts are a way to decentralize the controlling logic and payment functions of applications. Web3 DApps are about decentralizing all other aspects of an application: storage, messaging, naming, and so on (Antonopoulos & Wood, 2018). There are many advantages to creating a DApp that a typical centralized architecture cannot provide (Introduction to DApps, 2022):

- Zero down-time - Once the smart contract is deployed on the blockchain, the network as a whole will always be able to serve clients looking to interact with the contract. Unlike an application deployed on a centralized server, a DApp will have no downtime and will continue to be available as long as the platform is still operating. Malicious actors, therefore, cannot launch denial-of-service attacks targeted towards individual DApps.

- Privacy – Users don’t need to provide real-world identity to deploy or interact with it.
- Resistance to censorship - No single entity on the network, no service provider, not even the owner of the smart contract can block users from submitting transactions or reading data from the blockchain. Users will always be able to interact with a DApp without interference from any centralized control.
- Complete data integrity - With decentralized consensus mechanisms and cryptographic algorithms, data stored on the network cannot be manipulated by malicious actors.
- Incentivization – Through the use of tokens, users get rewarded for participating in the network. Tokens can be traded and used within the DApp ecosystem. They also help to align the incentives of all participants, ensuring that everyone is working towards the same goal.

DApps are however still in the early stages and face several issues:

- Scalability - As the number of users and transactions grows, the network can become congested and slow. Currently, the network can only process about 10-15 transactions per second; if transactions are being sent in faster than this, the pool of unconfirmed transactions can quickly balloon.
- User experience - The ability to develop a user-friendly interface is another concern, since most users of apps developed by traditional centralized institutions have an ease-of-use expectation. To attract a wider audience and reach mass adoption, DApps need to offer a

user experience that is on par with, or even better than, traditional centralized applications.

- Maintenance - Once deployed, a DApp will likely need ongoing changes for the purposes of making enhancements or to correct bugs or security risks. It’s hard for developers to make updates because the code and data published to the blockchain are harder to modify.
- Vulnerability to hacks, which can lead to significant losses for users.

DApps are often developed on the Ethereum platform for a variety of purposes including gaming, real estate and social media.

4.4 Ethereum

Ethereum's whitepaper was published in 2013, by Vitalik Buterin. In it, he explains the motivations for its development (Buterin, 2014):

“The intent of Ethereum is to merge together and improve upon the concepts of scripting, altcoins and on-chain meta-protocols, and allow developers to create arbitrary consensus-based applications that have the scalability, standardization, feature-completeness, ease of development and interoperability offered by these different paradigms all at the same time. Ethereum does this by building what is essentially the ultimate abstract foundational layer: a blockchain with a built-in fully fledged Turing-complete programming language allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions, simply by writing up the logic in a few lines of code. Smart contracts, cryptographic "boxes" that contain value and only unlock it if certain conditions are met, can also be built on top of our platform, with vastly more power than that offered by Bitcoin scripting because of the added powers of Turing-completeness, value-awareness, blockchain-awareness and state.”

Ethereum's creators aimed to overcome Bitcoin’s structure limitations beyond simple financial transactions and enable the creation of a wide range of decentralized applications, including smart contracts and decentralized autonomous organizations (DAOs). The state of Ethereum is composed by objects called “accounts". Each account contains four fields: a nonce, the ether balance, the contract code (if there is one) and the account's storage (empty by default). There are two types of accounts in Ethereum: external accounts, controlled by external actors and associated to private keys, and contract accounts, controlled by code stored in the blockchain that is activated each time it receives a message. The currency used by Ethereum is named ether and is used to pay transaction fees. Ethereum doesn't have a supply cap, this raises worries about the inflationary issues that the currency may present in the future. There have been several proposals for imposing a maximum amount in the future. Vitalik Buterin, himself, proposed that the total supply of ETH be capped at 120 million (Buterin, 2018). However, this was most probably just an April Fool’s joke. Ethereum has so far managed to keep inflation in check. The number of ETH newly issued has been steadily declining, which means that inflation isn’t a major concern. Even more so, thanks to the rollout of Ethereum 2.0, whose proof-of-stake is designed to lower the issuance, and of the EIP-1559, which introduced burning, Ethereum may soon become deflationary (Memoria, 2022). Bitcoin’s transactions are close to the concept of “messages” in Ethereum, but three important differences arise. An Ethereum message can be created by both types of account, can contain data and the recipient can return a response, in case of contract accounts. The concept of transaction in Ethereum involves the signed data package that stores a message sent from an external account. Transactions in Ethereum contain the recipient of the message, the sender's signature, the amount of ether and the data to transfer, and two variables called STARTGAS and GASPRICE. In Ethereum, the gas system is used to prevent the exponential blowup and infinite loops of code. It is necessary to block any attacks or coding errors in smart contracts that could overload the blockchain. The STARTGAS value, sets a cap on the number of computational steps of code execution. The GASPRICE value is the fee that the sender of a transaction must pay to the miner for each computational step. This fee incentivizes miners to prioritize transactions with a higher gas price and helps to limit the computational load on the Ethereum network, since users are forced to pay more for computationally intensive operations. The Ethereum state transition function, APPLY(S,TX) → S', takes the current state of the Ethereum blockchain (S) and a transition (TX) as inputs and returns the new state of the blockchain (S'). It follows the following steps:

1. Check if the transaction is well-formed, with a valid signature and a matching nonce.
2. Calculate the transaction fee as the product of the gas limit (STARTGAS) and the gas price (GASPRICE), and subtract such fee from the sender.
3. Initialize GAS = STARTGAS and take off a certain quantity of gas per byte to pay for the bytes in the transaction.
4. Transfer the value from the sender to the receiver. If the recipient account is a contract, the contract's code is executed until it either completes or runs out of gas.

5. If the value transfer or contract execution fails, cancel the transaction and revert all state changes except the payment of the fees to the miner.
6. Otherwise, refund the unused gas to the sender and pay the miner for the gas consumed.

One of the key features of Ethereum is its Turing-complete programming language, known as Solidity and used to create smart contracts. The Ethereum Virtual Machine (EVM) is the environment in which smart contracts are executed. A Virtual Machine (VM) is a software that allows to run an operating system on top of another operating system. For example, a virtual machine allows to use Windows on a MacOS device. The EVM is responsible for executing all operations performed by smart contracts, and it operates independently of the underlying computer hardware. The EVM ensures that every contract runs the same way on every (full) node, in an isolated and secure environment. The Ethereum blockchain, compared to the Bitcoin blockchain, has a more complex architecture. In fact, Ethereum blocks contain a copy of both the transaction list and the most recent state, apart from the block number and the difficulty. The block validation algorithm in Ethereum starts by verifying the existence and validity of the previous block, then checking the block's time stamp, number, difficulty, uncle root and gas limit. After that, it checks the proof of work on the block to make sure it was mined properly. The algorithm then takes the transactions in the block and applies them to the current state of the blockchain to update it to a new state. If all steps are successful, the updated state is compared to the value recorded in the block, and if they match, the block is considered valid and added to the blockchain.

5 Contract management

While the previous chapter introduced the concept of smart contracts, computerized transaction protocols that have the capability to execute the conditions of a contract automatically and securely through decentralized consensus, this chapter will discuss the broader topic of contract management. Any organization's performance depends heavily on its ability to handle contracts, which is crucial for ensuring that they are carried out effectively and efficiently. In fact, people, money and material resources have to be properly organized within a contract framework. The goal of contract management is to ensure that the organization's contracts are executed in a manner that satisfies all involved parties’ interests and supports the organization’s objectives. To discuss contract management topics, the construction industry was picked as a reference for several reasons. First of all, construction projects are often complex, large, and involve multiple stakeholders and large investments. Secondly, risks associated with delays and disputes, as well as financial and technical risk, are considerable when it comes to construction projects. By clearly defining duties and responsibilities, setting expectations, and establishing dispute resolution procedures, effective contract management can assist to reduce these risks. Additionally, construction projects are subject to a range of well-established legal requirements. To summarize, the construction industry is a good reference for contract management because it emphasizes the importance of efficient contract management in complex, high-risk, high-cost projects that call for cooperation between several stakeholders, to achieve successful project outcomes, minimize risk and safeguard the interests of all parties involved.

5.1 Contracts

A contract is a legally enforceable agreement between two or more parties, express or implied, source of rights and obligations, defined as law between the parties (Rainelli, 2021). An agreement is a meeting of the minds. This mental condition must be manifest through words, oral or written, or actions, by which a reasonable person, such as a member of a jury, can determine the intent of the parties. The fact that a contract is informal or made through gestures or a course of conduct does not make it any less legally binding. The presence of a written document is not essential to determine the meeting of the minds. In fact, people make contracts every day. Buying an item from a vending machine, for example, constitutes a contractual relationship. A first difference can be made between synallagmatic and aleatory contracts. In synallagmatic contracts the parties know and can assess since the beginning the reciprocal obligations, which are not related to an uncertain event. In an aleatory contract, the existence and extent of the obligation is contingent on the occurrence or non-occurrence of an uncertain event. An example of the latter can be a car insurance: the existence and extent of the obligation on the car insurance firm depends on the risk that the insured has an accident while driving. Contracts can be classified by type of formation, by type of performance or by enforceability. The first category includes express, implied-in-fact and implied in law contracts. An express contract is created when the terms of the agreement are explicitly stated through spoken or written words. An implied-in-fact contract is created when the parties conduct indicates that a mutual agreement has been reached. An implied-in-law or quasi contract is created by operation of law, when a party has rendered a benefit to another which requires a fair compensation, even if there was no meeting of the minds, in order to avoid unjust enrichment. For example, a physician that gives assistance to an unconscious patient is entitled to the fair rate of pay, defined by market price, from him, even if he did not consciously ask for the provision of such services. The second category includes bilateral versus unilateral contracts and executed vs executory contracts. A bilateral contract is based on an exchange of promises or obligations. In a unilateral contract only one party defines an obligation which is contingent on the performance of a specific act. The parties in a bilateral contract are both obligated to perform, while in a unilateral contract, only one party is obligated to perform, based on the performance of a specific act from the other party. With executed contracts, performance is executed only once, while with executory contracts, execution is continuous over time. The third category includes valid, unenforceable, void and voidable contracts. A valid contract meets all legal requirements and can be enforced by either party. An unenforceable contract misses legal requirements and cannot be enforced. A void contract has no validity, is not legally binding and cannot be enforced by either party. A voidable contract can be rendered void by one of the parties, who has the option to withdraw from it.

There are four essential elements in a valid contract: capacity of the parties, mutual agreement or meeting of the minds, consideration and legality of the subject matter. Capacity of the parties refers to the legal ability of the parties to enter into a contract. For example, minors, mentally incapacitated and intoxicated individuals may lack the capacity to enter into a contract. The mutual agreement depends on the presence of two elements, a binding offer and the acceptance. The offer can be presented by spoken or written word, or by actions, through any medium. The following requirements are needed: it must indicate a clear intent to make a contract, it must be sufficiently definite and it must be communicated to the other party. The acceptance of the offer seals the contact. However, it must meet certain standards. It must be clear and unqualified, that is, it should not ask for different conditions or modifications; that would be the case of a counteroffer. The acceptance should also abide by any manner required by the offer (the latter may request acceptance through a specific medium or by a certain time, for example). Consideration is something (a promise or an action) a party provides in exchange for something from the other party. Consideration is essential to a contract because it establishes a quid pro quo ("something for something"), or mutual exchange of promises, between the parties. It shows that both parties are entering into the agreement voluntarily and with the intention of fulfilling their obligations under the contract. A contract cannot be one-sided; it wouldn't be enforceable without a promise on each side. Usually, a court is not concerned with the adequacy of consideration. Consideration may be absent in case of illusory promises, obligations of something that a party is already bound to do, moral obligations and past consideration (as a reward for a benefit received in the past, for which something in return was not expected to be obtained). The final requirement for a contract to be valid is the legality of the subject matter. In fact, a contract is not deemed valid if the contents of the contract do not abide by the prescriptions of the law.

Even though oral words and actions represent evidence that the meeting of the minds occurred, and thus a contract was established, certain agreements must be mandatorily incorporated in a written document, called memorandum. This was established by the British Parliament in 1677 with the Statute of Frauds, to prevent perpetration of frauds arising out of purely oral agreements (Johnston, 1996). The list of agreements is jurisdiction specific and may regard contracts over real estate properties, with value in excess of a certain amount or duration above a specific time, and so on. The memorandum in writing must meet some requirements: it must include the essential conditions of the contract (it's not needed that it contains all terms of the transaction), it must identify the parties of the agreement, who also must sign it. Essential conditions of a contract usually are a description of the consideration and of the goods to be sold or of the service to be provided. Regardless of the Statute of Frauds, a written contract brings significant advantages. The writing process helps to identify matters that should be covered, helps to clarify contractual terms thus reducing the risk of potential future disputes, and serves as factual evidence of the existence of an agreement and of its terms, helping in their future recollection (*verba volant scripta manent*). The parol evidence rule affirms that when two parties have made a contract and have expressed it in a writing, evidence, whether parol or otherwise, of prior or contemporaneous understandings and negotiations cannot be used to invalidate the contract (Corbin, 1944). All parties, thus, can rely on what is written on the contract, since it is presumed that its content constitutes the real arrangement between the parties. Evidence clarifying, explaining, supplementing, completing or elaborating upon the agreement may be introduced as exception to the parol evidence rule. Rights and obligations stated within a contract regard only the legal entities parties to that contract (privity doctrine). There are some exceptions to this principle. Rights declared in the contract, unless prohibited by law or the contract itself, can be assigned to a third party. Routine duties can be delegated, but duties requiring personal skill or reliability generally cannot, since the performing party was most certainly chosen due to its skill, abilities and reputation. If possible, an original party to the contract may decide to assign the contract to a third party (not just some rights or obligations). Assignment of contract may be with release, if the the original party is released from its obbligations, or without release, if the original party remains responsible and liable for potential damages or breaches of contract. If the previous circumstances lead to the old contract being replaced by a new one, novation occurs. The new contract replaces the old contract and assumes all of its obligations and rights. A final exception to the privity doctrine are third-party beneficiaries, who receive some benefits from the performance of the contract but have no obligations, since they are not actual parties to the contract.

5.1.1 Discharge

Discharge of a contract refers to the termination or completion of a contractual relationship between the parties involved in the agreement. Lack of mutual consent (including mistake), lack of consideration, lack of delivery in case of deeds, illegality, incapacity of parties, fraud, and duress do not constitute discharge but all these affect the formation the contract, either preventing the existence of any primary obligation; or making the contract voidable and the obligation imperfect, and giving to the defendant alone the option of avoiding or enforcing the obligation (Corbin A. L., 1913). Discharge can occur in several ways, including:

- Performance: the contract is discharged when the substantial performance of the contractual obligations has been fulfilled by all contractual parties. Substantial performance depends upon the main provisions of the contract and does not imply performance to the very last detail.
- Breach: If one party (the breaching party) fails in a material way to fulfill its obligations under the contract, the other party (the non-breaching party) can choose to terminate the contract, which gets discharged by breach. The non-breaching party can ask for

damages or other remedies. Even if the contract is discharged, the nonbreaching party is responsible for the value received and the breaching party can recover in quasi contract for the *quantum meruit* for his limited performance. The court will generally subtract the value of what was received from the damage granted.

- Anticipatory breach: when there are clear, unequivocal signs that a party won't be able to fulfill its contractual obligations (for example if it emits a statement of nonperformance or if it went bankrupt), the other party does not have to sit idly by and wait for the date of performance to see that the other party won't perform, before declaring the contract breached. The nonbreaching party might be entitled to potential damage claims.
- Agreement of the parties: the parties can agree to discharge the contract by mutual consent (mutual rescission); to substitute a new performance in place of the previous obligation (accord and satisfaction), whose satisfaction discharges the contract; to the payment of a smaller sum than what was originally contracted, with a conspicuous statement that payment is in "full", which discharges the debtor from the remaining debt and, so, the contract.
- Release and waiver: the two concepts are quite similar, a party can release the other from its remaining obligations or waive its rights under the contract (usually when the latter is unable to perform and agrees with the payment of a consideration in order to avoid being sued and the potential future payment of damages).
- Operation of law: a contract may be discharged by operation of law because of subsequent illegality, impossibility, bankruptcy or statute of limitations. The principle of subsequent illegality applies to contracts that are legal when made but become illegal after the

passage of prohibition laws. Impossibility refers to the occurrence of events that make the performance of the contract impossible, such as death of the performer and acts of God/force majeure. Force majeure is a common clause in contracts which essentially frees both parties from liability or obligation when an extraordinary event or circumstance, beyond the control of the parties, prevents one or both parties from fulfilling (even partially) the obligations under the contract. These events typically include natural disasters, such as hurricanes, earthquakes, typhoons, volcanic activity and tsunamis, as well as other events like war, rebellion, terrorism, pandemics, discovery of archaeological relics or historical artifacts, contamination by radioactivity or hazardous/toxic substances, and so on. In cases where performance is vulnerable to natural occurrences it is highly desirable to include force majeure clauses in the contract. The principle of strict impossibility and the rule of commercial impossibility are two different approaches that courts follow when interpreting force majeure clauses in contracts. The principle of strict impossibility affirms that a party is released from performance under a contract only if doing so is truly impossible, as opposed to being merely more difficult or less convenient. This approach is often seen as more rigid and places a higher bar for excusing performance. The rule of commercial impossibility holds that a party is excused from performance under a contract if the event in question causes excessive and unreasonable costs to fulfill contractual obligations. It is important to note that the approach that a court takes can depend on the jurisdiction and the specific language of the contract. Some courts may also apply a hybrid approach. A contract might be discharged in case of bankruptcy. When a company files for bankruptcy, a court-appointed official, known as a trustee, is nominated to manage the assets of bankrupt company and needs to ensure that contracts entered into by the debtor are either performed or discharged as part of the bankruptcy process. He will pick mostly those that are not beneficial for the running of the firm and the maximization of value of the company with an eye to creditors' returns. Sometimes a contract may be discharged by the statute of limitations (Callahan, 1955), a legal time over which a party may no longer enforce its rights under the contract or seek damages for breach of contract. Periods of limitation vary from state to state but generally run from 3 to 6 years.

