UniCredit

Blockchain Technology and Applications from a Financial Perspective

Technical Report

Version 1.0

Data & Analytics

February 26, 2016

Disclaimer

The aim of this document is to provide the basis for a discussion on the blockchain technology and its potential applications in the banking framework. An example of smart contract is presented in Appendix B. Other examples related to this technology are available upon request. For any further information not included in this document, please refer to the authors listed below.

Authors

Matteo Biella (`matteo.biella@unicredit.eu`)

Vittorio Zinetti (`vittorio.zinetti@unicredit.eu`)

Review

Ivan Luciano Danesi (`ivanluciano.danesi@unicredit.eu`)

Cristina Rea (`cristina.rea@unicredit.eu`)

Approval

Nicola Breda (`nicola.breda@unicredit.eu`)

Tommaso Pellizzari (`tommaso.pellizzari@unicredit.eu`)

Contents

1 Introduction 3

2 Blockchain Technology 5
2.1 The blockchain . . . 5
2.1.1 Cryptography . . . 7
2.1.2 Consensus algorithms . . . 7
2.1.3 Second-generation blockchain . . . 8
2.2 Drawbacks and future insights . . . 8

3 Financial Use Cases 11
3.1 Payments . . . 12
3.1.1 Interbank payments - Conservative solution . . . 12
3.1.2 Interbank payments - Disruptive solution . . . 13
3.2 Know Your Customer process . . . 14
3.2.1 Conservative solution . . . 15
3.2.2 Disruptive solution . . . 15
3.3 Trade finance . . . 15
3.3.1 Disruptive solution . . . 16
3.3.2 Conservative solution . . . 17
3.4 Post-trade lifecycle . . . 17
3.4.1 Conservative solution . . . 19
3.4.2 Disruptive solution . . . 20

4 Discussion 21

A Cryptography 22

B Smart contracts 26

Chapter 1

Introduction

This article aims at presenting potential financial industry blockchain applications leveraging UniCredit laboratory experience.
Proposed vision promotes cross-effort and collaborative relationship amid financial institutions and fintech startups as blockchain initiatives' critical success factors.
Wide adoption of blockchain technology has the potential of reshaping the current financial services technical infrastructure. The change is expected to bring with it benefits to the existing business processes through removal of intermediaries, flat data structures that will reduce the lags of reconciliations among different local ledgers, compressed confirmation times and near real-time settlement of transactions.
Moreover, there are underlying technical aspects of the blockchain which will provide data and transaction immutability, resiliency against cyber-attacks and fault tolerance.
Formerly, blockchain technology is introduced from both a technological and a functional point of view. Then, financial use cases are proposed, showing financial industry impacts and benefits.

The idea of digital cash was first introduced in early '80s by David Chaum in [1] and [2]. Afterwards, institutions made some cryptocurrencies commercialization attempts introducing ecash and E-gold, to name a few. However, all these efforts failed due to different reasons, like lack of legal compliance, bad business management or network centralization (see [3]).

In 2008, a paper publication written by the pseudonym Satoshi Nakamoto [4] started a revolution in the cryptocurrency system introducing the Bitcoin framework. Bitcoin is a network that allows users to exchange ownership of a digital asset, called bitcoin. Until then, digital assets have always been conceived as easily replicable and a central authority that tracks balance of accounts has been used to solve the so called ”double-spending problem”[1]. For the first time, a system allows real-time exchanges of a digital asset between two unrelated entities without a central counterparty. Such transactions are subsequently recorded by network nodes in a public distributed database called blockchain.

[1] The double-spending problem refers to the fact that until Bitcoin, digital coins has always been conceived easily replicable, allowing the owner to double-spend them.

As reported in [5], blockchain technology can be conceived as the fifth paradigm of computing after (i) mainframe, (ii) personal computer, (iii) Internet and (iv) mobile and social network revolution.
Beyond Bitcoin, blockchain technology has the potential to reshape many other fields as shown by the interest of a wide range of private and public institutions (*e.g.* United Kingdom Government [6]).

From a financial institution's point of view, cryptocurrencies can be perceived as a threat. However, widespread adoption is still far from being certain, even though Bitcoin market capitalization is worth 6 billion dollars as of January 2016 [7].
Shifting focus from cryptocurrency to the technology underneath, it is possible to understand the real potential of blockchains in financial services. Post-Bitcoin phase has opened new opportunities, indeed financial institutions have already started to study and experiment with blockchains and a large number of related startups have been funded [8]. Venture capitalist (VC) investments in Bitcoin and blockchain industry show an increasing trend, considering that in 2015, investment summed up to 462 m$ until Q3, compared to a total of 230 m$ in 2014. [9]. A comparison between blockchain investments and VC investments in Internet years, presented in Figure 1, shows great similarity.