5.1.2 Damages

The term damage is used to refer both to the damage suffered by the non-breaching party and to the indemnification due by the breaching party to the non-breaching party to recover for the loss or injury caused by the breach. Necessary conditions for which a plaintiff may be granted damages are: proof of the existence of a valid contract, factual evidence that the breach was caused by the defendant, and proof that a loss was suffered by the claimant as consequence of the breach. Four types of damages can be distinguished: compensatory, consequential, liquidated and punitive damages. Compensatory damages are thought to compensate the non-breaching party for its actual losses suffered as a result of the breach. The goal of compensatory damages is to place, so far as money can do it, the party sustaining a loss by reason of a breach of contract, “in the same situation, with regard to damages as if the contract had been performed” (Robinson v Harman, 1848). An example can be given by a contractor that has not done certain work within the period agreed upon. He must pay a sum of money sufficient to make good both the loss caused by his delay, and the loss of benefit which would have been realized by his prompt completion of the work (Demogue, 1917). The following conditions operate to limit compensatory damages: damages must be proved to a reasonable certainty, the breaching party needs to have the possibility to foresee the potential damages at the time of the contract or at the time of the breach, plaintiff must use every reasonable effort to mitigate the damages, that is “a plaintiff may not let the meter run” (Slovenko, 1998). Consequential damages cover also the indirect consequences of the breach, such as reputational damages, always if the principles of foreseeability and certainty are met. Liquidated damages, commonly referred to as penalty clauses, are provisions included in the contract as damage clauses constituting part of the meeting of the minds. These provisions embed a pre-determined amount that will be payable in the event of a breach of contract. In order to be enforced by the court, these clauses need to reflect reasonable efforts by the parties to calculate a fair estimate of the actual damages that would be suffered in case of a breach of contract. A court may refuse to execute the clause and limit the parties to pursuing actual damages if it determines that the amount specified as liquidated damages is unreasonably high or constitutes a penalty. Punitive or exemplary damages get awarded to a plaintiff in a lawsuit in addition to compensatory damages, with the purpose of punishing the defendant for its outrageous, malicious and oppressive conduct, such as in cases of fraud, gross negligence, or intentional wrongdoing, and deterring similar conduct in the future, from himself or others. Punitive damages are only available in certain jurisdictions, their amount can vary widely and is often left to the discretion of the jury, which may not only consider the maliciousness of the conduct but also the wealth of the defendant.

5.2 Contract organization

A contract distributes rights and obligations between the entities entering in an agreement for the provision of goods or services: the owner, one or more contractors, and other professional entities providing finance, design, construction, operation and maintenance services. Having a clear and well-written contract is essential in the provision of goods or services, as it helps to minimize misunderstandings and disputes, and safeguards the interests of all parties involved. Contractors can either be responsible only for construction based on the provided design specifications, or act as design-build (or EPC, engineering-procurement-construction) firms and supply both design and construction. Subcontracting part of the work, unless clearly prohibited by the contractual terms, is generally possible. Complex projects require the assistance of varied professional entities able to provide financial, legal, design, assistance, quality assurance and coordination services. Contracting systems help allocate risks, responsibilities, duties and rights between all stakeholders. They are made up of three main components: a delivery system, a payment scheme and an award method. Private owners have a lot of flexibility and options when it comes to deciding how the work is to be carried out, while public owners have to abide by certain statutory and administrative requirements. The delivery system allocates the scope of work of a project, usually fragmented into design, financing and construction, to each contractual party. The payment terms define the agreement between the owner and the contractor regarding payment for the work or project. The award method establishes selection criteria for assigning the contract.

5.2.1 Delivery systems

As previously stated, the scope of work of a project can be fragmented into design, financing and construction. A contractor can be responsible for one, two or all three components. There are four main project delivery methods: construction services only (traditional design-bid-build), design-build, turn-key and build-operate-transfer arrangements. Design Bid Build contracts (DBB) provide that the general contractor's only obligation to the owner is to complete construction. This arrangement fully excludes the contractor from the design process and prevents them from contributing to the design. Design may be pursued by the owner with in-house capability or delegated to a private architect-engineer firm acting as an independent contractor, which provides production of a complete set of drawings and specifications, and coordination and monitoring of the operations during construction as an agent of the owner. The owner undertakes the financing and budget requirement. When all the documents have been approved, the designer announces the project to contractors by means of an advertisement, invitation to bid or notice to bidders, makes bid and contract documents available to contractors and respond to written requests for information (RFIs) from the contractors. The contractors then prepare their estimates based on the information in the contract and bid documents. The owner ultimately chooses a contractor based on the bid proposals received. After the contract has been signed between the owner and the contractor, the contractor will proceed to perform the contract requirements. The owner, architect-engineer, and contractor perform their defined obligations semi-independently of the others under this contractual agreement. The Design-Build, also known as Engineering Procurement and Construction (EPC), contract is rapidly gaining popularity and is set to become the most common method of delivering design and construction services. The owner enters into a single contract with a professional entity that has the responsibility of providing both design and construction services. This could be a joint venture between a construction firm and an architecture or engineering firm, or a construction firm with in-house design capability. The design-build method advantages compared to the traditional model consist in that it allows fast tracking (it is possible to overlap design and construction to expedite the project) and a single point of accountability which reduces the need of design changes and the chance of additional costs for the owner. Its main downside is that the monitoring and control task goes to the owner, who may however hire a Construction Manager (CM) to do that (De Marco, 2011). A Turnkey contract is quite similar to the DB/EPC model, with the only difference that the contractor also short-term finances the project during the construction period. Actually, in many cases in the construction industry the terms "design-build" and "turnkey" are used interchangeably. However, some professionals use the term "turnkey" to emphasize the idea that the design-build firm provides a complete and finished product that is ready to be used. Finally, the Build-Operate-Transfer (BOT) delivery system is a model in which the contractor is responsible for financing, designing, building and operating large-scale, long-term projects that require significant investment. This is usually done through a special purpose vehicle company (SPV). The owner gives concession rights for a specified period of time, usually not less than 20-30 years, after which it regains the right for its own O&M and usage, typically with no extra cost. The contractor intends to recover and profit from the original investment through revenue obtained from the service provided by the facility during the concession period. The BOT model is mainly used in Public-private partnerships (PPPs), which involve collaboration between the public and private sectors to deliver infrastructure projects. PPPs are used to leverage private sector expertise and investment to deliver infrastructure projects that would otherwise be too costly for the government to finance and manage on its own (Satish & Shah, 2009). The correct choice of the delivery system depends on how it is deemed best to distribute the project risk between owner and contractor and on who owns the project know-how. In a traditional DBB system the owner accepts to handle most of the risk, which is, on the contrary, mostly allocated to the contractor in a DB, turnkey or BOT contract.

5.2.2 Payment terms

Payment terms specify the amount of money that is expected to be paid to the general contractor by the client based on the agreed-upon price. The choice of the payment scheme depends on risk sharing motivations between client and contractor (Ward & Chapman, 1994). The main payment schemes follow. In a time and material (T&M) or cost plus fixed percentage of cost contract, the contractor is reimbursed for all direct costs incurred during project execution such as labor, material, and equipment. Additionally, the contractor is also paid a fee, typically a percentage, to include overhead costs and a fair profit. Most of the risk is on the owner's shoulders, since he cannot be certain about the actual contract value since the beginning and the contractor has no incentive to save on costs. In the unit-price payment scheme, after the owner and the A/E firm have provided the list of work activities and the estimated quantities, the contractor will calculate the overall cost of a work activity, for materials, labor, equipment, general overhead and markup, and divide it by the estimated quantities, so to obtain the unit price. The unit prices in the contractor's proposal will become the contract prices and as the work is performed, the actual quantities for each activity will be recorded by both the contractor and the A/E firm. The final contract price will depend on the actual quantity of each work item installed or performed at the contracted unit price, without regard to whether this actual quantity is greater than or less than the quantity originally estimated by the engineers during design (Clough, Sears, Segner, & Rounds, 2015). The owner bears the risk on quantities, while the contractor on the increase of unit prices. To reduce risk, the owner should carefully supervise work, thanks to the help of a CM. Refined payment schemes can be discussed to introduce incentives to the contractor to reduce time and costs. In cost-plus-fixed-fee arrangements the contractor is paid the actual cost plus a fee as a fixed amount of money.

$$PRICE = AC + FIXED\ FEE$$

The contractor is incentivized to reduce time and costs since a longer duration and higher costs reduce profitability and relative return on the project. A stronger incentive to reduce costs is given by the target cost plus incentive fixed fee contract, where the contract price is given by the actual cost plus a share on cost savings or extra costs (target cost minus actual cost) plus a fixed fee. If contractor saves on costs, he will get paid a higher price. Otherwise, he will see his profits reduced.

$$PRICE = AC + (\Delta C * \%share) + FIXED\ FEE$$

The Guaranteed Maximum Price (GMP) contract is a variation of cost-plus-fixed-fee, in that it also includes a ceiling on price. If the contractor exceeds such cap, he will have to bear the extra costs. GMP contracts may include a shared-savings provision to encourage further cost savings.

$$PRICE = \begin{cases} AC + FIXED\ FEE\ \ if\ GMP > AC + FIXED\ FEE \\ GMP\ if\ GMP < AC + FIXED\ FEE \end{cases}$$

In lump-sum or fixed-price contracts, contract price is determined by the lowest proposal amount received in the competitive bidding stage. Under this payment system owners minimize risk because they know the cost of the project before it begins. However, contractors have incentive to save on costs at the expense of quality.

5.2.3 Contract award

The method of contract award is important in determining the success of a project and ensuring that the project is completed on time, within budget and to the satisfaction of the client. There are three basic methods by which a contract may be awarded to a contractor: competitive bidding, negotiation, and competitive sealed proposals. Competitive Bidding is a process where contractors submit bids in response to a Request for Proposal (RFP) issued by the client. It is commonly used in public projects, where it is required by law. The process involves the owner and architect-engineer preparing complete drawings, specifications, and contract documents that fully describe the project. Advertising through newspapers and web sites is carried out. Contractors who obtain these documents will decide whether they are interested in bidding on the project. If they are, they will prepare a detailed estimate and proposal, which they will submit to the architect-engineer and owner on bid day. The client evaluates the bids based on a set of criteria, such as price, quality, and delivery time, and selects the best bid. Competitive bidding is a transparent process and is used in most public procurement contracts. This method is suitable for well-defined projects, where the requirements are clear and can be easily quantified. This method of contractor selection has been in use for many years and grants the owner the pursue of a project of specified quality at the lowest possible price. This method also helps to ensure that the project is completed on time, as the winning contractor is bound by the terms and conditions of the contract. While it allows for taking advantage of a good price and a transparent process, competitive pressures arising from competitive bidding, such as time pressure resulting in an insufficient consideration of design before pricing and downwards cost pressure due to fear of being underbid by the competition, can dwarf any cost savings secured by the owner due to elevated rates of change orders, cutting corners and dispute-oriented relationships. Other than open bidding, where all contractors are encouraged to submit a proposal, closed bidding, where access is restricted to a chosen set of bidders can also be an option. The restricted procedure should be used for procurement exercises with many potential bidders. It is a two-stage process. The first stage is a selection process, where the bidders' capability, capacity and experience to perform the contract are assessed. During this stage, the number of bidders gets reduced. The second stage involves the issue of the Invitation to Tender only to the shortlisted bidders. Bids are assessed to determine the most advantageous tender, which will get awarded the contract. In this regard, a largely used approach is the dynamic purchasing system, DPS. Two stages can be distinguished in a DPS, an application stage and an invitation to tender stage. In the first, contractors are asked to apply for the DPS, certifying that they meet the minimum requested requirements to apply. The owner will evaluate. If access is not granted, it is generally possible to reapply in the future. If the evaluation is passed, that contractor makes part of the DPS, together with other qualifying contractors. It is called dynamic because the application procedure is always open and new contractors may enter at any time. The second stage is limited only to suppliers on the DPS, who are allowed bid. This assures that the number of tenders received is restricted and that only the suppliers meeting the minimum requirements are bidding. Competitive bidding is usually associated with the integrated design-bid-build delivery system using lump sum or unit price payment terms.

Negotiation is a process where the client and the contractor negotiate the terms and conditions of the contract. This method is used either in very simple projects, where saving on time is more important than saving on costs, and in complex projects, where the design requirements are not well defined and need to be negotiated between the client and a well-experienced contractor. It’s possible to distinguish between competitive negotiation and negotiated contracting. Under the competitive negotiation process, which is a hybrid between competitive bidding and negotiation, prequalified contractors submit priced proposals. The agency then advises each contractor on how to improve its proposal in terms of both design and cost, and new proposals are submitted. Price and technical factors are the base for the selection process. Competitive procedure with negotiation can also be used where all of the submissions received for an open or restricted procedure that you have conducted are classed as either irregular or unacceptable. In negotiated contracting, the owner and the prime contractor negotiate the terms and provisions of the contract. Negotiated contracts are commonly employed on private projects. Negotiations are typically more time-consuming and less transparent than competitive bidding, but they offer the advantage of flexibility (agreements can include any provisions mutually agreeable to both parties and best suited to the specific work involved) and the opportunity for the client and the contractor to work together to find the best solution for the project. Handpicking a contractor based on reputation and qualifications, or on past experience with that contractor is common practice. Negotiated contracts are usually associated with design-build and turnkey delivery systems using cost-plus-fee type payment schemes.

Competitive Sealed Proposals is a process where contractors submit sealed proposals in response to a Request for Proposal (RFP) issued by the client.

The competitive sealed proposal will include more information about the contractor than just a description of the professional services he will offer, including his background and experience, a list of the projects he has completed, a history of the owners for whom he has worked and their contact details, the credentials of the main personnel that will be engaged in the project (often including the project manager's CV). The price that the contractor submits in his proposal may be a lump sum, or a series of unit prices, or it may be one of the variations of cost-plus. The client evaluates the proposals and selects the best proposal, based on a set of criteria. Sometimes the owner picks a few proposals and then invites the finalists to prepare verbal presentations and to answer questions or clarify issues. This method is similar to competitive bidding, but the proposals are sealed and not open for public inspection. The advantage of competitive sealed proposals is that it provides a higher level of confidentiality, as the contents of the proposals are not publicly disclosed, making it suitable for projects where the client wants to keep the details of the project confidential, such as government contracts. The award decision should be communicated by the means of standstill letters. During the mandatory standstill period, the draft contract should be finalized ready for signature upon expiry of the standstill period (UK Government commercial function, 2021).

5.3 Contract administration

Before awarding a contract the owner must engage in a bidding process, either as a competition or a negotiation. Before this process, the owner needs to prepare the contract documents that will later be managed during the procurement or construction and close-out stages. The information provided in this subchapter will mostly refer to public award procedures. When the owner has approved the design, the project needs to be announced to general contractors and bid and contract documents need to be made available to contractors, who are invited to prepare and submit proposals for negotiation or competitive bidding. In general, there are two ways to announce the project to contractors: contract notice and invitation to bid. A government organization or business entity, to inform all potential suppliers or vendors about an upcoming procurement opportunity, will issue a public announcement inside a tender portal, called contract notice. The contract notice provides information to potential bidders about the procurement opportunity, including details about the contracting authority (name, postal address, internet address, main activity, etc.) and if applicable, joint procurement details, i.e. the details of the other buying organizations involved, of where the procurement documents are available from, the type of contract (goods or services), a description of the type of goods or services being procured, the estimated value of the procurement, the place of performance (the geographical place(s) where the contract is to be performed), the scope of work, the award criteria, the duration of the contract, framework agreement or dynamic purchasing system, the type of procedure (open procedure, restricted procedure, etc.), the deadline for submitting bids or proposals, any specific (economic, financial, technical and professional) requirements or conditions that bidders must meet in order to be considered for the contract, and review procedures. The contract notice will include enough information to allow to decide whether to bid for the contract opportunity or not, as it must contain the minimum and specific requirements for the procurement exercise (Supplier Journey, 2023). It is required by law for public procurement opportunities. The notice fosters transparency and fairness in the procurement process by giving all interested bidders an equal opportunity to access information and prepare and submit bids or proposals in a timely and competitive manner. It’s very important that the notice includes a link to the procurement documents, so that a contractor is able to retrieve and download them. Some procurement opportunities still use old-fashioned models involving the duty to physically reach the contracting authority's offices to retrieve drawings, specifications, bid and contract documents copies upon payment of a refundable deposit, as a guarantee for the safe return of the documents within a predetermined number of days after bids are opened. The Invitation for Bid (IFB) is used by the contracting authority to notify preselected contractors about a project and invite them to submit proposals. Only the invited contractors are allowed to submit a bid. The IFB includes the same kind of basic information about the project as described above for contract notices. It is a legal practice for private companies but prohibited by statutory stipulations that regulate contract formation in public construction or procurement exercises. An alternative to IFB is the Request for Proposal (RFP) process. RFPs are used to solicit proposals from potential vendors for a particular project solution not considering price alone. The RFP details what the entity is seeking, lacking, however, a clear description of the project scope and requirements, and provides evaluation criteria for evaluating the proposals it receives. The main difference between the two is that an RFP typically involves a more complex project or service that requires a proposal outlining the supplier's approach to meeting the requirements, while an IFB is a more straightforward request for a price quote. Whereas an IFB will evaluate proposals largely based on price, RFPs will consider price as well as details of the bidding organization's operational plan, staff experience and education, timeline estimates and more. The review process for an RFP is typically longer than those of IFBs (Office of Contracting and Procurement, 2023). For DB and turnkey delivery systems, the process to answer to the RFP can be compared to a project by itself, due to the complex and time-consuming activity aimed at producing design, financial and contractual documents which requires multi-disciplinary competencies. After reviewing the basic information about the project included in the contract notice or Invitation for Bid, a prospective tenderer usually needs to submit an Expression of Interest (EoI[5]), sometimes referred to as 'request to participate', to the contracting authority in order to get access to the contract documents. The contracting authority usually has to review a large number of applications, which needs a lot of time, effort and resources. The comparison process can be eased by means of pre-qualification questionnaires (PQQs), which ensure that all submissions are made in the same format and provide the same information. On the flip side, this gives potential tenderers additional work, since they have to reformat their documentation to match the PQQ. Access will be granted only to contractors meeting the minimum requirements or conditions established in the contract notice. The contracting authority assembles the bid documents, the contract documents, drawings and technical specifications into a book or binding (or a zip file), which is referred to as the project manual.

5.3.1 Bid documents

On a competitive bid project, the contracting authority will typically produce and publish a set of documents referred to as the bid documents

[5] Not to be confused with a Request for Expression of Interest (REoI), a document that is issued by a buyer at the early stages of the procurement process, before a formal solicitation such as a Request for Proposal (RFP) or Invitation to Bid (ITB) is issued, to solicit interest from potential suppliers or vendors in a specific project or opportunity. The purpose of an REoI is to identify potential suppliers or vendors who are interested, qualified, equipped, and experienced enough to deliver the products or services needed for a project. Potential bidders may be requested basic information in the REoI, such as their credentials, expertise, and interest in the project. They are usually provided details about the procurement opportunity, including the scope of work, the estimated value, and the anticipated timeline. Responses to an REoI can be used by the buyer to create a shortlist of qualified bidders who will later be invited to participate in the formal bidding process.

for the project. These documents usually include the contract notice or the invitation to bid, along with the instructions to tenderers and the proposal form. Instructions to tenderers include details concerning the technicalities of the bidding process. These instructions include the requirements pertaining to the form and content of the bid established by the contracting authority, where and when it must be delivered, bid security required, and information concerning late bids and bids submitted by mail or e-mail. The owner reserves the right to reject any or all bids and to postpone the date of bid opening. Many public agencies have developed their own set of standard forms of instructions to tenderers (Clough, Sears, Segner, & Rounds, 2015). In case of open procedure, a prepared proposal form is usually included with the bid documents and must be used by the contractor to present its bid. Failure to do so will lead to the contractor's disqualification. Thanks to the prepared proposal form, all presented offers can be assessed by the contracting authority on the same basis, making the comparison of the figures easier. In some instances, contractors would like to input additional information concerning their bid that the proposal form fails to include. In general, this is not permissible on public bidding, and doing so will make the bid subject to rejection. In private scenarios, owners may be more inclined towards proposal qualifications, however, provisions in the instructions to tenderers may be present to indicate that any qualification of the contractor's proposal may be cause for rejection of the proposal. In case of restricted procedure, the form of the proposal submitted is frequently left up to the individual bidder, who will try to structure his bid with the aim to make a staggering impression on the review body, since the contract is not necessarily awarded to the lowest bidder.