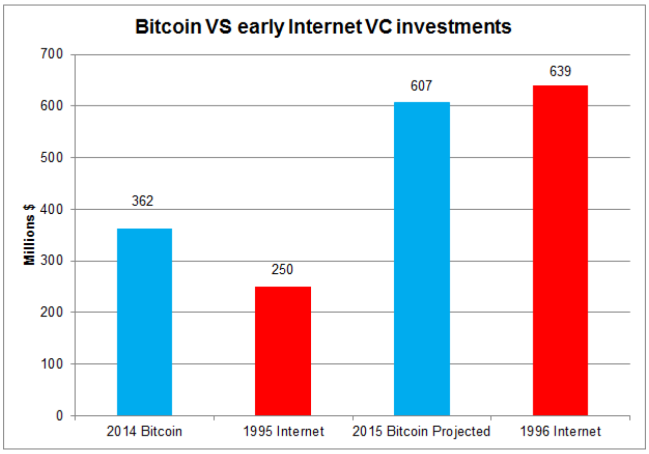

Figure 1.1: Comparison between VC investments in Internet and Bitcoin.

In Chapter 2 blockchain technology is introduced, discussing both benefits and limitations. As manifestation of the blockchain technology, cryptocurrency and smart contract frameworks are presented. Then some financial use cases are introduced in Chapter 3. Final discussion in Chapter 4 describes UniCredit's initiatives and outlines future perspectives in blockchain industry.

Chapter 2

Blockchain Technology

2.1 The blockchain

From a technical point of view, the blockchain is defined as a distributed replicated database that allows secure transactions without a central authority.
The first blockchain implementation is the cryptocurrency bitcoin: a cryptocurrency is a virtual currency that uses cryptography for security. In this context, the replicated database acts as global ledger tracking all cryptocurrency transactions between participants. A blockchain transaction is not limited to cryptocurrency, but can refer to any change in state of a digital asset defined on top of it. While first generation blockchains were created for cryptocurrency application, new generation blockchains allows custom digital assets' definition and will be explained in Section 2.1.3. Even if Bitcoin and blockchain are often used as synonyms, Bitcoin consists of a specific blockchain implementation.

From a functional point of view, the blockchain key element is the absence of a central authority for transaction validation, which is performed by a peer-to-peer network throughout a consensus process. A blockchain system is composed by two types of entities:

- **Participants**, who perform transactions secured by means of cryptographic signatures (see Appendix A).
- **A peer-to-peer network of nodes**, designated to validate transactions and to participate in the consensus process.

To understand where the name "blockchain" comes from, it is necessary to visualize how validated transactions are recorded. Transactions are grouped into blocks that are submitted to a network of validating nodes. Every time a block is validated, it is broadcasted to the network and added on top of the blockchain. Since every block contains a timestamp and a reference to the previous block, the blockchain is fundamentally a time stamping system represented by the chain of all blocks, starting from the first block. In Figure 2.1 a representation of the last two sequential blocks is given.

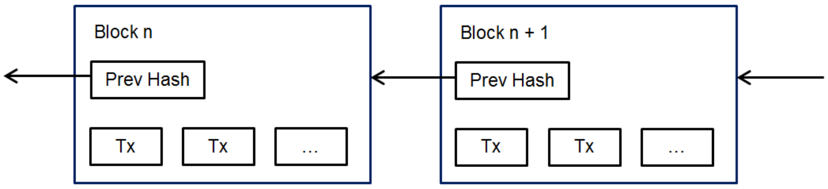

Figure 2.1: Simplified representation of the Blockchain.

The peer-to-peer network guarantees transactional security throughout the consensus process. Every time a new block is validated, each node verifies the block and updates its local copy of the database, adding a new block to the chain. Nodes follow the protocol that is embedded in the blockchain software, which determines a single state of the database even if there is no single authoritative copy of it.
While in a centralized architecture there is a single authoritative database operated by a single entity, in the blockchain every node has a local synchronized copy.
Even if in some cases there can be two different states of the blockchain in the network for a short time period, the consensus protocol allows fast and smooth convergence on a single state based on consensus outcome. Different types of consensus mechanism can be identified:

- **Proof-of-work**: it is the mechanism used by Bitcoin and is based on nodes competing to a computationally hard problem solution [4]. Network energy consumption is a major drawback.

- **Byzantine-fault-tolerant**: it is based on consensus method between authenticated validators, which is resilient to Byzantine attack[1].