5.3.2 Contract documents

The contract documents for a project are the set of documents that include: the general conditions, the special conditions, technical specifications or terms of reference (ToRs), drawings, the agreement and supplementary documentation. General conditions, also known as general provisions, are standard terms and conditions that define the rights and responsibilities of the parties involved in a contract, cover the nontechnical requirements and apply to the project as a whole. General conditions of the contract have been formulated and developed over the years by special committees representing various industry and professional groups, government agencies, and professional bodies. They are designed to protect both the contracting parties and to ensure that the contract is performed in accordance with legal and ethical standards. In addition to outlining the rights and responsibilities of the parties to the contract, the general conditions present requirements governing their business and legal relationship, and include guidelines to be used in administering the contract. They also define duties and responsibilities of other parties affected by the contract, such as subcontractors. Other nontechnical matters describe how the contract will be performed, the obligations of the parties, the scope of work, control of work, legal relations and responsibility to the public, payment terms, dispute resolution, warranties and guarantees, termination clauses, and other administrative provisions. (Goldbloom, 1989). The Special Conditions or Special Provisions amplify and supplement, if necessary, the General Conditions governing the contract and add specific project-related issues. Unless the Special Conditions provide otherwise, those General Conditions remain fully applicable. Drawings and technical specifications usually go together to form the design documents. The drawings depict, in graphical form, the elements of the design. Drawings may be produced on paper or electronically in the form of computer-aided design (CAD) or building information model (BIM) files. The technical specifications (for procurement of goods), scope of work (for services and infrastructure projects) or Terms of Reference (for consulting services) is the document that provides the detailed description of the deliverables to the supplier, contractor or consultant. The following guidelines are considered helpful in writing the TS/SOW/TOR (Arcadio & Cuenco, 2016):

- If the procurement is complex, highly technical, or high value, and contracting authority does not have sufficient technical proficiency, it would be prudent to engage the services of technical experts who will serve as consultants in the preparation of the TS/SOW/TOR.
- Unless the procurement involves goods or services that are covered by intellectual property rights, are only offered by one source, or there are no suitable substitute products or services, the description should be general and flexible rather than product-specific or seeming tailored for a particular brand, product, contractor, or consultant. If the procurement involves products or services that have intellectual property rights attached or are exclusively offered by a single source without any suitable substitute, then the PMO or end-user unit must be able to satisfactorily justify the need for such a product or service.
- The description should be clear and unambiguous, to avoid confusion and to facilitate the evaluation process. It is noted that the TS/SOW/TOR is also the basis of the evaluation.
- Considering that planning is done sometime before actual procurement and even farther from the date of actual delivery or project implementation, changes in technology or changes in the

concerned industry or field of expertise should already be taken into account in writing the TS/SOW/TOR. This will allow the procuring entity to procure the latest products and/or services available in the market.

The manner in which the specs describe the functional requirements with regard to quality standards for materials and products to be used is defined as specification language. It can either be “open” or “closed”. Open specification language is used in both private and public projects, while closed specifications, also known as proprietary specifications, are often outlawed for use on public projects and are thus used mostly on private initiatives. Open specifications for materials and products describe their characteristics in generic terms, usually in terms of compliance with recognized industry standards, which will apply to several brand names or manufacturers. Closed specifications define quality requirements by listing a specific brand-identified product, sometimes including model name and/or model number. They may be so detailed and precise that only a product made by a preselected supplier can satisfy them. For these reasons, in order to avoid favouritism and promote fair trade and open competition, public bodies only allow the use of open specifications. The use of open specifications can, however, lead to later disputes between owner and contractor due to different views regarding the conformity of the provided product or service to the contract specifications. An example is given by the “or-equal” provision, which allows the use of alternative materials or products similar or equal to the product specified as standard of quality. Views may diverge between owner and contractor with regards to what can be deemed as a valid substitute, since it is unusual for comparable products of different manufacturers to be identical in every respect. Appearance, size, configuration, or design usually differ. It is good practice for the contractor to submit a proposed substitute to the owner together with samples, descriptive and technical data, and test reports as means to demonstrate equality. The risk is, however, that the contracting authority may reject the proposal as irregular. It is, therefore, very frequent to run into disputes leading to a court ruling. In some segments of the construction industry, to avoid time and effort in the preparation of project specifications, standard specifications are being implemented. The project manual must include just a reference to them. They can also be augmented with modifications in order to make them conform to the unique aspects of a particular project.

Depending on the delivery system, design can be provided by a contracting authority to the contractor as basic design or detailed design. Basic design, also known as conceptual design or preliminary design, bridges the gap between design conception and detailed design. It is a high-level overview of the project which involves drafting functional descriptions, drawings and block diagrams, defining high-level lists of components and modules, and comparing and selecting alternative solutions (Cantamessa & Montagna, 2016). It provides a general understanding of the project, focuses on creating the general framework to build the project on and serves as a starting point for the detailed design. Basic design is typically prepared by the contracting authority or a consultant in case of design-build, turn-key and build-operate-transfer arrangements. Detailed design is the process of creating detailed drawings and specifications for every aspect of the project. This includes the structural, mechanical, electrical, and plumbing components in the case of a construction project. Detailed design is typically prepared by the contractor in case of design-build, turn-key and build-operate-transfer arrangements or by the contracting authority (or a firm to which the contracting authority assigns the task to) in case of design-bid-build contracts.

The agreement is a draft contract form summarizing the main elements of the contract, namely scope, price, baseline schedule. When signed by the owner and the contractor, both parties bind themselves to all of the elements of content of all the contract documents. Additionally, the agreement will typically include details about the project duration, the place of acceptance, the performance guarantee, the price that the contracting authority agrees to pay the contractor in consideration of the execution of the contract and a reference to the addeda. Addenda are amendments to the bid documents or to the contract documents issued by the contracting authority during the bidding period, which goes from the announcement of the contract notice or invitation to bid to the deadline for bids submission. All contractors that have requested contract and bid documents will be timely notified when addenda are issued.

5.3.3 The contract in practice

Following is a contract sample for a project I have analyzed to potentially submit a tender when I was working in the Business Development department of AquaBioTech Group, an international consulting company strategically located in the center of the Mediterranean on the island of Malta, although operating globally with clients and projects in over fifty-five countries. AquaBioTech Group undertakes a variety of aquaculture, fisheries and aquatic environmental projects through its regional offices and selected partners throughout the world. The vast majority of the company’s work is related to the marine or aquatic environment, encompassing aquaculture developments, market research/intelligence, through to project feasibility assessments, finance acquisition, project management, technology sourcing and technical support and training. The project title is “Marine ecological surveys for Ħofra ż-Żgħira, Delimara”. The tender was issued by Enemalta plc, a limited liability company mainly responsible for the dispatch and distribution of electricity to the Maltese islands with a limited capability for electricity generation to be used in case of emergency situations. The company’s plants are located at Delimara. The operation of the plants is subject to the issue of an Integrated Pollution Prevention and Control (IPPC) permit by ERA. ERA (Environment and Resources Authority), the regulatory agency responsible for safeguarding the natural environment in Malta, requests IPPC permit holders to carry out annual marine ecological surveys using the methodology agreed with the authority. The monitoring for each assessment year shall be carried out during the summer months, preferably the same month, to assess the impact of the cooling water outfall on the habitat types and species including Pinna nobilis, and Posidonia oceanica beds and Cymodocea nodosa meadows, in the surrounding waters. The area of influence under study is the Ħofra ż-Żgħira embayment. The monitoring extends to the area 300m beyond the mouth of il-Ħofra ż-Żgħira bay. If any decline in the conservation status of the habitat types and species in the area is detected, a report of the current status must be sent to the authority, followed up with proposals for mitigation measures, which shall be reviewed and agreed to by the authority prior to their implementation. Methodologies employed by the marine ecological surveys should reflect the criteria and indicators used to assess habitat status and condition by relevant EU policy including the EU Habitats Directive and the EU Water Framework Directive. With reference to records of the nonindigenous species Penicillus capitatus and in view of the disturbance caused by the discharges which could promote the spread of this species, the marine ecological survey shall also monitor the extent of spread and abundance of this species, and its interaction with local species or communities, taking into consideration links with the relevant discharges. The survey shall also monitor for Cladocora caespitosa which was observed at il-Ħofra ż-Żgħira from 2017. The operator shall submit to the authority a proposed methodology for this study, which shall be to the authority's satisfaction.

Enemalta plc was seeking for a contractor to delegate the execution of a marine ecological survey in compliance with the above requirements. Before that, the awarded contractor was requested to submit to the contracting authority a Method Statement giving details of the methodology for the marine ecological study, which would have been forwarded by the contracting authority to the Environment and Resources Authority (ERA). The marine ecological survey couldn't be carried out prior to the approval of the Method Statement by ERA. The survey should include comparison of the findings in a particular year between the different sampling stations as well as the control station at il-Ħofra il-Kbira, comparison of results of the current year with those of previous years, a trend analysis comparing the findings and results over a number of years to show the status of the species over a number of years and conclusions should be drawn depending on the findings of the survey. Mitigation measures should also be proposed in case the findings show that there is a decline in the conservation status of the species.

Table of Contents

SECTION 1 – INSTRUCTIONS TO TENDERERS

4. Clarification Meeting/Site Visit/Workshop

5. Selection and Award Requirements

6. Criteria for Award

7. Data Protection

SECTION 2 – SPECIAL CONDITIONS

Article 1: Definitions and conventions

Article 2: Notices and Written Communications

Article 5: Supply of Information

Article 6: Assistance with Local Regulations

Article 7: Obligations of the Contractor

Article 8: Code of conduct

Article 13: Medical, Insurance and Security Arrangements

Article 15: Scope of the Services

Article 16: Personnel and Equipment

Article 18: Execution of the Contract

Article 20: Modification of the Contract

Article 24: Technical Reports

Article 26: Payments and Interest on Late Payment

Article 27: Financial Guarantee

Article 34: Termination by the Contracting Authority

Article 35 Termination by the Contractor

Article 39: Dangerous chemical handling and transportation

Article 40: Handling, Transportation and Maintenance of equipment containing HCFCs and fluorinated gases (F-gases)

Article 41: Contamination of soil and water

Article 42: Waste management

Article 43: Certification of vehicles

Article 44: Environmental audits

Article 45: Environmental liability

SECTION 3 –SPECIFICATIONS/TERMS OF REFERENCE

SECTION 4 – SUPPLEMENTARY DOCUMENTATION

4.1 – Draft Contract Form

4.2 – Glossary

4.3 – Specimen Performance Guarantee

4.4 – Specimen Tender Guarantee (Bid Bond) – not applicable

4.5 – Specimen Pre-Financing Guarantee – not applicable

4.6 – Specimen Retention Guarantee – not applicable

4.7 – General Conditions of Contract

4.8 – General Rules Governing Tendering

SECTION 5 - FINANCIAL OFFER

SECTION 6 – DOCUMENTS INCLUDING DRAWINGS

List of documents and drawings attached

6 Blockchain and smart contracts in Contract Management

As they offer a safe and transparent mechanism to track and manage project-related information and agreements, blockchain technology and smart contracts are two innovative technologies that may significantly impact the field of project and contract management. These technologies have the potential to revolutionize the way contracts get created, executed, and enforced, bringing greater efficiency, security, and transparency to the contract management process. In fact, smart contracts, self-executing, self-verifying and self-enforcing contracts embedding the terms of the agreement between client and contractor into lines of code, can be used to automate various aspects in the process of managing contracts and agreements, including contract creation, negotiation, performance, and payments, reducing the need for manual processes and increasing efficiency. Smart contracts present an innovative alternative for enhancing the traditional contract management process, in particular for automated execution of contract clauses involving payments. For example, a smart contract can be programmed to automatically send a payment to a contractor when a specific project milestone is achieved, or instantly pay a supplier at the delivery of materials or to automatically renew a contract when certain conditions are met. They can also allow for better transparency and security. In fact, smart contract technology provides a secure, immutable, and transparent platform not only for automated execution of contract clauses but also for storage of contractual data. The terms of the contract and the details of the project are stored on the blockchain, a decentralized and reliable environment, and cannot be modified without the consent of all parties. Because blockchains are decentralized ledgers maintained by a network of nodes, it is very difficult to hack or manipulate them. This can help to improve collaboration and communication among the parties, and increase the transparency and accountability of the contract management process, since all relevant data are recorded and stored in an accessible and secure location, and all parties can view the terms and the status of the contract at any time. Establishing trust among client, main contractors and subcontractors is traditionally challenging, and deficiencies in trust have been identified as a major obstacle to supply chain integration and collaboration efforts. Trust is, however, essential for reducing transaction costs, responding to new information, and achieving significant time and cost savings (Manu, Ankrah, Chinyio, & Proverbs, 2015). Notarization-related applications are intended to eliminate the verification time of documents' authenticity. The blockchain can be used to store every document in a distributed ledger, providing a perfect notarization of each creation, deletion, and updating across the system. This type of application can be employed for recording construction quality data, such as the quality of raw materials, installation, and construction progress information, as well as resource consuming data, such as concrete, scaffold, formwork, steel, and equipment. One key advantage of using smart contracts in contract management is that they reduce the need for lawyers, banks, insurance companies and other financial intermediaries. This reduces complexity, expedites the process and decreases costs, namely transaction and administrative fees associated with the management of contracts. The accuracy of the process is also increased thanks to the reduction of the risk of errors and fraud. Smart contracts may be employed for the purposes of contract dispute resolution providing a solution for the problem of enforcing online dispute resolution (ODR) decisions and bypassing the recognition and enforcement procedures through which State courts traditionally exert a certain control over arbitration. Overall, the use of blockchains in project management and contract management offers numerous benefits, including process streamlining, automating routine tasks, such as sending out reminders or updating project timelines, and payments, improving efficiency, security, trust and transparency, reducing the risk of errors and fraud, minimizing the need of intermediaries. Despite their many benefits, there are also drawbacks and challenges associated with the adoption of the blockchain technology and smart contracts in contract management. First of all, the technology is still relatively new and complex, not much study has been made on the application of smart contracts in traditional contract administration processes and there is a shortage of technical experts who know how it works or how to build and maintain it. The presence of many different blockchain platforms, each with its own set of protocols, standards, and interfaces, makes it difficult for organizations to pick the right one and integrate their existing systems. Another challenge is that there may be legal and regulatory barriers to the adoption of blockchain and smart contracts, particularly in highly regulated industries. Many jurisdictions are still developing laws and regulations, and there is uncertainty around issues such as data privacy, intellectual property, and liability. Security remains a significant concern when it comes to blockchain technology, as frequent have been the instances of smart contract bugs or vulnerabilities that hackers managed to exploit. Finally, it is difficult to image a context where smart contracts will completely replace the use of traditional methods of project and contract management.

6.1 Legality of smart contracts

Innovations brought by digital technology have required to re-evaluate the current mechanisms employed to regulate society. Code has emerged as the dominant medium to regulate people’s conduct on the internet. With the advent of blockchain and smart contracts, the role of code has acquired an even more significant influence. While it is true that code is increasingly assuming some of the typical functions of law, it is also true that law is progressively starting to assume the characteristics of code. The main consequence lays in the passage from the traditional notion of "code is law" to the new concept of "law is code". "Code is law" refers to an increasing reliance on technology not only as an aid in decision-making but also as a means to directly enforce rules. "Law is code" is an expression that summarizes the tendency by private and public institutions to replace current laws and regulations, which can only be enforced ex-post through State intervention, by technical regulation, which can be enforced ex ante through code (De Filippi & Hassan, 2018). Blockchain technology, combined with smart contracts, reinforces the trend to rely on code rather than on law, especially for regulating transactions, and, now more than ever, paves the way to the prospect that law progressively assumes the characteristics of code.

A contract is a legally enforceable agreement between two or more parties (Kronman, 1985), characterized by mutual promises or obligations. When a smart contract is created, it includes all the terms and conditions of the agreement, as well as the code that will be run when those criteria are satisfied. This indicates that smart contracts are fully self-executing and do not require any human intervention to enforce the provisions of the agreement. While enforceability of traditional contracts is ensured by public law and courts, with smart contracts the agreement is automatically enforced by lines of code embedding the terms of the contract and performance occurs without recourse to the courts. A smart contract asks its parties to tie themselves to the mast like Ulysses and ex ante commit to abiding by the terms of the agreement (Raskin, 2016). Legal validity of smart contracts has been a subject of debate in the legal community. The fact that it is possible to prove mathematically that transactions on a blockchain are valid, know who owns the data saved in a blockchain-based ledger and demonstrate that that data has not been tampered with, does not however mean that blockchain-based transactions are legally binding. From a legal perspective, a contract is regarded as valid if it meets the legal requirements already discussed in the previous chapter: capacity of the parties, mutual agreement or meeting of the minds, consideration and legality of the subject matter. Even if these requirements are met, there are additional legal issues that must be addressed. It's uncertain whether existing legal and regulatory frameworks adapt to the wide variety of blockchain applications and use-cases or there is the need to develop a tailored and standalone blockchain legislation. Several legal rules are conceived to be quite generic so to be applied to various situations and encompass as many cases as possible. In these instances, it's difficult to foresee how the role of a judge will be replaced by a smart contract. The Statute of Frauds requires that certain agreements must be incorporated into a written form and signed by both parties in order to be legally binding; evidence that the meeting of the minds occurred is not sufficient in those instances. However, smart contracts are not written in the traditional sense, and there is no physical signature. Furthermore, a known benefit of Ethereum contracts is that their code is immutable, enabling parties to execute them without trusting each other. However, the assumption that contracts are complete and fixed arrangements undermines the flexibility and contingency of contract law and its ability to execute variations, rescission, rectification, restitution, and specific performance. Just as an example, the law recognizes certain excuses for non-performance or modification, such as impossibility or impracticability, which may pose a problem for smart contracts that need to be updated to incorporate changes in the legal landscape. Flexibility is a key feature of ensuring that commercial agreements operate as intended, and is not something that is offered by smart contracts. While computer coding tends to be unambiguous, traditional legal contracts are frequently drafted with some "flex room" in them, e.g. expressions such as "reasonable efforts" and "material adverse change" have a long history of legal interpretation behind them, and in many cases helps parties resolve issues by not determining in advance exactly what the obligation involves. (Leung, 2018). The inflexibility of smart contracts needs to be addressed to better align with traditional contract law. The transparent contract standard, ERC1538, constitutes an attempt to solve this issue providing a contract architecture that makes upgradeable contracts flexible, unlimited in size, and transparent (Mudge, 2018). It comes with the following benefits: a way to add, replace and remove multiple functions of a contract atomically (at the same time); a user interface which shows all the upgrades a contract has had, with written descriptions of the changes; and possibility for an upgradeable contract to become immutable in the future if desired. Immutability is an essential feature to build trust, but it can become a problem when the smart contract does not behave in the way that it was intended. A notable case around this topic happened in 2016. The DAO, a decentralized autonomous organization and a form of investor-directed venture capital fund, was subjected to an attack by an individual or a group of individuals who exploited some bugs in the withdrawal mechanism which resulted in the transfer of 3.6 million ether, around a third of the 11.5 million ether that had been committed to The DAO - valued at the time at around $50M. The damage was far greater considering the collapse of the market price of ether as a result of market panic. Some members of The DAO and the Ethereum community debated that the attack was valid and remained strong advocates of the “code is law” principle, others wanted the re-appropriation of the subtracted amount and The DAO to be shut down. This led to the Ethereum hard fork. Reality is that writing large amounts of code with no flaws is extremely difficult, so the contractual parties should agree on a process for implementing a new version of a smart contract. Finally, since smart contracts can be hacked due to the possibility of bugs and errors in the code, the responsibility and liability for ensuring their integrity must be addressed. Two key aspects of liability should be focused (Agostini, 2021): liability of core software developers and liability of network actors. Charging core software developers with responsibility for a potential unlawful usage of the program does not seem proper. Enforcing liability on the network’s participants is not an easy task. Despite these issues, smart contracts are gaining legal recognition in many jurisdictions. The Uniform Electronic Transactions Act (UETA) establishes the legal equivalence of electronic records and signatures with paper writings and manually signed signatures. In 2018, three states, Arizona, Nevada and Tennessee, made legislative amendments to specifically recognize records and signatures secured through blockchain and smart contracts as electronic records and signatures under the UETA (Bosco, 2018). All of the enacted and proposed statutes give legal recognition to electronic records created, stored, or verified by the use of a blockchain. A contract that contains a smart contract term cannot be denied legal effect, validity or enforceability solely on that basis. Later, Florida, Illinois, Nebraska and Ohio, following the path traced by the previous three states, proposed their own revision to the UETA (or to the Electronic Signatures and Records Act, New York’s equivalent to the UETA). Similarly, in the European Union, the Electronic Identification and Trust Services Regulation (eIDAS) provides legal recognition for electronic signatures and electronic contracts. According to eIDAS, digital documents cannot be denied legal force simply because they are in electronic form. This supports the potential for legal standing for the data contained in a blockchain-based registry or contract. The situation becomes more complex when it comes to digital signatures. eIDAS recognizes three different levels of eSignatures: simple, advanced and qualified. Blockchain technology can only meet the technical criteria for the first two but, to be legally binding, it needs to meet also the highest standard, which forces to either use a recognized Trust Service Provider (TSP) or become a recognized TSP (Lyons, Courcelas, & Timsit, 2019). As it happens with many innovative technologies, the creators and early adopters of smart contracts are ideologically driven and believe that the innovation can significantly change the structure of society and its relationship with the conventional centralized system. Many authors claim that smart contracts, especially when stored and executed with the help of blockchain technology, allowing automatic execution of legal obligations, will cause the end of external enforcement mechanisms such as lawyers and courts, or at least greatly reduce the need and the extent of monopolized police and legal services. They believe that smart contracts are autonomous in nature, do not require a legal system for their existence, may operate without any overarching legal framework and represent a technological alternative to the whole legal system. Smart contracts are transnational and executed uniformly regardless of the differences in national laws, making them a new type of regulator governing relations in cyberspace (Savelyev, 2017). The notion that smart contracts make contract law and the entire legal system obsolete can be challenged. Smart contracts need contract law just like traditional contracts, and the applicable contract law can be determined with the help of the traditional rules of private international law (Rühl, 2020). Smart contracts depend on a legal system to determine whether there is an enforceable legal obligation. The code of a smart contract cannot determine whether an obligation has been validly created or whether the parties have validly agreed to use the smart contract. Smart contracts need a legal system as a normative point of reference to determine whether they are valid or invalid, legal or illegal. The question, therefore, is not whether smart contracts are subject to law but rather which law they are subject to, and which law determines whether a contractual obligation has been validly created and can be enforced with the help of a smart contract. The fact that smart contracts are generally operated by computers located in different jurisdictions may make it more difficult to identify the law or jurisdiction applicable to the contract (Durovic, 2019). However, when this happens with traditional contracts, the question of which law applies to a contract is determined by the rules of private international law (Rühl, 2020). The principle of party autonomy, which allows parties to choose the governing law for their contracts, is a cornerstone of the Rome I Regulation (European Parliament, 2008). The parties can make an express choice of law, which can be included in the smart contract or in a separate declaration. Alternatively, a choice of law may be implied if the smart contract or the contract it serves to execute is so obviously tailored to a particular legal system that it can be assumed that the parties wanted the contract to be governed by that law. Article 4 of the regulation provides specific choice of law rules for certain types of contracts, such as contracts for the sale of goods or contracts for the provision of services. The same applies for smart contracts that fall under one of these categories. However, if the smart contract in question does not fall under one of these categories, the residual choice of law rule calls for the application of the law of the country where the party required to effect the characteristic performance has its habitual residence. The claim that smart contracts will put lawyers out of their job is also too catastrophic. Lawyers can provide guidance to computer coders on how to draft and code contracts, ensuring that they are legally compliant and reflect the intentions of the parties involved. Additionally, lawyers can help with contract interpretation in case of disputes, as smart contracts may not always reflect the nuances of real-world scenarios. It’s true, however, that lawyers will need to adapt to new skills required for working with smart contracts.

6.2 Payment automation

Many industries are notorious for payment disputes, with contractors and subcontractors often struggling to receive payments owed to them. The impact of payments withheld or not paid by clients to contractors results in significant cash flow problems and financial instability, making it challenging to pay employees, suppliers, and subcontractors, which can ultimately lead to business failure. The knock-on effect on subcontractors needs to be addressed too. When the owner becomes insolvent, both the contractor and subcontractors are often left in a position where they have not been paid for the work they have performed. Contractors know about this risk and place “pay-when-paid” or “pay-if-paid” provisions in their subcontracts. Such provisions attempt to make receipt of payment from the owner a condition precedent to the general contractor’s obligation to pay subcontractors, thus transferring the risk of the owner’s nonpayment from the general contractor to the subcontractors (Hill & McCormack, 2011). Some clients may engage in opportunistic behaviours, intentionally withholding payments to threaten and gain leverage over contractors and subcontractors, or to punish them (Cohen, 1991). According to a report by the Australian Senate Economics References Committee (Senate Economics References Committee, 2015) contractors can use strategies to get subcontractors to accept long payment claim periods, ranging anywhere between 30 and 90 days. Payments are in general made through a cascade system: the employer to main contractor, main contractor to the subcontractors, and so on down the chain. The traditional progress payment system may cause serious issues if payments are not made or are made late because of its cascading nature. Payment problems are identified as one of the top disputes causes for construction projects, and the number one source for international project disputes in China (Chan & Suen, 2005). Late payments from the employer have a detrimental effect on the contractor's cash flow and frequently cause the payment of suppliers and subcontractors to be postponed as well. A questionnaire survey to subcontractors, contractors, and employers in the United States showed that 89% of subcontractors claimed that the payments are delayed by more than 45 days after the completion of the work (Arditi & Chotibhongs, 2005). The UK Office of Government Commerce performed interviews with senior management personnel of contractors, sub-contractors, suppliers, and consultants, which affirmed that delays up to 60 days for the payments from the employers were common (2007). Even if the contractor receives payment on time, the prompt flow of payment down to the subcontractors is not always assured because the contractor may utilize the funds to finance other projects or may try to increase his or her profits by accruing interest (Latham, 1994). The advice that businesses should take proactive steps to protect their rights and mitigate risks by ensuring they have strong contracts and performing due diligence on clients may not always prove useful. Therefore, a reliable and efficient progress payment system is becoming increasingly necessary for resolving payment issues and achieving successful projects. The Project Bank Account (PBA) system is an alternative payment method proposed to improve the traditional payment system for construction projects. Under it, the employer can deposit the entire project lump sum or the amounts due upon approval of progress payment reports in a separate bank account

(Macaulay & Summerell, 2019). The amounts owed to the contractor and subcontractors are released from the PBA upon the employer's authorization. The benefits of this system include faster payments, savings due to reduced recourse to financing and debt, and protection from contractor’s insolvency. However, set-up and administration costs are high, which is a barrier to widespread adoption. The report “Inquiry into construction industry insolvency in NSW” (Collins, 2012) made a proposal to establish a Construction Trust aimed to provide greater financial security for contractors and subcontractors working in the construction industry. The Construction Trust was intended to address the problem of payment disputes and non-payment through a more secure payment mechanism. The fund would hold payments from a client in trust until they were due to be paid in order to safeguard and protect subcontractors and suppliers from a head contractor insolvency. The money subject to the trust is all money for which a progress claim has been certified and a payment made in respect of that claim. It may be money paid by the principal into a trust account for a contractor as a consequence of a claim received by the principal, or by a contractor into a trust account for a subcontractor as a consequence of a claim received by the contractor, or by the subcontractor into a trust account for a sub-subcontractor as a consequence of a claim received by the subcontractor, and so on (Gaussen, 2018). The Construction Trust would guarantee that payments from the principal to the head contractor would follow through to subcontractors and suppliers. That of appointing a Construction Trust is a robust proposal, which however comes with some downsides. One of the primary concerns is trust. In order for the trust to function effectively, clients must be willing to contribute their funds and trust that they will be used according to what was intended. Another potential issue is centralization. The proposed trust would be a centralized entity that would hold funds from multiple clients and distribute them to contractors and subcontractors. This creates a single point of failure, and there could be significant consequences for all parties involved in case of mismanagement or a fallout of security systems resulting in a breach to the fund's reserves. Finally, the establishment of an intermediary entity requires administrative and legal work, leading to slower times for the deployment of procedures and an increase in costs. Overall, while the Construction Trust and PBA models, or any centralized model having the same working principle, have the potential to address some of the challenges associated with payment disputes, they also come with their own set of challenges and risks. An alternative solution, which grants the same benefits while minimizing the aforementioned issues and risks, is constituted by smart contracts. Smart contracts allow for a set of instructions to be incorporated into a contract and the automatic execution of contract clauses upon the occurrence of agreed conditions, which, following the use case of the trust model above, could be the self-enforced release of payments to subcontractors and suppliers once a contractor receives the contracted amount from the owner. Blockchain technologies and smart contracts have the potential to improve payment security, thus protecting main contractors, subcontractors and suppliers against risk of the insolvency of the principal or late payments. Temporary barriers to their adoption in the construction industry have been identified in acceptance, reliability, interoperability and financial regulation (Mason, 2017). While it is true that current payment applications can achieve payment automation, those, even if computerized, cannot support reliable automation of progress payments due to their reliance on centralized control mechanisms and lack of guaranteed execution (Hamledari & Fischer, 2021). With a smart contract, the funds or cryptocurrencies can be embedded into the contract to protect general contractors, subcontractors, and suppliers against the risk of insolvency and late payments. Moreover, smart contracts can also be interlinked with each other to create a web of payments, and payment milestones can trigger automated payments to contractors, subcontractors, and suppliers. For instance, when a construction project achieves a payment milestone such as structure completed to building Level 10, the general contractor will get an automated payment through the smart contract with the project client. This event will also automatically activate all the related payments through the smart contracts between the general contractor and their subcontractors or suppliers, based on the contract conditions (Wang, Wu, Wang, & Shou, 2017). Here follows a description of the most commonly used types of payments in the construction industry. Interim payments are stage payments or instalments paid by the client to the contractor during the course of a project. They are designed to avoid that contractors and subcontractors need to wait until the end of a project, which may take months or years, before getting any remuneration for partially completed work, protecting their cash flow and operational efficiency. Interim payments can either be made at regular intervals, such as on a monthly basis, or can be made following the completion of stages of the works. Progress payments are a type of interim payment made by the client to the contractor based on the completion of predetermined milestones or stages of the project. Retention payments refer to the amount that the employer typically retains from the interim payments until the project is completed to the requested level of satisfaction. They serve as security for future performance, ensuring that the contractor completes the work to the required standard and addresses all issues that may arise at project completion or during the warranty period. They however bring the downside of damaging contractors’ cash flow. Advance payments are payments made by the client to the contractor before work on the project has commenced, to assist him with the initial expenditure in respect of site mobilization, materials, equipment, a fair proportion of job overheads and preliminaries. Advanced payments increase the contractor’s liquidity in the form of cash, constitute an important source of financing and can be classified as deferred income, which represents funds received by the contractor for the services he has agreed to supply in the future (Palliyaguru, Amaratunga, & Rameezdeen, 2006). They provide several benefits (Omopariola, Windapo, Edwards, & Chileshe, 2022): they improve contractors’ cash flow, aid contractors’ prompt performance on construction project operations, help mitigate the risk of price fluctuations that lead to project cost overruns. After the works have been completed, the project is finalized and all necessary approvals have been obtained, the employer and contractor will engage in the process for final payments. Apart from the disbursement of the outstanding amount, they may also include a final release of retention payments. Liquidated damages are payments made by the contractor to the client in the event that the terms of the contract are breached, mostly when failing to meet agreed-upon deadlines or performance standards. In building contracts, liquidated damages usually relate to the contractor failing to achieve practical completion (i.e. completing the works so they can hand over the site to the client) by the completion date set out in the contract. They are defined by predetermined amounts to be paid on a daily or weekly rate that parties agree upon during the formation of a contract. They must not be regarded as penalties or a form of punishment, since they aim at recovering the actual loss the client is likely to incur if the contractor fails to meet the completion date. Lastly, performance incentives are payments made by the client to the contractor to reward him for performing above performance targets, such as completing the project ahead of schedule or under budget.

6.2.1 Smart contract-based payment systems

The use of smart contracts and blockchain technology has been identified as a viable way to automate contract administration and project management workflows. Such automation would eliminate the need for third-party intermediaries and manual payment processing. A smart contract-based solution for the autonomous handling of progress payments fills the gap between contractors' cash flow and the actual progress at job sites. A significant percentage of a project's cash flow is made up of progress payments, which serve to compensate the general contractor, subcontractors, suppliers, and equipment providers for the work performed. A smart contract-based payment security system named SMTSEC was introduced to safeguard subcontractors and contractors from the insolvency of the project owners (Ahmadisheykhsarmast & Sonmez, 2020). The SMTSEC ensures security of payment of construction contracts through an automated computerized protocol that runs on a decentralized blockchain. By ensuring security of payment for works in progress without requiring the administrative fees and burdens of trusted middlemen like attorneys or banks and eliminating the complex and confusing nature, administrative demands and set-up and administration costs of PBAs, the SMTSEC system offers a novel approach for the timely and transparent payment of construction projects. A real construction project was used to study its potential benefits and pitfalls. SMTSEC guarantees availability of the funds for a progress payment period by blocking the projected progress payment amount at the beginning of the progress payment period following a contractor's request. It also provides transparency as the contractors, subcontractors, and suppliers can track all the transactions made from the employer's wallet. Once the employer approves a progress payment, the smart contract automatically transfers the payment amount to the contractor's, subcontractors', and suppliers' wallets according to the agreed terms. The procedure is repeated for each progress payment period until the project is completed. The smart contract conditions are unique for each project and should include the fiat currency (Cur), smart contract cryptocurrency type (CryT), contingency amount (Cong), period that the employer's funds will be blocked and cannot be used (PeriBloc), period that the employer's blocked funds can be used for progress payments (PeriPay), and the percentage amount of payments that will be paid directly by the employer to the subcontractors and suppliers ($SubPer_i$). The employer's, contractor's, and subcontractors' and suppliers' wallet addresses should also be included in the smart contract. The smart contract procedure enables the release of the blocked amount if the contractor does not request a progress payment, or the employer does not approve the contractor's payment request within PeriPay, to prevent the employer's funds from being blocked indefinitely in case of a dispute. Hence, the maximum period that the funds of the employer will be blocked is PeriBloc plus PeriPay. SMTSEC consists of two modules. The first module is an add-on software developed in Microsoft Project 2019, which transfers the required schedule and cost data to the smart contract. The second module is a decentralized application based on a smart contract designed to be deployed on the Ethereum blockchain. The DApp has a frontend that provides the user interface and a backend that executes the smart contract procedure. The SMTSEC system was implemented as a shadow system alongside the conventional payment system for a construction project in Turkey. The project involved the construction of a 3000 m2 powerhouse building, with a budgeted cost of $20 million for the civil works. There were two subcontractors involved in the project, one, SC1, responsible for the reinforcement works and the other, SC2, for the structural concrete works. The main contractor (MC) performed the rest of the civil works. The project started on March 1, 2019, and the MPP Parser module of SMTSEC was used to calculate the projected progress payment (PPP) amount for March 2019 and to create a ".TXT" file to transfer the first PPP amount to the DApp. The first month's PPP of $272,417.74 was converted to 1980.50 ETH at the exchange rate of 137.55 ETH/$. With a Cong of 20%, the Bloc was calculated as 2376.60 ETH by the DApp, which was then blocked by the smart contract upon approval by the employer. The first progress payment request was made on April 1, 2019, with the actual progress data of the contractor at the end of March 31, 2019 used to calculate the progress payment amounts for the MC, SC1, and SC2. The progress payment amounts for the MC, SC1, and SC2 were calculated as $144,897.92, $52,334.95, and $47,277.50, respectively, by the MPP Parser. The fiat currency payment amounts were converted to 1369.23 ETH, 184.02 ETH, and 166.24 ETH, respectively, by the DApp module of SMTSEC at an exchange rate of 142.20 ETH/$ for April 1, 2019. The excess amount of 657.12 ETH was released and transferred to the wallet address of the employer (WAdEM) by the smart contract. The transactions on the Ganache blockchain were performed within seconds, while standard transactions on the Ethereum blockchain take between 15 seconds and 5 minutes. The deployment cost of the smart contract for the case project was 0.0400752 ETH or $5.51 at an exchange rate of 137.55 ETH/$. SMTSEC revealed the potential of smart contracts to protect main contractors, subcontractors and suppliers against the insolvency of the principal or late payments through a decentralized, secure, efficient, transparent, and trustworthy payment system. However, it has some limitations. Fluctuations in the value of cryptocurrencies represent a risk. The workflows for submitting and reviewing payment applications still follow traditional solutions and are susceptible to shortcomings caused by centralized control. The authors affirm that “the proposed smart contract payment security system receives the schedule and cost data through a project management software, hence it does not present an automated progress payment system." The mere computerization of payment application is not the goal for smart contracts, as automation can be achieved using other alternatives. There is a need for a smart contract-based solution that can automate the transition from progress data, enabled by reality capture technologies, to construction payments. They conclude suggesting that the integration of building information modeling presents a major potential for representing the construction works as digital objects in the smart contracts and could enable automated smart contract progress payment systems.