- **Proof-of-stake**: it creates a disincentive mechanism for nodes that do not follow the consensus protocol. Validators are required to put “at stake” a predefined amount of a digital asset betting on consensus process outcome, so that malicious nodes that do not follow the protocol lose assets [11].

The consensus protocol not only determines a single authoritative state but also builds the blockchain so that it is **immutable**. Once a block is validated, it is infeasible[2] to change the blockchain without any tampering evidence. Immutability property is achieved throughout the combination of public-key cryptography and digital signatures. A detailed explanation of the subject involved is beyond the scope of this article, but the interested reader can refer to [12].

[1] In this context, a Byzantine attack refers to the possibility that a subset of the network nodes behaves maliciously [10].

[2] The economical cost for achieving the modification of a validated block protects the network from external attack described in section 2.2. The immutability property is guaranteed under the hypothesis of the network majority following the protocol.

Based on network access permission, there are two main categories of blockchain:

- **Permissionless**: network access is free and anyone can set up a node to validate transactions. Bitcoin and Ethereum [13] are the major examples.
- **Permissioned**: network access is restricted to a set of known participants. Ripple [3] [14] is an example of permissioned systems.

In permissionless systems, a reward in cryptocurrency is usually given to nodes for each validated block as an incentive to join and protect the network. While this was typical for cryptocurrency permissionless systems, in new generation permissioned blockchains an incentive mechanism is not required. This architecture is more suited for enterprise solutions where authentication of both nodes and participants is usually required. In the Appendix A first generation blockchain workflow is presented.

2.1.1 Cryptography

The blockchain uses cryptography to secure transactions. In order to interact with the blockchain, participants create a cryptographic key pair with a wallet software:

- A **private key**, which the user must not reveal, since it is used to sign transactions and to unlock cryptocurrency funds.
- A **public key**, which corresponds to the address of the associated account. It is used from participant to identify the receiver of a transaction.

Digital signatures protocols are employed in blockchains, in order to provide authentication and non-repudiation so that only the key-controlling entity can perform transactions from its associated account. An extensive explanation of cryptography goes beyond the scope of this article, but the interested reader can refer to [12].

2.1.2 Consensus algorithms

While in classical centralized architectures a single authoritative database is the point of truth, in blockchains every node has a local copy of the ledger. Blockchains systems employ a consensus mechanism for the definition of a single state of the ledger. Several consensus algorithms with different features can be employed depending on the use case.

Consensus mechanisms guarantee transaction security and ledger integrity even if a certain amount of nodes behave maliciously (*i.e.*, do not follow the blockchain protocol). The most common solution involves proof-of-work, in which security is guaranteed if at least 51% of the network's computational power follows the

[3] Even if Ripple has been initially conceived as a permissionless system, Ripple Labs turned the project in a permissioned one focusing on financial institutions.

protocol. Since each node is required to perform a certain amount of computational work in order to create a valid block, controlling a large part of the network for a malicious actor would be costly, difficult and would probably lap the gained advantage. Such an attack's aim would be to replace transactions that have already been included in blocks and perform double-spending of funds. Thus, blockchain can be considered immutable if no single malicious entity controls the network majority.
The drawback of proof-of-work is the huge computational power and energy amount required to validate transactions. For this reason new consensus algorithms were proposed in order to reduce the computational cost of mining. For example, proof-of-stake scheme is based on a disincentive mechanism to punish nodes acting regardless of protocol. Examples of this consensus protocol are Tendermint [15] and the future replacement of the current Ethereum protocol [16]. Other variations are related to the following algorithms: Practical Byzantine fault-tolerant [10], Quorum slicing [17].

2.1.3 Second-generation blockchain

In most cases blockchains support not only cryptocurrency transfer, but also smart contracts. A smart contract is a representation of a real world contract throughout a computer program. Rules of the contract are embedded into code deployed on the blockchain. This allows the definition of new custom assets modelling and management, as shown in Chapter 3 and also creates potential applications outside financial domain [6]. Smart contracts are supported by Bitcoin, but they offer limited functionalities. Second generation blockchains frameworks allow smart contracts' implementation with more advanced capabilities[4]. An example of smart contract for the Ethereum framework written in the Solidity [18] programming language is presented in the Appendix B.

2.2 Drawbacks and future insights

In this section the challenges that can limit the applicability of the blockchain to financial institutions are described. The goal is to address existing limits for the use case analysis of the next section. Future technological developments that can overcome these issues are also considered.

Throughput and latency

Network **latency** is one of the main limitations in many blockchain systems. In Bitcoin, for example the average confirmation time per block is 10 minutes, and for security purpose it is highly recommended to wait a certain number of block validations in order to lower the probability of a successful double-spending attack. The validation time is enforced by a mechanism in the Bitcoin protocol that regulate the difficulty of proof-of-work in order to maintain an average of 10 minutes per block confirmation. For comparison, transactions on credit card payment networks only take seconds to be confirmed.