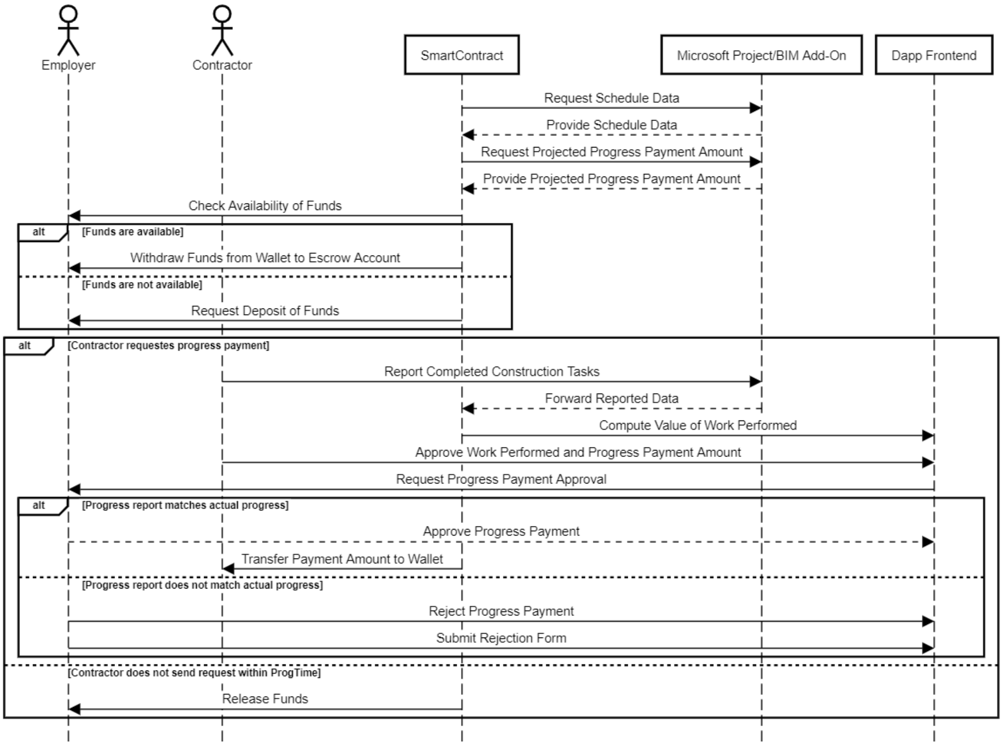

Employer
Contractor
SmartContract
Microsoft Project/BIM Add-On
Dapp Frontend
Request Schedule Data
Provide Schedule Data
Request Projected Progress Payment Amount
Provide Projected Progress Payment Amount
Check Availability of Funds
alt
[Funds are available]
Withdraw Funds from Wallet to Escrow Account
[Funds are not available]
Request Deposit of Funds
alt
[Contractor requestes progress payment]
Report Completed Construction Tasks
Forward Reported Data
Compute Value of Work Performed
Approve Work Performed and Progress Payment Amount
Request Progress Payment Approval
alt
[Progress report matches actual progress]
Approve Progress Payment
Transfer Payment Amount to Wallet
[Progress report does not match actual progress]
Reject Progress Payment
Submit Rejection Form
[Contractor does not send request within ProgTime]
Release Funds

Figure 2 - Sequence diagram of a smart contract for progress payments

The architecture, engineering, and construction (AEC) industry has shown a growing interest in using automated progress monitoring at job sites, utilizing reality capture technologies, such as robotics, artificial intelligence, and building information modeling (BIM). These technologies are used to capture, analyze, and model job site conditions, creating semantically rich as-built BIMs and progress data. The potential benefits of this approach are streamlining payments and automating the transition from product flow to cash flow in the construction supply chain. However, payment automation is still far from a reality, and traditional payment applications are still in use. Digital payment management systems have been developed, but they cannot support payment automation due to their inability to use the output of reality capture technologies and their reliance on manual and intermediated workflows. To address these limitations, a smart contract-based solution was proposed (Hamledari & Fischer, 2021) to automate and decentralize the conditioning of construction payments on progress assessments. However, smart contracts alone cannot support payment automation because they lack a link to physical reality. So, the authors propose using reality capture solutions, sensors, machine intelligence and 4D/5D BIM to capture, analyze, and document the status of physical reality at construction sites. The blockchain based methods require the off-chain real world information to be connected to the blockchain and broadcasted to a smart contract which automates the on-chain payment settlements and the transfer of lien rights. The InterPlanetary File System (IPFS) is used to create a distributed and content-addressable file-sharing solution to securely share physical reality among project participants. The proposed method was evaluated in two commercial construction projects, where robotic reality capture was used to document progress at job sites. The solution is composed of an off-chain component, which captures the product flow and securely stores it in a distributed manner, and an on-chain component, the smart contract that uses progress data to achieve on-chain payment settlement. It was successfully used to process payments to a series of subcontractors in two commercial construction projects where progress monitoring was performed using a camera-equipped unmanned aerial vehicle (UAV) and an unmanned ground vehicle (UGV) equipped with a laser scanner. The captured construction progress data are then analyzed using machine intelligence to determine the percentage completion data for various scopes of work. The resulting progress data is automatically incorporated into the as-built 3D/4D BIM, allowing for a data-driven approach to valuing completed work. However, there are challenges associated with storing the product flow on a public blockchain, among which the prohibitive cost of on-chain storage and the risks to project information integrity due to the auditability of the public blockchain, so content-addressable file sharing on a private IPFS network was created among the owner, the general contractor, and the subcontractors. A centralized or distributed data storage and management system, like the Common Data Environment (CDE), may also be used (Sigalov, et al., 2021). Once a file containing key data such as as-built BIMs and progress data, is stored on IPFS, a unique content identifier (CID) is automatically generated, which can be used to retrieve it from the network. The CIDs corresponding to key project information used in payment processing are communicated to the smart contract and included in state transitions, recorded on the public ledger. The off-chain client constructs a transaction that is broadcasted to the smart contract via a JSON-encoded remote procedure call, communicating the valuation of the work, the public key of the subcontractors performing the valued work, and the CIDs of input data used for the valuation of incremental progress. The on-chain component has two objectives: to settle payments between project participants according to off-chain product flows and to transfer lien rights alongside payments. Lien rights refer to contractors' claim on a property if they are not compensated for their work. The proposed system utilizes the ERC-721 standard to create a non-fungible "LIEN" token that represents the lien rights to a property. The ownership of the token, and therefore the right to the underlying physical asset, is managed by the smart contract, which acts as an escrow account, and gets recorded on the Ethereum blockchain. The smart contract's public function, ReceiveUpdate, receives and formats the information before triggering internal functions to initiate on-chain payment settlement. The internal functions handle payment settlement, including transferring payment in Ether and issuing lien tokens to the contractors responsible for the work. The payment metadata are incorporated into the lien token, and upon payment settlement, the corresponding lien token is transferred to the owner. The method was used for processing progress payments on two commercial construction sites: one in the state of Ontario (Canada), visited for data collection over a period of 4 weeks, and the other in the state of California (United States), where data capture was performed for a period of 5 months. The accuracy of the remote sensing solution, defined as the percentage of elements for which the state of progress was correctly identified after manual verification from the human personnel, was equal to 95% for the first project. The smart contract’s payment settlement was 100% accurate and payments were successfully processed for the entire scope of work. This approach could lead to more efficient and reliable payment processing in the construction industry, with potential benefits for all stakeholders. The method can process payments within a few days, making it beneficial for an industry where it takes engineering and construction firms around three months to receive payment. Some steps involved in preparing inputs and analyzing outputs still require human involvement. The accuracy of the input data is essential for reliable payments and that of the payment system is dependent on the smart contract and reality capture technique. A similar model, consisting in a novel building information modeling (BIM) integrated smart contract progress payment administration system, in which as-constructed BIM is used to link the real world with the blockchain, was more recently presented (Sonmez, Ahmadisheykhsarmast, & Güngör, 2022). The system is based on a Building Information Model that is used to accurately track the progress of the construction project. It was applied to a $8.6 million process building project contracted on a lump sum basis. The BIM model used for the project included a total of 2487 BIM objects for the foundations, elevated slabs, walls, doors, structural steel, stair and guardrail items, pipe fittings, pipes, and mechanical equipment. The deployment cost of the proposed smart contract depends mainly on the number of BIM objects included in the contract. As the number of BIM objects increases, the deployment cost increases. The deployment cost of the smart contract for the case project with 2487 BIM objects was 1.197 ETH or $4907.7 on the Ganache blockchain, at an exchange rate of 4100 ETH/$. The system enables the contractor to select the BIM objects that have been completed for the period and request the payment. The smart contract will first import the progress data from the contractor's computer, then will calculate the progress payment for the period in U.S. Dollar ($) based on the embedded costs of the BIM objects, and convert it to ETH. The employer should compare the claimed progress with the actual progress using the visualization property of the system, Once the employer approves the transaction through MetaMask, the smart contract will transfer the payment to the contractor's wallet. Although the system represents a step forward over the conventional progress payment system, it still requires the involvement of the contractor and the employer and does not provide a fully autonomous payment administration system.

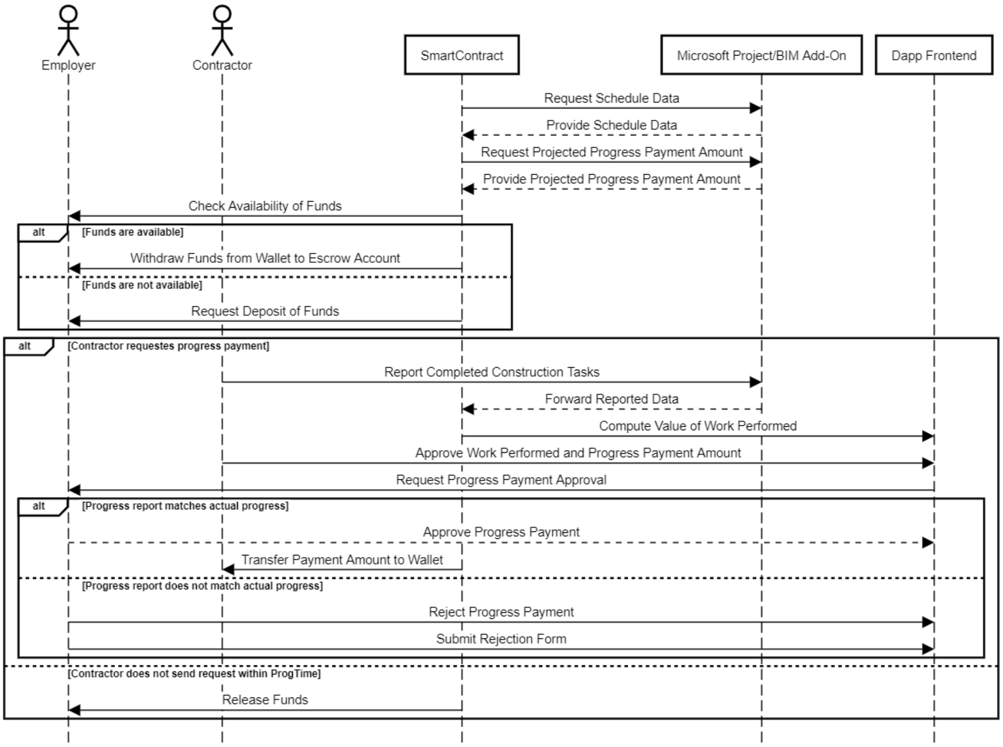

Employer
Contractor
SmartContract
Microsoft Project/BIM Add-On
Dapp Frontend
Request Schedule Data
Provide Schedule Data
Request Projected Progress Payment Amount
Provide Projected Progress Payment Amount
Check Availability of Funds
alt
[Funds are available]
Withdraw Funds from Wallet to Escrow Account
[Funds are not available]
Request Deposit of Funds
alt
[Contractor requestes progress payment]
Report Completed Construction Tasks
Forward Reported Data
Compute Value of Work Performed
Approve Work Performed and Progress Payment Amount
Request Progress Payment Approval
alt
[Progress report matches actual progress]
Approve Progress Payment
Transfer Payment Amount to Wallet
[Progress report does not match actual progress]
Reject Progress Payment
Submit Rejection Form
[Contractor does not send request within ProgTime]
Release Funds

Figure 3 - Sequence diagram of a BIM-integrated smart contract

A smart contract application enables to execute secure and automated payment of the retention at project completion or after the warranty period has expired (Ahmadisheykhsarmast, Sönmez, & Sönmez, 2020). The proposed system is named Retention Payment (RETPAY) and is designed to address the issue of delayed or non-payment of retention amounts in the construction industry. In general, a portion of the retention is paid to the contractor during project completion and remaining is paid after the warranty period. It does not only enable automated payment of retention but also performs storage and record keeping of the project completion data on a secure, reliable and trustworthy blockchain platform. The RETPAY system consists of two modules. The first module is an add-on software that captures data from Microsoft Project. The purpose of this module is to enable contractors to use their existing project data and software to prepare the list of completed works. Once the list is prepared, the contractor exports the list of completed works' unique IDs and their completion dates to a TXT file, which is used by the second module. The second module is a decentralized application (DApp) consisting of a web part and a smart contract part. The retention amounts of works are embedded into the smart contract along with their unique IDs. When the contractor requests approval for partial completion and retention payment, the DApp uses the list of completed activities in the TXT file to determine the amount of retention payment. The client gets notified to approve the payment amount. If the client approves the payment amount, the agreed retention amount for the report period is transferred from the client's wallet to the contractor's wallet, and the list of completed activities that are stored in the blockchain are updated along with their completion dates. Once the contractor receives the retention amount in ETH, they can convert it to any fiat currency in the local cryptocurrency exchanges. If the client does not approve the list of works completed and the retention payment amount, the DApp will notify the contractor with the reasons of rejection given by the client. The proposed RETPAY system was tested on a real construction project of a process plant. The retention payments were made 50 to 60 days after initiation of the partial completion process, and 30 days after the approval of partial completion of works, negatively impacting the contractor's cash flow. RETPAY reduced the duration of retention payments substantially, allowing payments to be made within seconds after approval of the client. All retention clauses are enforced by the smart contract, minimizing the need for third parties such as banks and lawyers. As a final note, locking up the funds for retention in the smart contract retention payment application, to release them at completion of the works, would enable security of payment of retention.

Construction supply chain management faces several challenges including lack of trust. IoT and BIM integration can alleviate them by connecting physical construction objects with virtual BIM objects. Blockchain technology can provide accountability in the construction industry by making supply chains traceable, transparent, and immutable. However, executing smart contracts in the construction supply chain often requires exchanging real-world data, which cannot be accomplished by blockchain alone. Oracles are middleware agents that can capture and validate real-world information and feed it to a blockchain for the use of smart contracts. Smart construction objects (SCOs) may be used as trustworthy hardware oracles for blockchain applications in the construction industry (Lu, et al., 2021). SCOs are IoT devices that sense, process, and communicate information among construction resources such as men, machines, and materials. SCOs possess the properties of awareness, communicativeness and autonomy, which match well with the design patterns of blockchain oracles, can act as oracle node operators and enable data exchange between physical construction processes and blockchain technology. A Site Smart Contract (SSC) allows each stakeholder to request location data by providing his address (to check whether that stakeholder has access rights) and the SCOs of which he wants to know the location. An article (Lanko, Vatin, & Kaklauskas, 2018) considers the application feasibility of blockchain in logistics of construction materials, specifically in the process of manufacturing and delivery of ready-mixed concrete, through the usage of RFID technology. The goals are to eliminate problems of trust between participants, remove information barriers and avoid suit costs. The combination of RFID and blockchain technology can be used to monitor the delivery of goods, such as ready-mixed concrete, to avoid data errors and ensure quality control. RFID (Radio Frequency IDentification) is a method for automatically identifying objects, which consists in reading or writing data stored in so-called RFID tags with radio signals by RFID scanners. RFID tags are placed in the concrete mix during production and registered in the blockchain system to record all stages of production and operation. The blockchain system prevents manufacturers from using materials that do not meet standards and ensures transparency for purchasers by providing comprehensive data on the purchased products, among which time and place of production, and route details of the transportation vehicle through satellite. QR tags can be used as substitutes, but RFID tags are more preferable due to their higher information capacity and stronger protection against forgery. The introduction of RFID tag technologies in the ready-mixed concrete industry can accelerate workflow by allowing automatic and semi-automatic filling and reading of documents. RFID tags can also record data remotely and in motion. Combining RFID with GLONASS/GPS technology can enable real-time tracking of concrete delivery. A smart contract can allow the automated payment of materials and supplies at the moment of the delivery. The combination of RFID and blockchain technologies can eliminate the problem of trust between participants and reduce losses from human factor influence and intentional false information.

A further potential application of smart contracts pertains to leasing. The high cost of heavy equipment, including maintenance and repairing costs, often poses a financial challenge for construction contractors. To address this issue, leasing heavy equipment is a more cost-effective option for both large and small contractors to reduce their expenses on construction projects. Conventional leasing processes are time-consuming and inefficient, involving lengthy negotiation cycles, insurance quoting procedures, burdensome financing applications, and reams of paper documents that need to be signed and maintained (Wang, Wu, Wang, & Shou, 2017). In contrast, blockchain-based leasing of heavy equipment offers a more efficient, secure, and cost-effective solution for both the manufacturer and the construction contractor. IBM's blockchain platform (IBM, 2016) was used to record and manage the leasing process for a crane. The process starts with the manufacturer recording the crane's identity on the blockchain system. A prospective construction contractor chooses the crane they want to lease, and the crane's identity is then registered on the leasing blockchain to record transactions over broadly distributed computer networks. The construction contractor also chooses their insurance options. Payment details are included to pay for the lease and insurance, and payments for training, maintenance, and repairing services will be covered automatically. All of the above processes take a matter of minutes to complete.

6.3 Surety Bonds

Suretyship is a legal agreement between three parties, where the surety guarantees to the obligee that they will be responsible for any debt, default, or misconduct committed by the principal. This agreement is typically documented in writing. In the United States, several laws mandate that contractors provide surety bonds to protect public entities and suppliers on most federal, state, and local public construction projects. At the discretion of the project owner, surety bonds may also be required for private contracts. Obligees may elect to use a bond form promulgated by organizations, institutes or committees from the relative industry, develop a bond form of its own (possibly under the assistance of legal counsel) or let the surety of the low-bid contractor provide its own bond form. Every bond form should delineate the parties to the bond, the penal amount, contractor’s obligations and a reference to the contract and to the terms of reference. The contractor has to pay a risk premium to a security company which in turn guarantees to reimburse the owner for any damages incurred if the contractor fails to fulfill the contract obligations. This arrangement offers financial protection to project owners and ensures that contractors complete their work as agreed. Following is a list and a description of the main types of surety bonds (Russell, 2000).

- Bid bond: its purpose is to ensure that the contractor will enter into the contract at the bid price. If the contractor fails to enter into the contract without a valid legal excuse, the surety may be held liable for the difference between that contractor’s bid and the next qualified tenderer’s bid, or the penal sum of the bid bond, whichever is less. The "actual damages" type of bid bond requires the surety to reimburse the obligee for all costs and damages caused by the contractor's failure to enter into the contract and post any other required bonds. The contractor pays a premium to the surety for the issuance of the bond. The premium is typically a percentage of the bid amount (between 5% and 10%) based on factors such as the contractor's creditworthiness, experience, and the level of risk involved in the project.
- Performance bond: its purpose is to ensure that the contractor will complete the project and perform its obligations in accordance with the terms of the contract. In case of breach of contract, the bond provides financial protection to the owner, who can make a claim on the bond to recover any losses. The properly executed performance bond should be delivered to the obligee after the contractor's bid has been accepted and before the commencement of any work on the project. The bond usually covers an amount that ranges from 4% to 20% of the total value of the project to

completion with an annual cost between 1 and 5% of the remaining work to do depending on reliability and capacity of the contractor. In the construction sector default rates are less than 1% and performance bonds add an average of 1,5% to the cost of every construction project (Kraft, Park, & Gransberg, 2014).