[4] Second generation blockchains smart contracts are based on Turing complete programming languages [18].

Transaction **throughput** in Bitcoin can reach a theoretical maximum of 7 transactions per second (txps). By contrast, retail payment networks allow an average of thousands txps with a peak in the order of 10^4 txps. In Bitcoin a block size limit is enforced by the protocol for security purposes, even if there is a debate in the Bitcoin community about possible solutions to the problem [19]. In second-generation blockchain adoption of proof-of-stake or Byzantine fault-tolerant protocol, which can validate blocks in seconds and provide a higher transaction throughput is implemented. These protocols are most suited in permissioned and private systems, where each node's identity must be known. One example of permissioned system is Ripple [20] for cross-border payments. Another issue is the computational cost of proof-of-work consensus that produces a huge energy consumption [21]. This problem can be solved with proof-of-stake or Byzantine fault tolerant consensus algorithm, which are much less computational intensive.

Privacy

In blockchain, every node is required to hold the entire database since the network beginning in order to check every transaction legitimacy during consensus. Especially for the financial industries, there are two major consequences:

- **Regulatory requirements**: in some context information about customers shall be kept private.
- **Industrial strategy**: in the financial industry, competitors wants to keep information about their transactions private.

A possible solution to this problem is homomorphic encryption: it allows to directly perform operations on encrypted data, so that nodes can check legitimacy of transaction and store data, without revealing anything about them. Examples of such a technology are the Enigma project developed at MIT [22] and the Zerocash project [23].

Size and bandwidth

As of January 2016, the size of the Bitcoin blockchain is over 50 GB [24] and grows about 15 GB/year. In systems with a high transaction rate and smart contracts, the amount of data can become a challenge, even for nodes employing enterprise hardware. Increasing transaction throughput to VISA standards would cause a growth of 3.9 GB/day or 1.42 PB/year [5]. A possible solution is the Bitcoin Lighting protocol [25], which allows to perform off-chain micro transactions, using micropayment channels.

Security

As previously stated, Bitcoin network is considered secure to a threshold and design assumption that network majority is controlled by fair entities following the protocol. Attacks against this hypothesis are called 51% attacks, because theoretically an attacker would need to control at least 51% of Bitcoin network to perform a double-spend attack[5].

[5] In [26] it is showed that this threshold can be instead 33% in case of "Selfish mining".

Furthermore blockchain systems are potentially more resilient to cyber-attacks compared to centralized systems. Being the blockchain replicated and updated throughout consensus, a successful attack would require to compromise a large portion of the network. Keys management is another important issue in blockchain systems: the use of e-wallet services or cold storage wallets are possible solutions to mitigate the associated risks [27].

Blockchains that possess all these features do not exist at the time of writing and for some of the issues reported in this chapter a proper solution still needs to be defined. Anyway, the fintech industry is evolving at a fast rate of innovation. Considering that in 2009 there was just the Bitcoin and now there are about 600 cryptocurrencies [7] it is reasonable to think that most of these issues could be solved with near future technological advancements.

Chapter 3

Financial Use Cases

In this Chapter blockchain technology application is presented in different financial use cases. Two approaches are discussed: the former considers a restricted scenario in order to ensure fast adoption, while the latter describes a "disruptive solution" spanning through a longer period.

Financial institutions can create permissioned blockchain platforms. In this context, banks will be both participants and validators in the network. Banks can either use these platforms to interact in a decentralized way, or they can offer their customers access as a service.
Furthermore, banks can offer blockchain services to access external platforms. In recent years, several fintech startups developed blockchain-based systems and services, but their complexity and lack of customer confidence often prevented a wide adoption. For such reasons, banks can act as blockchain services gateways developed in the fintech world, integrating them in existing systems. A concrete example is blockchain invoice management service Tallystick developed by Applied Blockchain [28]. Their service offers companies a simplified invoice management processing, but requires integration with existing financial systems. In this context, banks can introduce their customers to the network, acting as gateways and facilitators. In other cases, banks participate in blockchain platform created by several fintech startups. An example is Ripple, a payment network in which banks not only perform transactions, but can also control validator nodes. In future many blockchains with specific target domains and therefore tailor-fit technological features will probably coexist, such as in payments, financial securities and digital identity to name a few. Banks can offer tools to manage all these different blockchains, facilitating users' access and further guarantee interoperability.