- Payment bond (also known as labor and material bond): its purpose is to ensure that a contractor will pay all of its subcontractors, material suppliers, and laborers in accordance with the terms of their contracts for the labour or service that they perform and the material they furnish. If the contractor fails to make the required payments, the bond provides financial protection to the affected parties, who can make a claim on the bond to recover any unpaid amounts. Both public and private projects usually require both a performance bond and a payment bond. Typically, the surety is liable for an amount which must be not less than 50% of the contract price (Reynolds, 2021). The premium that the contractor pays to the surety for the issuance of the bond can range from 1% to 3% of the contract value. Payment bonds also protect private owners against mechanic’s liens. A mechanic's lien is a legal claim made by a contractor, subcontractor, or supplier against the owner’s property where they have provided labour, materials, or services, but have not been paid for their work. The lien may also give the claimant the right to force a sale of the owner’s real property to recover the debt. The protection of a payment bond alleviates the owner from the risk that potential lien claimants may file a mechanic’s lien. The use of payment bonds is mandatory by law in case of public projects, since subcontractors and suppliers cannot file for mechanic’s liens against public property and without the protection of a payment bond, they would be left with no remedy to collect what they're owed by the contractor.

- Advance payment bonds protect the employer for the full amount advanced to a contractor from the risk that it defaults on the agreement and fails to provide goods and services to an equivalent value. In international construction projects, it is rather standard that the contractor starts the project just after an advance payment has been paid by the employer, which is generally between 5% and 10% of the contract value. Such bonds usually contain a reduction clause, whereby the amount of the bond reduces in accordance with monthly certificates until the certified value of work done exceeds the advance payment (Norton Rose Fulbright, 2010). On average, the cost of an advance payment bond ranges from 1% to 5% of the bond amount.
- Maintenance bond: its purpose is to ensure that contractors will return to completed projects to perform maintenance work within a guarantee period. Although they are estimated to account for less than 5% of the total annual bonding volume, maintenance bonds are required in some areas of public construction such as highways and co-generation projects. Maintenance bond claims can be some of the most expensive for the surety to investigate and may result in lengthy litigation.
- Retained percentage bond. Retention refers to the total amount held back from the progress payments to ensure that the client is protected in case the contractor does not correct the defects at project completion, or during the warranty period, as usually established by contractual terms. Typically, only a portion of the retention is paid to the contractor upon completion of the project,

with the remaining amount paid after the warranty period has ended. A Retained Percentage Bond may replace the cash retained so as to have the contractor’s cash flow improved by allowing for total progress cash payments. This type of bond allows the owner to be reimbursed by the surety company for a percentage of the contractor payments in the event that the contractor's work results in defects or liabilities, and the surety company typically charges a nominal fee for this service. Generally, the cost of a retained percentage bond is between 1% to 3% of the bond amount (typically 5 or 10% of the contract amount).

Surety bonds constitute a fundamental part of the contract management process and provide financial protection for both contractors and owners. They come, however, with a set of issues. The reliance on intermediaries to manage the bond issuance and claim process imposes a financial burden on contractors and owners. Contractors are required to pay a premium to obtain a surety bond, which can range from 1% to 15% of the bond amount. Owners are forced to engage with their own intermediaries to manage the bond claim process, incurring in additional costs. Additionally, the involvement of intermediaries increases the chance of errors and does not protect the parties from the third-party risk of default. Smart contracts can be used to eliminate the need of intermediaries, reduce costs and streamline the claim process, by automatically enforcing the terms of the contract. Both contractors and owners would benefit from it. To overcome the use of bid bonds, owners and contractors may agree to use a smart contract that includes a clause imposing that, if the awarded contractor fails to enter into the contract without a valid legal excuse, an amount equal to the difference between the contractor’s bid and the next qualified tenderer’s bid shall be transferred to the owner’s wallet address.

In place of the retained percentage bond, the same solution proposed by Ahmadisheykhsarmast, Sönmez Ferda Özdemir and Sönmez, Rifat and discussed in the previous subchapter can be employed. Or alternatively, retention amounts could remain embedded inside the smart contract, and paid only after the warranty period has expired and all complaints about the maintenance status have been satisfied. The protection that payment bonds provide to clients against mechanic's lien can be granted equally by the previously presented system proposed by Wang, Wu, Wang, & Shou. To replace advance payment bonds, a solution blocking the funds in the smart contract as long as the certified value of work done exceeds the advance payment may be implemented.

6.3.1 Letter of credit

Procurement is an important part of construction projects. Its costs can reach up to 40-45% of total project cost (Agapiou, Flanagan, Norman, & Notman, 1998). Buyers and sellers in international trades have concerns regarding transactions, mostly because they did not have a commercial relationship in the past and are geographically separated. This is why letter of credit use in construction procurement is a common practice particularly for international trades in order to build trust between buyer and seller. The traditional process involves several steps that rely heavily on communication between sellers, buyers and banks as intermediaries. A letter of credit (L/C) is a written commitment from a bank or financial institution to pay a certain amount of money to the beneficiary, typically a seller or supplier, on behalf of the buyer or applicant, assuming the seller satisfies certain specified requirements. Typically, specific documents, such as a bill of lading or an invoice, that demonstrate that the goods or services have been shipped or delivered in accordance with the terms of the contract, must be presented. The letter of credit acts as a guarantee to the seller that he will be paid for their goods or services. After buyer and seller make an agreement concerning the procurement items, type and quantities, delivery schedule, quality standards and so on, the buyer applies to an issuing bank in order to start L/C procedures. The seller uses his representative bank as an advising bank. The issuing bank issues the letter of credit to the advising bank. After the seller ships the materials, the bill of lading is received by the advising bank and sent to the issuing bank, which will investigate it and then send the payment to the advising bank, to be forwarded to the seller.

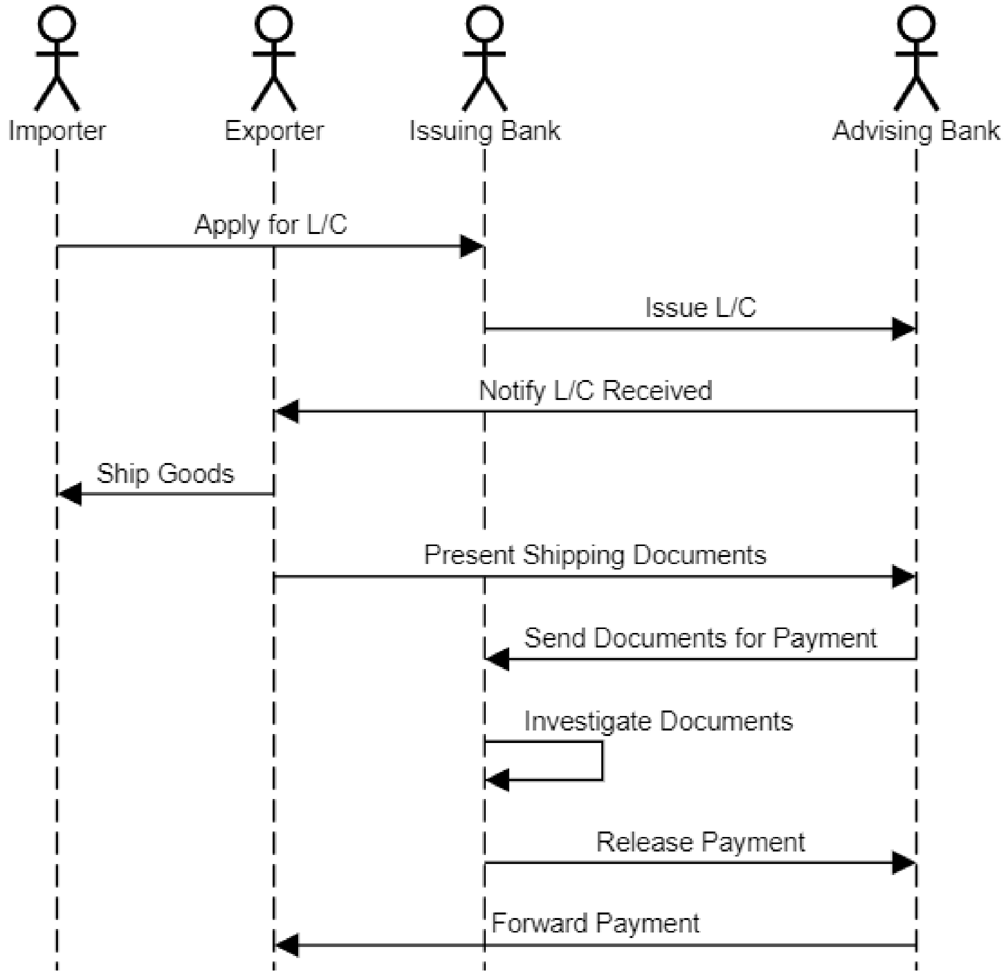


Figure 4 - Sequence diagram of the letter of credit process

The current system brings along some issues: the bill of lading may be forged to claim the consignment fee, the letter of credit may be forged to defraud the exporter, and fees of intermediary banks increase costs (Kumar, et al., 2022). The use of smart contract technology in construction procurement can significantly reduce costs, accelerate the process, minimize intermediary involvement, and overcome lack of trust between buyers and suppliers. A framework that uses smart contracts to ensure secure and automated payment for construction procurement material and equipment supply is proposed (Uysal, Ahmadisheykhsarmast, & Sonmez, 2022). The smart contract enables the payment to be blocked in such a way that is made inaccessible to the parties and triggered automatically once the specified conditions are met. Physical control over delivered items is possible with IOT devices, which significantly reduces the risk of buyer and makes procurement more reliable. Payments for material are made in two stages. Initial payment is deducted from the blocked amount and transferred to the supplier's wallet address at the time of the shipment (automatically through the use of RFID tags) and final payment is made once the material is on site and inspected successfully. The real time shipment status could be provided by the GPS data stored in the blockchain. The purchaser deposits a sufficient level of funds into a smart contract, which will remain in escrow until he confirms satisfaction with the quality of the materials, or initiates a dispute. In this last instance, it's essential that a dispute resolution application was designated ex-ante, as will be discussed in the following section.

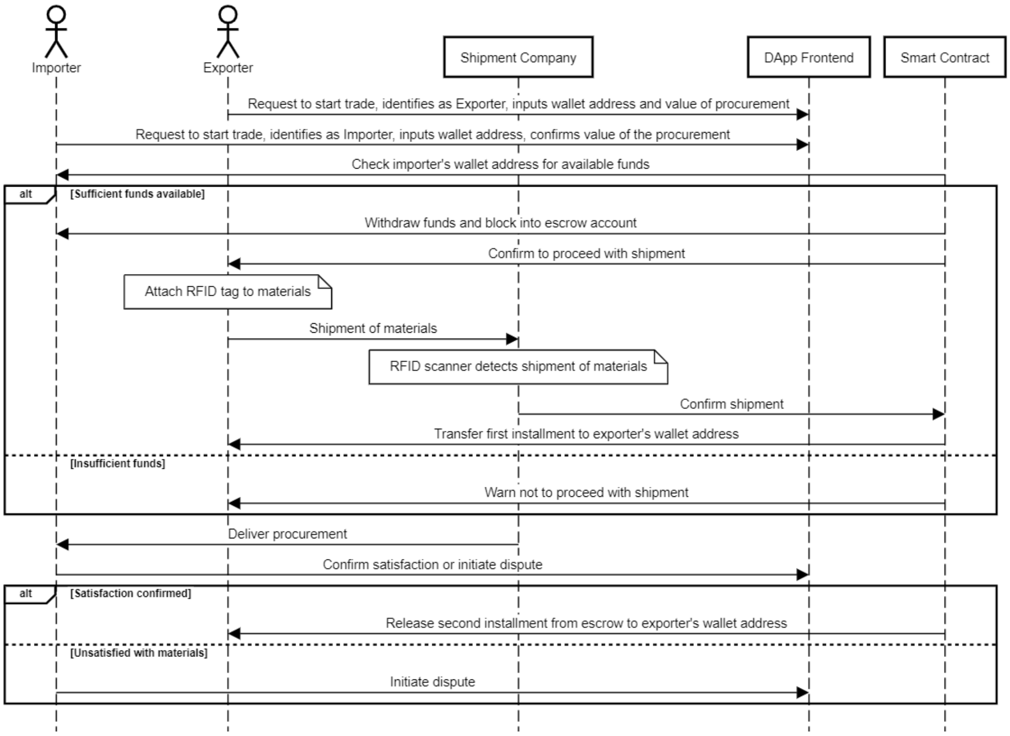

Importer
Exporter
Shipment Company
DApp Frontend
Smart Contract
Request to start trade, identifies as Exporter, inputs wallet address and value of procurement
Request to start trade, identifies as Importer, inputs wallet address, confirms value of the procurement
Check importer's wallet address for available funds
alt
[Sufficient funds available]
Withdraw funds and block into escrow account
Confirm to proceed with shipment
Attach RFID tag to materials
Shipment of materials
RFID scanner detects shipment of materials
Confirm shipment
Transfer first installment to exporter's wallet address
[Insufficient funds]
Warn not to proceed with shipment
Deliver procurement
Confirm satisfaction or initiate dispute
alt
[Satisfaction confirmed]
Release second installment from escrow to exporter's wallet address
[Unsatisfied with materials]
Initiate dispute

Figure 5 - Sequence diagram of a smart contract for international trade

6.4 Claims and disputes

A change order is a written agreement that allows for modifications, additions, deletions, or alterations to the work described in the contract documents at the time of opening bids. The change order may alter the contract price, schedule of payments, completion date, or the plans and specifications. Change orders are the only legally accepted method to alter the contract provisions after the contract has been awarded. A price change may be involved in a change order. However, this price change does not necessarily benefit the contractor and may also result in a cash credit for the owner or there might be no price change at all. It is standard practice in construction contracts to allow the owner the right to request changes in the work after the contract has been signed and during the construction period. The proposal should recognize the contractor's right to include overhead and profit percentages in change order estimates and in time and material change order billings. Contractors, too, can request a change order. This typically happens due to internal factors, such as late start in operations, inadequate resources, subcontractors and suppliers' failures, and external factors, such as market conditions, unforeseen site conditions, bad weather, labor disputes, etc. While the term change order refers to a bilateral agreement between the owner and the contractor to effect a change in the terms of the contract, a unilateral change order is referred to as change directive. Unilateral change orders may arise when the requested change is deemed necessary to avoid an emergency situation that could cause harm to people or property and are intended to expedite issuance of a change order. The change directive must be replaced by a regular bilateral change order that addresses the effect of the change on contract cost and time before payment can be made to the contractor. A change order should include the following information: description of the change, reason for change, new contract price and new contract time. Disagreements can surface around the topics of financial compensation and effects of the change on the construction schedule. If the contractor believes that the change order is unfair or not in compliance with the contract terms, they may request a negotiation or dispute resolution process to address the issue. A major source of construction disputes are constructive changes. Issued by the contractor when he thinks that certain acts or failure to act by the owner increase his cost and time of performance, a constructive change is aimed at obtaining extra compensation for performing the work. The owner frequently disagrees that a change in the contract requirements has occurred. Most of the disputes concerning constructive changes center around the interpretation of the plans and specifications (Fisk & Reynolds, 1988). Change orders may be issued also to adjust the contract schedule in the event of delays. What kinds of delays will justify an extension of time for the contractor depends on the provisions of the contract. The contractor is granted an extension in case of serious occurrences outside of his control, such as acts of God, labor disputes, those listed as excusable in the contract documents, and delays caused by the owner. Contractors must notify the owner in writing of any delay over which the contractor has no control, within a specified period of time (the notice provision), and must provide specific details about the cause of the delay, including times, dates, and supporting data. The notice shall be given as soon as practicable and not later than 28 days after the contractor became aware, or should have become aware, of the event or circumstance. Failing to do so jeopardizes the contractor's chances of obtaining an extension of time and discharges the employer from all liability in connection with the claim. Within 42 days after the contractor became aware of the event, he shall send a fully detailed claim (FIDIC, 1999). Internal sources of delay, such as inadequate project planning and scheduling, delayed procurement of material and equipment, poor workmanship, and equipment breakdowns are non-excusable and the contractor should shoulder the potential economic consequences, namely the monetary reimbursement of liquidated damages. Instead of determining the actual damages suffered by the owner if the contractor fails to complete the work within the time specified by the contract, which would be a long and complex process, it is common practice to write in the agreement the fixed sum of money for each calendar day of delay that the contractor owes to the owner, known as liquidated damages. These delay damages shall be the only damages due from the contractor for such default (sole remedy) and their total amount shall not exceed the 10% of the contract price. These damages shall not relieve the Contractor from his obligation to complete the works, or from any other duties, obligations or responsibilities which he may have under the contract (Coiro, 2022). The amount of liquidated damages set forth by the Contract can be adjusted (reduced) by the tribunal in case the penalized obligation has been partially performed or in case the amount of liquidated damages appears to be blatantly excessive (Article 1384 Civil Code). Disputes between whether a delay was excusable or not and around the amount of liquidated damages frequently arise. Disputes are a growing problem to the point that rarely a project ends with no dispute over differences between the parties. Thus, there is a high need to focus on claim prevention and dispute resolution. Incorporating proactive or preventive contracting techniques and dispute prevention processes in business relationships and contracts can help avoid costly and disruptive litigation. The disadvantages of litigation include loss of control, long and expensive processes, lack of expertise, public exposure, uncertainty, and disruption of business relationships. Proactively agreeing on a dispute processing system at the beginning of a business relationship brings on many advantages (Groton & Haapio, 2007). The beginning of the relationship is the best time for the parties to discuss this topic since there is an atmosphere of business-like cooperation and no disputes have arisen. Including this subject as an element in the negotiations helps gather more details about the other party's attitude and may lead to questioning whether to enter into the deal at all. It can also help create a satisfying, constructive and collaborative business relationship. Overall, agreeing on a dispute processing system early on can help create and preserve continuing business relationships. A number of causes of disputes in projects have been presented in literature. The rationale behind the efforts to identify the sources of disputes has been the premise that if the origins of the "illness" can be identified, ways to "cure" the industry from unnecessary litigation can be developed (Pena-Mora, Sosa, & McCone, 2003). Vorster (1993) cites four causes behind conflict in the construction industry: incomplete scope definition, inappropriate contract type, poor communication and uncertainty. They summarize sources of dispute such as errors in design, workmanship, tendering pressures, payment delays, quality and performance issues, orders to accelerate work, inappropriate contract type, contract documents and delivery system, misunderstandings, unrealistic expectations, negligence, differing site conditions, social, economic and political changes… Incompatible objectives and interests between the parties cause divergent interpretations of contract documents, terms and conditions leading to frequent disagreements. Goals that contractors and owners share are quite limited, examples are completing the project within budget and schedule, reduce the risk of liability and accidents, and minimize litigation. The remaining objectives are misaligned and foster the development of conflicts and disputes. The owner wants to maximize product quality, flexibility, capacity while minimizing operating and maintenance costs, disruptions and downtime. The contractor aims at achieving the greatest profit meeting the minimum requirements of the scope of work by means of limited expenditure on material resources and employment. What results is that conflicts are an intrinsic aspect in the relationship between owner and contractor and make it impossible to develop a unique theory on dispute prevention. Project managers should assess the specific project characteristics and establish a joint, creative, and effective approach to deal with and resolve conflicts before they can lead to disputes. In an industry that is excessively adversarial and concerned with disputes and litigation, costs soar due to excessive and expensive lawsuit, and profitability, productivity and quality struggle. In such a scenario, no one ultimately wins. It is important to seek and pursue alternatives to litigation, like alternative dispute resolution (ADR). ADR includes arbitration, mediation, mini-trials, dispute review boards, and other means of dispute resolution that do not involve the litigation process. Arbitration is defined as the reference of a dispute to one or more impartial persons for final and binding determination, and it is a more informal and economical process than a court proceeding. Many construction contracts provide for arbitration as the method by which disputes will be resolved, and arbitration clauses in contracts typically cover all claims, disputes, and other matters arising in relation to the contract or its breach. The general arbitration procedure is well-established and involves one party making a written demand to the other party for arbitration, selecting an impartial board of one or three arbitrators with knowledge and experience in the relevant field, conducting a hearing where each side can present evidence and witnesses, and receiving a written award from the arbitrators within a reasonable period of time. The arbitrators' authority exists only by virtue of the agreement between the parties, and their decision is binding. The award can be confirmed by a court and enforced like any other court judgment. While there is no appeal of the arbitrator's findings, the amount of the award can be challenged under certain circumstances, but this is rarely successful. Mediation is a less formal method of dispute resolution than arbitration in the construction industry. It involves a neutral third party, a mediator, who hears each party's positions and offers a suggested settlement that the parties can accept, reject, or use as a basis for negotiation, but lacks the power to impose a decision. Mediation is voluntary and can be provided for by contractual agreement or mutual consent. A common clause in construction contracts requires the parties to mediate disputes before escalating to other forms of dispute resolution. A mini-trial is an abbreviated trial overseen by a so-called judge or referee selected by both sides to the dispute. After both sides present their case, the judge conveys his findings to the parties and attempts to reach a settlement. If no settlement is reached, the parties move on to the next level of dispute resolution. Dispute Review Boards (DRBs) are formed before construction work begins, with one board member chosen by the contractor, another chosen by the owner, and a third chosen by the two board members to serve as chairman. The recommendations of the board are not binding but provide expert opinions from a disinterested impartial panel of experts. The aim is to avoid legal action.