Financial institutions should also monitor cryptocurrencies evolution. Banks involvement inside cryptocurrency domain is controlled by regulatory authorities, which used to discourage financial institutions to enter cryptocurrency business [29]. Lately, regulators have started to consider the definition of a legal framework [30] and certain States have already adopted established rules [31]. In this scenario, banks could offer cryptocurrency custody services (*i.e.*, acting as current e-wallet services) and enable customers to perform transactions on the blockchain. Furthermore, they could also offer exchange services between cryptocurrencies and *fiat*[1] currencies.

3.1 Payments

The first blockchain application was a cryptocurrency system, in which payments are executed between sender and recipient without a central counterparty. To better understand such architecture's potential benefit for financial institutions, it is useful to point out how the current payment system works. In a standard inter-bank funds transfer, if the sending and receiving banks do not have a reciprocal account, they have to rely on an intermediary clearing-house or a correspondent bank, as shown in Figure 3.1. Payment workflow from execution to settlement takes days, and fees have to be paid to intermediaries.

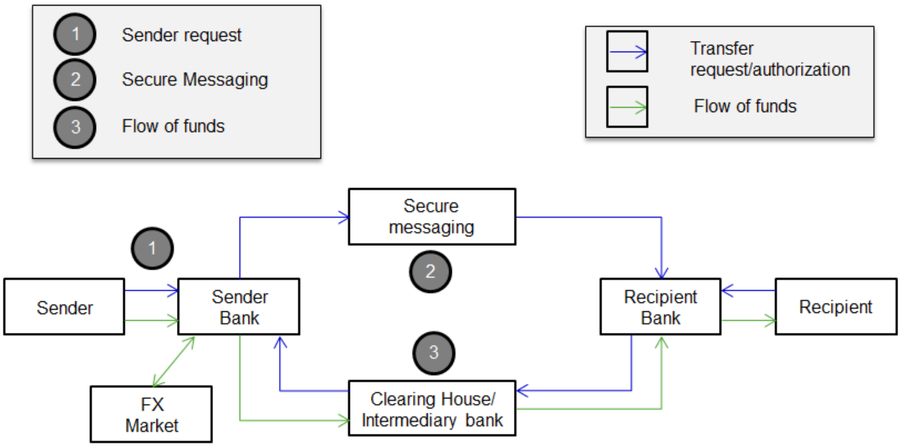

Figure 3.1: Current inter-banks payment system workflow.

3.1.1 Interbank payments - Conservative solution

Let us first consider payment systems between legal entities belonging to the same banking group. Inter-bank payments are usually performed using a central counterparty and every bank has a local database, which acts as authoritative ledger where all account balances and transactions are recorded. This implies several drawbacks: first, the local databases should be reconciled and keep in sync. Second, payments are performed settling net obligations throughout accounts recorded by a central counterparty, as shown in Figure 3.2.

[1] Fiat money is a currency established as money by government regulation or law.

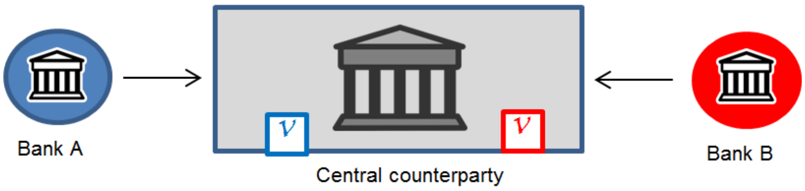

Figure 3.2: Centralized payment architecture.

A conservative solution could be blockchain adoption as ledger for payments between banks belonging to the same group. Every bank would be a private blockchain network participant and be able to perform transactions and participate to consensus, as shown in Figure 3.3.

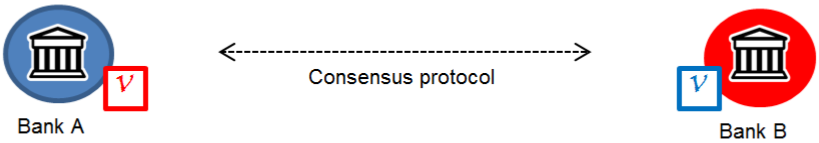

Figure 3.3: Decentralized payment architecture.

By adopting this solution, reconciliation between different databases is no longer needed, since a single ledger authoritative state is obtained by consensus. Furthermore, payments can be settled between banks without using an intermediary and virtually eliminating fees involved. Execution happens in near-real time and in a peer-to-peer fashion, reducing counterparty risk and lowering time to settlement to seconds. From a regulatory point of view, a blockchain would be the shared immutable ledger of all transactions and access would be granted to regulators and auditors.

Transaction privacy between legal entities can be an issue, since with traditional blockchains every legal entity node would have access to other participants' records. This may represent a problem for compliance with the privacy legal framework. Solutions to maintain participants' data private are currently under study [22].