6.4.1 Smart arbitration

The recent development of smart contracts has particularly significant implications for traditional contract law and dispute resolution mechanisms. New models of dispute resolution may be needed to handle these new types of contractual relationships. Here is where "smart arbitration" could step in to replace conventional judicial dispute settlement processes. Traditional arbitration systems can be slow, expensive, and often opaque, leading to a lack of trust in the process and its outcome. Traditional arbitration models are also inadequate to handle the unique characteristics of smart contract conflicts. On the other hand, smart arbitration offers a faster, cheaper, and more reliable way to resolve conflicts. Decentralized arbitration solutions leverage the power of blockchain technology and a network of independent arbitrators to create a decentralized and transparent system that removes intermediaries and enables parties to resolve their disputes without relying on a central authority. Smart contracts call for redress mechanisms that enable dispute settlement without the necessity of leaving the digital world or bringing legal action before an ordinary court. The ability of smart arbitration to directly enable the sharing of documents and pleadings using the blockchain as a verification mechanism is one of the main advantages over conventional arbitration. Close cooperation between lawyers on the one side and computer, mathematical and cryptography experts on the other side is required. Given the self-enforced nature of smart contracts, it is crucial that the process for conflict settlement gets specified up front

(Durovic, 2018). To address the upsurge of disputes in cross-border e-commerce, online dispute resolution (ODR) has emerged as a method of conflict management alternative to public courts. However, ODR faces challenges in enforcing decisions. Smart contracts could also be employed to provide a solution for the problem of enforcing ODR decisions (Koulu, 2016). In this regard, a way to resolve disputes efficiently and fairly may consist in ODR built into smart contracts. The ODR clause can operate like an escrow arrangement and create a role for a third party, the dispute resolution service provider, who can help determine the appropriate path forward in the case of a dispute (Schmitz & Rule, 2019). As it happens in the case of traditional contracts, a blockchain-based smart contract should include a dispute resolution clause, or arbitration clause. Typically, this clause would specify a procedure for the parties to settle their disputes and state the parties' consent that the outcome of the dispute resolution process can be automatically enforced on the blockchain. As previously stated, contractors can request change orders in case of differing and inadequate site conditions. Wang et al. (2017) presented a simple example of a smart contract developed on the Ethereum blockchain platform, which states that if the temperature of a construction site is higher than 40 degrees centigrade, the client will pay a certain amount of dollars to the construction contractor. It may be better if, instead of forcing an automated payment to compensate the contractor, thus impacting the owner’s cash flow, the smart contract postponed the deadline date by one day for each of the days that the temperature of the construction site was higher than 40 degrees centigrade, or that other measurable and detectable conditions not allowing the contractor to perform occurred.

In recent years, many technology startups have emerged with the aim to manage blockchain dispute resolution processes tailored to smart contracts. They have developed blockchain-based arbitration platforms, each with its own unique features and advantages, that allow parties to enter into smart contracts and submit related disputes to arbitration by a crowdsourced, decentralized, and anonymous decisionmaker (jury) that is economically incentivized (using game theory principles and cryptocurrency rewards) to reach consensus and issue a decision (Bergolla, Seif, & Eken, 2022). Kleros (Ast, et al., 2019) is a blockchain-based decentralized dispute resolution system that uses crypto-economics-based incentives to provide fast, affordable, and transparent arbitration services. Two parties can submit a claim to a crowdsourced jury, and Kleros acts as an ad hoc decentralized arbitration system to ensure the fairness of the jury. Jurors are randomly selected from a pool of users who hold the Kleros token, and they are incentivized to make fair and accurate decisions by earning fees and rewards for their participation. Jurors are rewarded for making correct decisions and penalized for making incorrect decisions. The "stochastic drafting" to select jurors for each dispute ensures that they are selected randomly from a large pool, which makes it extremely difficult for a single entity to coordinate their votes and manipulate the outcome of a dispute. Kleros has been developed as reusable components that can be used in other smart contracts that become "arbitrable" (Perdrisat, 2021). OpenLaw is a platform for creating and executing legal agreements using smart contracts. Parties that transfer assets via a blockchain or enter into blockchain-compatible agreements will inevitably get into disputes, and they will need tools to manage disagreement in a low-trust environment. The blockchain ecosystem needs several baseline tools, including smart contracts to manage an arbitration procedure to ensure the enforceability of any arbitral awards, and solid reputation systems to help the community select arbitrators to resolve disputes. OpenCourt (OpenLaw, 2018), a dispute resolution system that uses a decentralized network of arbitrators to settle disputes, is an initiative developed by OpenLaw. It proposes a form of private arbitration conducted on the blockchain, which is an analogue of existing off-chain and offline private arbitration frameworks (Herian, 2018). Any smart contract can rely on the OpenCourt system to easily incorporate a dispute resolution procedure. Once configured, OpenCourt will send the smart contract notice of a confirmed dispute once invoked. The smart contract will then transfer any identified digital assets to a virtual escrow account, thus locking these assets until an arbitral decision is reached. The dispute resolution procedure can be accessed via parties to the contract and arbitrators through a basic user interface. In summary, OpenCourt is a blockchain-based arbitration system that could provide a globally accessible "online court" where people have an equal opportunity to receive fast, low-cost, sophisticated, and transparent dispute resolution services online that could be integrated with existing judicial systems. The advantages of OpenCourt over traditional processes for dispute resolution are diverse. Differently from the conventional dispute handling routes, OpenCourt, is designed to be a low-cost system that can be accessed from anywhere in the world and not forcing parties to travel to a specific jurisdiction to resolve their dispute. Traditional dispute resolution can be slow, with cases often taking months or even years to resolve. OpenCourt, by contrast, is designed to be fast and efficient, with disputes resolved through a smart contract-based system that can operate in real-time. Finally, the decisions taken by the decentralized court and the decision making of the arbitrators are transparent, publicly auditable and can be reviewed by anyone. Other platforms are JUR, Aragon Network Jurisdiction and OpenBazaar. They are similar in many aspects but try to differentiate on unique juror-incentivization strategies, different levels of legal enforceability, and specialized tribunals. Certain procedural elements of the dispute, such as the number of jurors, a specialized subcourt, and a list of possible future remedies, must be specified upfront depending on the platform. It is often allowed to accompany the code-based smart contract with agreements expressed in natural legal language. Thus, when choosing on-chain resolution, the agreement is initiated by means of two components, a pre-coded smart contract and its natural language counterpart. Indeed, many smart contract disputes arise because the intentions of the parties and code draftings diverge (Buchwald, 2020). OpenCourt, for example, offers templates for drafting a natural language contract to supplement the Solidity code-based agreement. On the other hand, Kleros avoids the incorporation of a natural legal language contract and asks the plaintiff to later present proofs of communication with the defendant that show that the latter did not meet the agreed-upon obligations. In terms of dispute resolution initiation there is little variation among available platforms. In all cases, the dissatisfied party must use the application to trigger dispute resolution. Parties should upload a statement of facts, filling a simple form explaining the claim and why they believe that they are entitled to relief, together with any evidence suitable to best support an argument. The textbox should be filled as thoroughly as possible since it is usually not possible to provide further clarifications later and it will be the only basis for the jurors' verdict. The process of selecting jurors is similar to what happens in the United States, where, however, in most cases it is mandatory for eligible citizens to serve as jurors. Smart arbitration juror candidates instead self-volunteer by staking a deposit in the form of cryptocurrency. Jury selection is done randomly among all the users that staked the cryptographic token. This is also called randomized lottery. Kleros requires candidates to stake a cryptographic token called pinakion (PNK) in order to have the possibility of being drawn as jurors. The probability to be drawn as juror is proportional to the amount of tokens a user deposits in a subcourt. The higher the amount of tokens a user stakes, the higher the probability he will be drawn as juror (Ast, et al., 2019). The fact that the probability of selection in the lottery is directly proportional to the size of one's deposit renders null the action of a malicious party to create a high number of addresses to be drawn more often, get more votes thus controlling the system. Theoretically, a candidate may be drawn more than once for a specific dispute. The number of times a user is drawn for a dispute (called its weight) will define the number of votes he will get in the dispute and the amount of tokens he will win or lose as a result of his vote. OpenCourt instead uses an alternative method to this volunteer lottery: the parties should select a mutually agreed upon third-party arbitrator and input his Ethereum address. His decision on the matter will be final and binding. In the United States, *voir dire* is the process by which the judge, prosecutor, and defense attorneys determine a juror's eligibility to serve on the jury and identify potential biases or prejudices that could affect the fairness of the ruling. This ensures that the jury is composed of fair and impartial individuals who can make an objective decision based on the evidence presented in court and render a just verdict in the case. Decentralized dispute resolution systems generally do not use a *voir dire* process for screening jurors. Jur offers an exception in Hub virtual tribunals where jurors are requested some qualifications (to hold an engineering degree for example), which get reviewed by application administrators. Aragon offers disputants the possibility to obtain a pool of jurors with high positive reputation by paying a higher fee, which is proportional to the jury's total reputation. Users that are drawn as jurors will have access to the evidence for analysis and will vote a decision. They are also required to provide a justification for their decision. After all jurors have voted, the decision is produced. Users have two economic incentives for serving as jurors: collecting arbitration fees and token redistribution. Arbitration fees are payments from parties to all jurors as a compensation for the time and expertise invested in analyzing evidence and voting. Furthermore, they will gain or lose tokens depending on whether their vote was coherent with the rest. This financially incentivized majority-voting scheme imposes minority voters to forfeit their tokens in favor of the majority. This redistribution is based on the Schelling point principle, also known as the Focal Point principle. Thomas Schelling (1960) theorized in his book "The Strategy of Conflict" that people tend to converge on the same solution (called focal point) in situations where there is no clear reason to choose one option over another or where there is no communication between the parties involved. Based on this principle, assuming that jurors have the same incentives and access to the same evidence, they are expected to reach a similar verdict. If not, reasons lie in them not being properly qualified, not conducting a thorough review of the case, or being corrupted. A juror who chooses cases where he does not have the right expertise, who does not analyze the evidence carefully or who does not vote honestly is more likely to vote incoherently with others and, as a result, will suffer an economic loss. No on-chain platform currently asks jurors to rely on jurisdictional precedent. Appealing a court decision is a fundamental right recognized by most legal systems and most decentralized platforms allow the dissatisfied party the opportunity to appeal. In Kleros, decisions can be appealed several times. In each round, a new jury will be formed with twice as many jurors than the previous instance plus one. The appealing party will be required to make a new deposit in order to pay for arbitration fees. The cost of appeal is proportional to the number of jurors and increases steeply instance after instance, discouraging excessive appellate proceedings. Argon uses a different approach: each appeal requires double the reputational weight of the jury and thus double the price of the arbitration fees. The first appeal is called “Prediction Market” and all jurors on the Aragon Network are invited to partake. If a party is still dissatisfied, it can appeal to the Supreme Court, consisting of the nine jurors who hold the highest reputational ranking. Their ruling is final on the dispute. In case of appeals, the financial redistribution of the tokens for the jurors in previous rounds can be reversed according to the ruling of the higher court. The on-chain incentivized voting carries, however, some weaknesses and flaws. First of all, a majority vote does not necessarily lead to the correct legal result. Jurors may be incentivized to predict the popular opinion on the specific matter and vote accordingly, even knowing what the correct stance is, not to lose their cryptocurrency at stake. Most applications do not allow disputants and jurors to select the specific jurisdiction. In a scenario in which jurors come from different geographic regions, the risk for an arbitrator lies in the possibility of being penalized even if making the correct legal decision in its jurisdiction. This will make users wary to serve as arbitrators. Furthermore, the absence of precedents makes the decision-making process arbitrary and subjective. However, these issues could be addressed by including a choice-of-law provision to guide jurors in their decision-making process and allowing time to a resolution application to develop its own set of law precedents. Traditional voting systems do not resort to majority voting incentivization schemes. In the United States federal court system jurors receive a flat fee and have no financial incentive to vote with the majority. This ensures impartiality and allows judges and jurors to make decisions based purely on the law. On the other hand, a decentralized, anonymous method of juror voting cannot be conceived without an incentivized majority voting scheme to produce coherent votes. On a final note, artificial intelligence (AI) may assist fair and efficient dispute resolution for smart contracts by providing predictive analysis and quickly suggesting resolutions that may be subsequently entered into the blockchain (Schmitz & Rule, 2019).

Conclusions

The purpose of the thesis was to explore the integration of smart contracts into contract management and evaluate the associated benefits and challenges. The main findings indicate that smart contracts bring several advantages in contract management compared to other forms of traditional and electronic contracting. They arise from the self-enforceability of smart contracts and the underlying features of the blockchain. Decentralization and cryptographic mechanisms help to prevent the risk of data loss and malicious data manipulations. Thus, they allow to create a secure and reliable environment where to perform storage and record keeping of contract and project data. This helps create transparency and solve trust issues among the parties. Smart contracts provide an innovative and safe platform for automated execution of payments, such as progress payments, payments for materials and equipment supplies, retention payments, and liquidated damages. Smart contracts constitute innovative types of contractual arrangement and require a new way to deal with disputes, smart arbitration. This innovative framework makes the employment of traditional intermediaries, like attorneys, banks and insurance companies, redundant. The overall effect consists in reduced costs, delays, paperwork, and bureaucracy, which are often associated with traditional contract management processes, and increased efficiency, accuracy, security and transparency. However, the use of smart contracts in contract management presents several challenges, such as legal and regulatory barriers to the recognized acceptance of the use of smart contracts as legal contracts, security concerns, technology's complexity and the consequent need for the development of new skills and expertise, lack of standardization and interoperability, complexity in broadcasting off-chain real world information to the blockchain as to create fully autonomous procedures and resistance to the complete replacement of traditional methods of project and contract management. The findings of this thesis can be used to guide future research and development in the field of smart contracts and contract management. As blockchain technology and smart contracts continue to evolve, it is expected that more companies will embrace these innovations, leading to further learnings and significant improvements.

Bibliography

European Parliament. (2008). Regulation (EC) No 593/2008 of the European Parliament and of the Council of 17 June 2008 On the law applicable to contractual obligations (Rome I). *Official Journal of the European Union*.

Abeyratne, S., & Monfared, R. (2016). Blockchain ready manufacturing supply chain using distributed ledger. *International Journal of Research in Engineering and Technology*, 1-10.

Agapiou, A., Flanagan, R., Norman, G., & Notman, D. (1998). The changing role of builders merchants in the construction supply chain. *Construction Management and Economics*, 351-361.

Agostini, L. (2021). Blockchain and Smart Contracts: the EU's (lacking) view.

Agrawal, M., & Mishra, P. (2012). A Comparative Survey on Symmetric Key Encryption Techniques. *International Journal on Computer Science and Engineering*.

Ahmadisheykhsarmast, S., & Sonmez, R. (2020). A smart contract system for security of payment of construction contracts. *Automation in construction*.

Ahmadisheykhsarmast, S., Sönmez, F. Ö., & Sönmez, R. (2020). Smart contracts for contract management: a retention payment system. *Blockchain for cybersecurity and privacy*, 307-319.

Antonopoulos, A. M. (2014). *Mastering Bitcoin: Unlocking digital cryptocurrencies.* O'Reilly Media, Inc.

Antonopoulos, A. M., & Wood, G. (2018). *Mastering ethereum: building smart contracts and dapps.* O'reilly Media.

Arcadio, B., & Cuenco, J. (2016). Updated guidelines in the audit of procurement. *Commission on Audit.*

Arditi, D., & Chotibhongs, R. (2005). Issues in subcontracting practice. *Journal of construction engineering and management.*

Article 1384 Civil Code.

Ast, F., Bergolla, L., Braga, P., D'Agnillo, N., Deplano, R., Dimov, D., . . . Monegro, J. (2019). Dispute Revolution - The Kleros Handbook of Decentralized Justice. *Kleros.io.*

Atlam, H. F., Azad, M. A., Alzahrani, A. G., & Wills, G. (2020). A Review of Blockchain in Internet of Things and AI. *Big Data and Cognitive Computing*, 28.

Atlam, H. F., Azad, M. A., Alzahrani, A. G., & Wills, G. (2020). A Review of Blockchain in Internet of Things and AI. *Big Data and Cognitive Computing*, 28.

Azzi, R., Chamoun, R. K., & Sokhn, M. (2019). The power of a blockchain-based supply chain. *Computers & Industrial Engineering*, 582-592.

Back, A. (2002). Hashcash-a denial of service counter-measure.

Bahram, N. (1995). *The hypercube of innovation.* Res Policy.

Bashir, I. (2017). In I. Bashir, *Mastering Blockchain: A Deep Dive Into Distributed Ledgers, Consensus Protocols, Smart Contracts, DApps, Cryptocurrencies, Ethereum, and More.* Packt Publishing Ltd.