3.1.2 Interbank payments - Disruptive solution

Former discussion can be extended to banks either belonging to different groups or cross border payment networks. Interbank payments settle net payment obligations using multiple central counterparts, acting in a particular network. To minimize counterparty risk, each bank has to maintain a reserve account for each payment network, as shown in Figure 3.4.

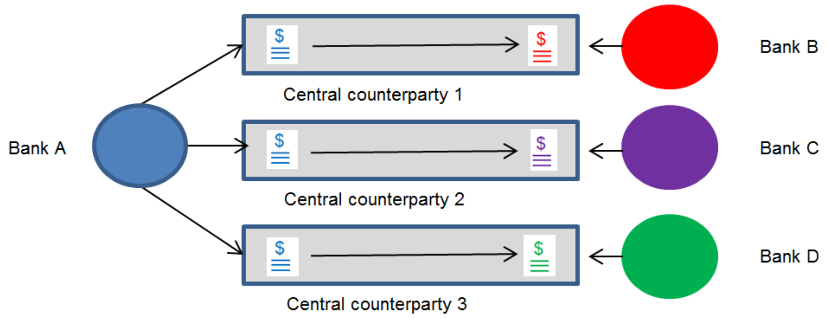

Figure 3.4: Interbank payments scheme with multiple reserves.

In this case, implementation of a permissioned blockchain can be done between banks belonging to different groups. Key advantage is that cross border payments can be executed without correspondent banks, consistently lowering capital requirements associated to intermediaries and increasing resources that can be allocated for banking business instead. In Figure 3.5 a global blockchain platform architecture for payments with a single reserve account is presented.

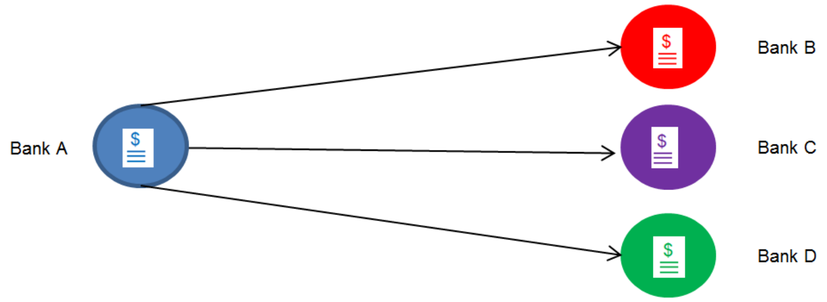

Figure 3.5: Interbank payments scheme with single reserve.

This solution can become attractive if a large number of banks participate on the network. An example of cross-border payment system based on a blockchain is Ripple, in which validator nodes are operated by identified financial institutions participating to consensus. Ripple also integrates currency exchange capability to provide liquidity for cross-border payments.

3.2 Know Your Customer process

Know Your Customer (KYC) is a process required to banks and financial institutions in order to verify clients' identity. It is a time consuming process which requires a lot of paperwork resulting in high costs. One attempt to increase process efficiency is identity registries creation, such as SWIFT interbank registry. This centralized registry provides instantaneous access to reliable data about customer's identities to SWIFT's members. A Blockchain has already been applied in the KYC domain, outside financial industry [32]. In this context, a blockchain acts as a cryptographically secured identities database in which banks and financial institution can access customer data.

3.2.1 Conservative solution

In a KYC process a lot of entities are involved: customers are required to provide documents issued by government institutions or trusted companies to banks, which have to verify their identity.Conservative solution assumes that the bank itself will adopt blockchain technology.

In this use case, blockchain will be the identity registry for banking group legal entities. Every customer would have a single cryptographic identity, even if it has accounts or assets in multiple legal entities. Customers' data will be securely recorded on the blockchain and will be available to all group's banks, which will have a consistent picture of customer data. Limits to this solution are related to privacy, due to confidentiality requirement being different in various legal frameworks.

An alternative approach consists in recording customers' documents' hashes on blockchain, rather than digital files. Hashed code acts as a proof of existence and authenticity. Digital documents presented by customers to banks, are hashed and verified using blockchain-recorded fingerprint. This can potentially speed up customer identity verification process.

3.2.2 Disruptive solution

In a disruptive solution people identities will be almost completely managed using blockchains. Identity cards, passports, driving licenses will have also a digital blockchain-recorded version. Any institution could possibly issue documents' fingerprint on the blockchain and customer identity will be cryptographically and digitally proved. Banks which will receive documents' digital versions from their customers will use blockchain-defined fingerprint to prove document authenticity and validity without need of further due diligence. In an alternative scenario, customers' data will be directly recorded on the blockchain, even if this would require a proper solution to scalability problem. Solutions with multiple blockchain connected with inter-chain protocol have also been proposed to improve scalability [33]. Implementation of such a disruptive solution would certainly require a long time frame, since major collaboration between governments and other institutions is mandatory.