Bazzanella, D. (2021). *Blockchain and Cryptoeconomy course slides.*

Bentov, I., Lee, C., Mizrahi, A., & Rosenfeld, M. (2014). Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake. *Association for Computing Machinery*, 34-37.

Bergolla, L., Seif, K., & Eken, C. (2022). Kleros: A socio-legal case study of decentralized justice & blockchain arbitration. *Ohio St. J. on Disp. Resol.*

Best, R. d. (2022, November 18). *Number of cryptocurrencies worldwide from 2013 to November 2022.* Retrieved from Statista: https://www.statista.com/statistics/863917/number-crypto-coins-tokens/

Blockchain.com. (2023, February). *Network difficulty.* Retrieved from www.blockchain.com: https://www.blockchain.com/explorer/charts/difficulty

Bocek, T., Rodrigues, B. B., Strasser, T., & Stiller, B. (2017). Blockchains everywhere-a use-case of blockchains in the pharma supply-chain. *2017 IFIP/IEEE symposium on integrated network and service management (IM)* (p. 772--777). IEEE.

Bosco, A. (2018). Blockchain and the Uniform Electronic Transactions Act. *Bus. Law.*, 243-251.

Buchwald, M. (2020). Smart Contract Dispute Resolution: The Inescapable Flaws of Blockchain-Based Arbitration. *University of Pennsylvania Law Review*.

Bunz, B., Kiffer, L., Luu, L., & Zamani, M. (2020). FlyClient: Super-Light Clients for Cryptocurrencies. *IEEE Symposium on Security and Privacy (SP)*, (p. 928-946).

Buterin, V. (2014). A next-generation smart contract and decentralized application platform.

Buterin, V. (2018, April 1). *Meta: cap total ether supply at ~120 million*. Retrieved from Github: https://github.com/ethereum/EIPs/issues/960

Cahn, C. I. (1972). Contractors' Payment Bonds in Maryland. *HeinOnline*.

Callahan, C. C. (1955). Statutes of Limitation--Background. *The Yale Law Journal*, 130-139.

Cantamessa, M., & Montagna, F. (2016). *Management of innovation and product development.* Springer.

Chan, E. H., & Suen, H. C. (2005). Dispute resolution management for international construction projects in China. *Management decision*.

Clough, R. H., Sears, S. K., Segner, R. O., & Rounds, J. L. (2015). *Construction contracting: A practical guide to company management.* John Wiley & Sons.

Cohen, G. M. (1991). The Negligence-Opportunism Tradeoff in Contract Law. *Hofstra L. Rev.*

Coiro, D. (2022). Introduction to Contract Management. *Transport market liberalisation, tendering and competition. The role of contract management.*

Collins, B. (2012). Inquiry into construction industry insolvency in NSW. *NSW Government*.

Corbin, A. L. (1913). Discharge of Contracts. *The Yale Law Journal*, 513-530.

Corbin, A. L. (1944). The Parol Evidence Rule. *The Yale Law Journal*, 603-663.

Dai, H.-N., Zheng, Z., & Zhang, Y. (2019). Blockchain for Internet of Things: A Survey. *IEEE Internet of Things Journal*, 8076-8094.

De Filippi, P. (2015). *Smart contracts. Legal aspects.* Retrieved from P2P Foundation: https://wiki.p2pfoundation.net/Smart_Contracts

De Filippi, P., & Hassan, S. (2018). Blockchain technology as a regulatory technology: From code is law to law is code.

De Marco, A. (2011). *Project management for facility Constructions.* Springer.

Demogue, R. (1917). Validity of the theory of Compensatory Damages. *The Yale Law Journal*, 585-598.

Durovic, M. (2018). Law and Autonomous Systems Series: How to Resolve Smart Contract Disputes - Smart Arbitration as a Solution. Retrieved from https://blogs.law.ox.ac.uk/business-law-blog/blog/2018/06/law-and-autonomous-systems-series-how-resolve-smart-contract-disputes

Durovic, M. (2019). How to Resolve Smart Contract Disputes: Smart Arbitration As a Solution. In N. Aggarwal, H. Eidenmüller, L. Enriques, J. Payne, & K. v. Zwieten, *Autonomous Systems and the Law.* Beck C. H.

Dwork, C., & Naor, M. (1993). Pricing via processing or combatting junk mail. *Advances in Cryptology—CRYPTO'92: 12th Annual International Cryptology Conference* (p. 139-147). Springer.

Ellis, S., Juels, A., & Nazarov, S. (2017). Chainlink: A decentralized oracle network.

Fernández-Caramés, T. M., & Fraga-Lamas, P. (2018). A Review on the Use of Blockchain for the Internet of Things. *Ieee Access*, 32979--33001.

FIDIC. (1999). *Conditions of contract for construction.* Fidic Geneva.

FINMA. (2018, February 16). *FINMA publishes ICO guidelines.* Retrieved from FINMA: https://www.finma.ch/en/news/2018/02/20180216-mm-ico-wegleitung/

Fisk, E. R., & Reynolds, W. D. (1988). *Construction project administration.* Wiley New York.

Gaithuru, J. N., Bakhtiari, M., Salleh, M., & Muteb, A. M. (2015). A comprehensive literature review of asymmetric key cryptography algorithms for establishment of the existing gap. *2015 9th Malaysian Software Engineering Conference (MySEC).*

Gaussen, B. (2018). Adjudicate Today response to the proposal for “deemed” statutory trusts. *Adjudicate Today*.

Goldbloom, J. (1989). The General Conditions of the Contract. In J. Goldbloom, *Engineering Construction Specifications: The Road to Better Quality, Lower Cost, Reduced Litigation* (p. 31-142). Springer.

Groton, J., & Haapio, H. (2007). From reaction to proactive action: dispute prevention processes in business agreements. *IACCM EMEA Academic Symposium.* London.

Guegan, D. (2017). *Public Blockchain versus Private blockhain.*

Haber, S., & Stornetta, S. (1991). How to time-stamp a digital document. *Journal of Cryptology*, 99-111.

Hamledari, H., & Fischer, M. (2021). Construction payment automation using blockchain-enabled smart contracts and robotic reality capture technologies. *Automation in Construction*.

Hamledari, H., & Fischer, M. (2021). Role of blockchain-enabled smart contracts in automating construction progress payments. *Journal of legal affairs and dispute resolution in engineering and construction*.

Han, D., Kim, H., & Jang, J. (2017). Blockchain based smart door lock system. *2017 International conference on information and communication technology convergence (ICTC)* (p. 1165--1167). IEEE.

Hardjono, T., & Dondeti, L. R. (2005). *Security In Wireless LANS And MANS (Artech House Computer Security).* Artech House, Inc.

Herian, R. (2018). Legal recognition of blockchain registries and smart contracts. *EU Blockchain Observatory and Forum*.

Hill, W. M., & McCormack, M.-B. (2011). Pay-if-Paid Clauses: Freedom of Contract or Protecting the Subcontractor from Itself. *Constr. Law.*

IBM. (2016). *IBM Blockchain*. Retrieved from https://www.ibm.com/blockchain

IBM. (2022, November 20). Retrieved from ibm.com: https://www.ibm.com/topics/what-is-blockchain

Introduction to DApps. (2022). Retrieved from Ethereum: https://ethereum.org/en/developers/docs/dapps/

Johnston, J. S. (1996). The Statute of Frauds and Business Norms: A Testable Game-Theoretic Model. *University of Pennsylvania Law Review*, 1859–1912.

Joseph, P. D., Krishna, M., & Arun, K. (2015). Cognitive Analytics and Comparison of Symmetric and Asymmetric Cryptography Algorithms. *4th National Conference on Recent Trends in Information.*

Karantias, K., Kiayias, A., & Zindros, D. (2019). Proof-of-Burn.

Katz, J. (2010). *Digital Signatures.* Springer.

Kaur, R., & Kaur, A. (2012). Digital Signature. *2012 International Conference on Computing Sciences*, (p. 295-301).

Kerikmäe, T., & Rull, A. (2016). *The Future of Law and eTechnologies.* Springer.

Ketu File white papers. (2004). *Symmetric vs Asymmetric Encryption.* KetuWare, a division of Midwest Research Corporation.

Koulu, R. (2016). Blockchains and online dispute resolution: smart contracts as an alternative to enforcement. *SCRIPTed - A Journal of Law, Technology & Society*, 40-69.

Kraft, E., Park, H., & Gransberg, D. D. (2014). Performance bond: Cost, benefit, and paradox for public highway agencies. *Transportation Research Record*, 3-9.

Kronman, A. T. (1985). Contract law and the state of nature. *The Journal of Law, Economics, and Organization*, 5-32.

Kshetri, N. (2018). lockchain's roles in meeting key supply chain management objectives. *International Journal of Information Management*, 80-89.

Kumar, A., Abhishek, K., Nerurkar, P., Ghalib, M. R., Shankar, A., & Cheng, X. (2022). Secure smart contracts for cloud-based manufacturing using Ethereum blockchain. *Transactions on Emerging Telecommunications Technologies*.

Lamport, L., Shostak , R., & Pease, M. (1982). The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 382-401.

Lanko, A., Vatin, N., & Kaklauskas, A. (2018). Application of RFID combined with blockchain technology in logistics of construction materials. *MATEC Web of Conferences.*

Larimer, D. (2014). Delegated proof-of-stake (dpos). *Bitshare whitepaper*, 85.

Latham, S. M. (1994). Constructing the Team:Final Report of the Government Industry Review of Procurement and Contractual Arrangements in the UK Construction Industry.

Leung, H. T. (2018). Smart contracts - can code ever be law? *DIGITAL ECONOMY UPDATE*.

Lin, F. (2010). *Cryptography's Past, Present, and Future Role in Society.*

Lu, W., Li, X., Xue, F., Zhao, R., Wu, L., & Yeh, A. G. (2021). Exploring smart construction objects as blockchain oracles in construction supply chain management. *Automation in construction*.

Lundqvist, T., De Blanche, A., & Andersson, R. H. (2017). Thing-to-thing electricity micro payments using blockchain technology. *2017 Global Internet of Things Summit (GIoTS)* (p. 1-6). IEEE.

Lyons, T., Courcelas, L., & Timsit, K. (2019). Legal and regulatory framework of blockchains and smart contracts. *Thematic report for The European Union Blockchain Observatory and Forum.*

Macaulay, M., & Summerell, T. (2019). UK: Project Bank Accounts: Making Payment Fair.

Manu, E., Ankrah, N., Chinyio, E., & Proverbs, D. (2015). Trust influencing factors in main contractor and subcontractor relationships during projects. *International Journal of Project Management*.

Mason, J. (2017). Intelligent contracts and the construction industry. *Mason, Jim. "Intelligent contracts and the construction industry." Journal of legal affairs and dispute resolution in engineering and construction*.

Memoria, F. (2022, July). *Does Ethereum Have a Supply Cap?* Retrieved from Cryptoglobe: https://www.cryptoglobe.com/latest/2022/07/ethereum-supply-cap/

Mudge, N. (2018, October 31). *ERC-1538: Transparent Contract Standard*. Retrieved from Ethereum Improvement Proposals: https://eips.ethereum.org/EIPS/eip-1538

Nadikattu, A. K. (2018). Iot and the Issue of Data Privacy. *International Journal of Innovations in Engineering Research and Technology*, 23-26.

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.*

Nist, C. (1992). The digital signature standard. *Communications of the ACM*, 36-40.

Nithya, S., & Raj, E. (2014). Survey on Asymmetric Key Cryptography Algorithms . *Journal of advanced computing and communication technologies*.

Norton Rose Fulbright. (2010). Bonds and guarantees .

Nyffenegger, R. (2018). *Scaling Bitcoin.*

O'Dwyer, K. J., & Malone, D. (2014). Bitcoin Mining and its Energy Footprint. *25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)* (p. 280-285). IET.

Office of Contracting and Procurement. (2023). What are IFBs and RFPs? *Rowan University*.

Omopariola, E. D., Windapo, A. O., Edwards, D. J., & Chileshe, N. (2022). Attributes and impact of advance payment system on cash flow, project and organisational performance. *Journal of Financial Management of Property and Construction*.

OpenLaw. (2018, October 18). *OpenCourt: Legally Enforceable Blockchain-Based Arbitration.* Retrieved from https://media.consensys.net/opencourt-legally-enforceable-blockchain-based-arbitration-3d7147dbb56f

Palliyaguru, R. S., Amaratunga, R., & Rameezdeen, R. (2006). Financing contractors in developing countries: Impact of mobilization advance payment.

Pena-Mora, F. A., Sosa, C. E., & McCone, D. S. (2003). *Introduction to construction dispute resolution.*

Perdrisat, S. (2021). Case study of sociotechnical imaginaries in the making: Kleros decentralized dispute resolution protocol.

Polyviou, A., Velanas, P., & Soldatos, J. (2019). Blockchain Technology: Financial Sector Applications Beyond Cryptocurrencies. *The 3rd Annual Decentralized Conference on Blockchain and Cryptocurrency.*

Rainelli, P. (2021). Business Law course slides.

Raskin, M. (2016). The law and legality of smart contracts. *Geo. L. Tech. Rev.*, 305-340.

Reynolds, R. J. (2021). Handling Surety Performance Bond And Payment Bond Claims.

Robinson v Harman, 363 (Court of Exchequer Chamber January 18, 1848).

Rogaway, P., & Shrimpton, T. (2004). Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. *International Workshop on Fast Software Encryption* (p. 371-388). FSE 2004: Fast Software Encryption.

Rose, K., Eldridge, S., & Chapin, L. (2015). The internet of things: An overview. *The internet society (ISOC)*, 1-50.

Rühl, G. (2020). Smart (Legal) Contracts, or: Which (Contract) Law for Smart Contracts? In B. Cappiello, & G. Carullo, *Blockchain, Law and Governance* (p. 159–180). Springer.

Russell, J. S. (2000). *Surety bonds for construction contracts.* American Society of Civil Engineer.

Sankar, L. S., Sindhu, M., & Sethumadhavan, M. (2017). Survey of consensus protocols on blockchain applications. *4th international conference on advanced computing and communication systems (ICACCS)* (p. 1-5). IEEE.

Satish, D., & Shah, P. (2009). A study of public private partnership models. *IUP Journal of Infrastructure*, 22-37.

Savelyev, A. (2017). Contract law 2.0:‘Smart’contracts as the beginning of the end of classic contract law. *Information & communications technology law*, 116-134.

Schelling, T. (1960). *The Strategy of Conflict.* Harvard University Press.

Schmitz, A., & Rule, C. (2019). Online dispute resolution for smart contracts. *Journal of Dispute Resolution*.

Senate Economics References Committee. (2015). Insolvency in the Australian construction industry.

Shifferaw, Y., & Lemma, S. (2021). Limitations of proof of stake algorithm in blockchain: A review. *Zede Journal*, 81-95.

Siegel, D. (2018). Understanding the dao attack.

Sigalov, K., Ye, X., König, M., Hagedorn, P., Blum, F., Severin, B., . . . Groß, D. (2021). Automated Payment and Contract Management in the Construction Industry by Integrating Building Information Modeling and Blockchain-Based Smart Contracts. *Applied Sciences*.

Slovenko, R. (1998). Duty to minimize damages. *The Journal of Psychiatry & Law*, 579-594.

Sonmez, R., Ahmadisheykhsarmast, S., & Güngör, A. A. (2022). BIM integrated smart contract for construction project progress payment administration. *Automation in Construction*.

Sriman, B., Ganesh Kumar, S., & Shamili, P. (2021). Blockchain technology: Consensus protocol proof of work and proof of stake. *Intelligent Computing and Applications: Proceedings of ICICA 2019* (p. 395-406). Springer.

Stock, J. R., & Boyer, S. L. (2009). Developing a consensus definition of supply chain management: a qualitative study. *International Journal of Physical Distribution & Logistics Management, 39*(8), 690-711.

Subramanya, S., & Yi, B. (2006). Digital signatures. *IEEE Potentials, 25*(2), 5-8.

Supplier Journey. (2023). *Contract Notice*. Retrieved from Supplier Journey: https://www.supplierjourney.scot/supplier-journey/prepare/understand-market/public-contracts-scotland-pcs/contract-notice

Szabo, N. (1996). Smart contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought*.

Tabora, V. (2018). Databases and blockchains, the difference is in their purpose and design. *Hentet*, 2019.

Toyoda, K., Mathiopoulos, P. T., Sasase, I., & Ohtsuki, T. (2017). A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain. *IEEE Access*, 17465-17477.

UK Government commercial function. (2021). *Competitive dialogue and competitive procedure with negotiation.*

UK Office of Government Commerce. (2007). Guide to best “fair payment” practices.

Uysal, F., Ahmadisheykhsarmast, S., & Sonmez, R. (2022). A Smart Contract Framework as an Alternative Method for Letter of Credit Use in Construction Procurement. *The Twelfth International Conference on Construction in the 21st Century(CITC-12)*, (p. 643-649). Amman, Jordan.

Vailshery, L. S. (2022, November 22). *IoT global revenue 2020-2030, by vertical*. Retrieved from Statista: https://www.statista.com/statistics/1183471/iot-revenue-worldwide-by-vertical/

Vailshery, L. S. (2022, November 22). *Number of IoT connected devices worldwide 2019-2021, with forecasts to 2030*. Retrieved from Statista: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

Varma, J. R. (2019). Blockchain in Finance. *Vikalpa, 44*(1), 1-11.

Vorster, M. C. (1993). *Dispute prevention and resolution.* Construction Industry Institute.

Wang, J., Wu, P., Wang, X., & Shou, W. (2017). The outlook of blockchain technology for construction engineering management. *Front. Eng. Manag.*

Wang, X., Feng, D., & Lai, X. (2004). *Collisions for some hash functions MD4, MD5, HAVAL-128, RIPEMD.* Cryptology ePrint Archive, Report 2004/199.

Ward, S., & Chapman, C. (1994). Choosing contractor payment terms. *International Journal of Project Management*, 216-221.

Yang, Y., Chen, F., Zhang, X., Yu, J., & Zhang, P. (2017). Research on the Hash Function Structures and its Application. *Wireless Personal Communications*.

Yassein, M., Aljawarneh, S., Qawasmeh, E., Mardini, W., & Khamayseh, Y. (2017). Comprehensive study of symmetric key and asymmetric key encryption algorithms. *2017 International Conference on Engineering and Technology (ICET)*, (p. 1-7).

Ycharts. (2022, November 20). *www.ycharts.com*. Retrieved from https://ycharts.com/indicators/bitcoin_blockchain_size#:~:text=Basic%20Info,16.69%25%20from%20one%20year%20ago.

Yu, J., Kozhaya, D., Decouchant, J., & Esteves-Verissimo, P. (2018). RepuCoin: Your Reputation is Your Power.

Zandt, F. (2022, April). *Fake Goods Market Worth More Than Ireland's Economy.* Retrieved from Statista: https://www.statista.com/chart/27289/global-trade-volume-with-counterfeit-goods-compared-to-gdp-of-selected-countries-regions/

Zheng, Z., Xie, S., Dai, H.-N., Chen, W., & Chen, X. (2020). An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*.