3.3 Trade finance

Trade finance is the process of financing both domestic and international trades. A trade usually involves a seller of goods or services, a buyer and either a bank or a financial institution. The third party role is to reduce risks for both counterparties. Current trade finance processes are quite complex and require a considerable amount of manual and time-consuming steps. In this Section, letter of credit is analysed. Current letter of credit processes involve goods or service exchange between a seller and a buyer, which typically have a limited reciprocal knowledge. Buyer wants to mitigate risk of seller not fulfilling goods' sending, while seller wants to mitigate risk that buyer is not paying. The process workflow, which is shown in Figure 3.6, typically involves the following steps:

- After negotiation, buyer and seller sign a contract.
- Buyer's bank supplies a credit letter to seller's bank, which guarantees seller will be paid whether certain conditions are met.
- Seller gives goods to a carrier and receives a bill of lading.
- Seller provides the bill of lading to its bank and receives the payment.
- Bank gives buyer the bill of lading, so that he can present it to carrier and receive goods.

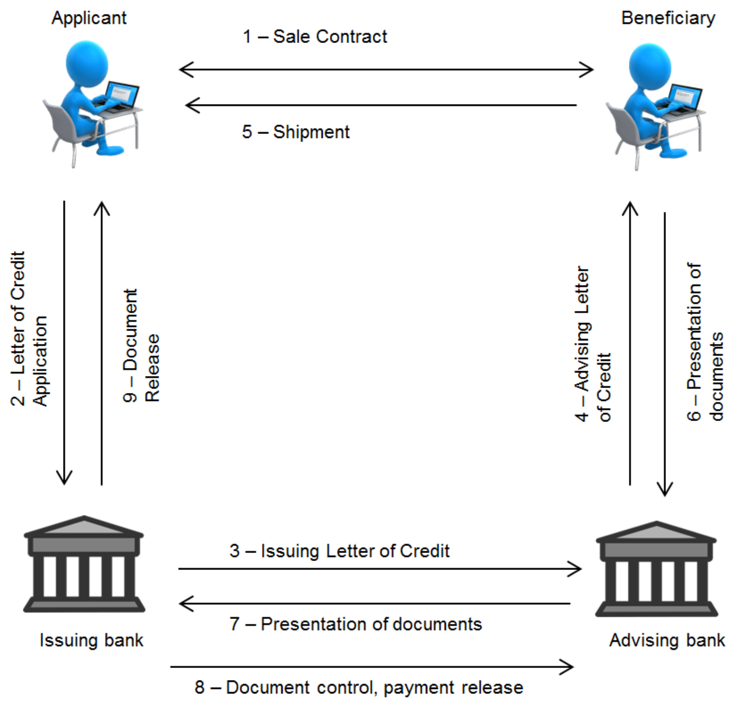
Applicant
Beneficiary
1 – Sale Contract
5 – Shipment
2 – Letter of Credit Application
9 – Document Release
4 – Advising Letter of Credit
6 – Presentation of documents
3 – Issuing Letter of Credit
7 – Presentation of documents
Issuing bank
Advising bank
8 – Document control, payment release

Figure 3.6: Credit letter workflow.

3.3.1 Disruptive solution

Formerly a disruptive solution is presented from which a conservative one can be easily derived. Since the letter of credit process is mandatory to solve trust problems between counterparties and typically requires days for settlement, application of a blockchain solution using smart contracts can speed up and automate this process. A permissioned blockchain platform involving banks, companies and carriers is considered. Letter of credit rules can be embedded in smart contracts created by the bank. Buyer and seller will have an account with funds on blockchain, which they can use for payments and smart contract interactions. A smart contract will be created and deployed on the blockchain by the issuing bank, which will encode credit letter's rules. Participants can interact with smart contract credit letter by the following methods:

- **Request contract terms**: both buyer and seller can call this method to receive information about credit letter terms. This would include, for example, information about buyer, seller, banks, carrier, amount to be paid and documents that seller has to present to carrier.
- **Carrier confirmation**: it will be sent by carrier once he has shipped the requested goods.
- **Bank confirmation**: it will trigger payment if carrier confirmation has already been sent.
- **Check confirmation**: this method can be called by bank, buyer or seller to see confirmation status.

When a seller gives goods to a carrier, the carrier will send a confirmation message to the smart contract. This process would be real time and will eliminate the need for the seller to take the bill of lading and present it to the bank. Once bank notifies carrier confirmation using check confirmation method, it will send a bank confirmation. When the contract detects that both carrier and bank have the confirmation signed, it will automatically trigger fund transfer between bank accounts. All interactions with smart contracts are cryptographically signed and recorded on the blockchain. Contract rules are enforced by the smart contract code and participants can query contract information on the blockchain. In more advanced scenarios it could be possible an integration with the Internet of Things. Smart devices could be used to monitor the state of goods during shipping and sensor data can be recorded on the blockchain.

3.3.2 Conservative solution

In a conservative solution, only banks would use blockchain to automate contract execution. This scenario will be simpler, since only banks will have to interact with smart contract to confirm that bill of lading has been received and triggers payment to seller. However, paperwork will still be in place and there would not be any benefit in terms of time reduction. Furthermore, such a solution can be put in place without using blockchain, by simply automating contract on classical platforms. Nevertheless, this solution can be well suited to test technology before a transition to target solution, whose advantages would be exploited only when a large number of entities would participate.

3.4 Post-trade lifecycle

Securities trading lifecycle is a process which involves a large number of steps and actors, which are introduced to mitigate different type of risks. The use of blockchain technology in this context has already been considered by some market infrastructure institutions [34]. This process today takes about three days between trade execution and settlement. Trading lifecycle structure can be generalized in the following steps:

- **Trade execution**: in this phase buyer and seller request an order to their brokers, who act on clients behalf and submit orders to an exchange. When orders are matched, a confirmation is returned to brokers.
- **Trade clearance**: details about orders are sent to a clearinghouse, and original contract novation[2] takes place. At this point clearinghouse acts as buyer to seller and as seller to buyer. In this way, clearinghouse will take settlement risk and guarantee trade executiond for both counterparties. In front on that, clearinghouse requests adequate margins to their clearing members for mitigating their default risk.
- **Trade settlement**: in this phase obligations settlement is performed using netting. This involves orders grouping into a single transaction leveraging a custodian. In the end, buyer sets his obligation with the custodians throughout delivery versus payment and sellers get paid.

The trading lifecycle workflow and timeline are represented in Figure 3.7 and 3.8.

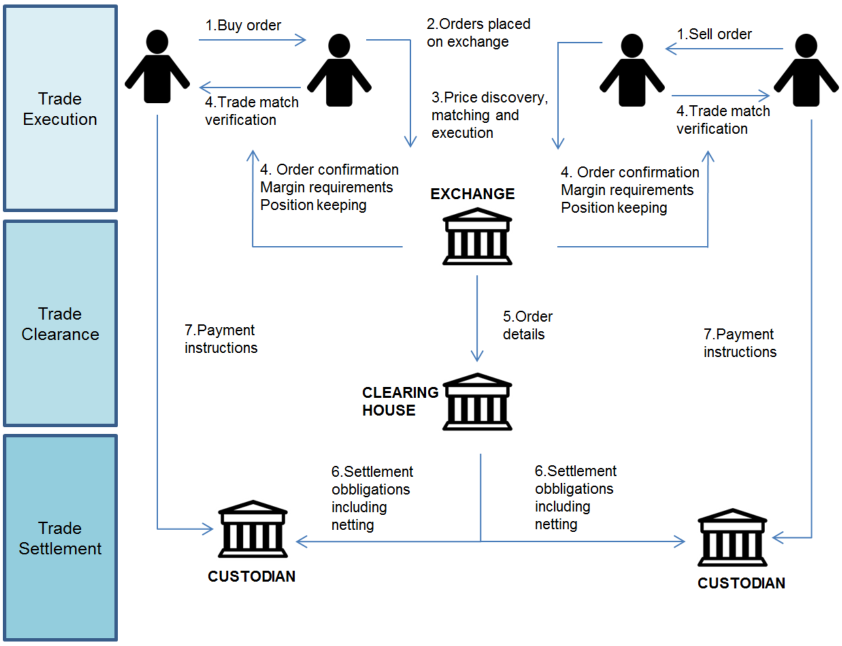
Trade Execution
Trade Clearance
Trade Settlement
1.Buy order
4.Trade match verification
2.Orders placed on exchange
3.Price discovery, matching and execution
1.Sell order
4.Trade match verification
4. Order confirmation Margin requirements Position keeping
EXCHANGE
4. Order confirmation Margin requirements Position keeping
7.Payment instructions
5.Order details
7.Payment instructions
CLEARING HOUSE
6.Settlement obbligations including netting
6.Settlement obbligations including netting
CUSTODIAN
CUSTODIAN

Figure 3.7: Security trading lifecycle.

[2] Trade tiles transferring to clearinghouse process is called novation.

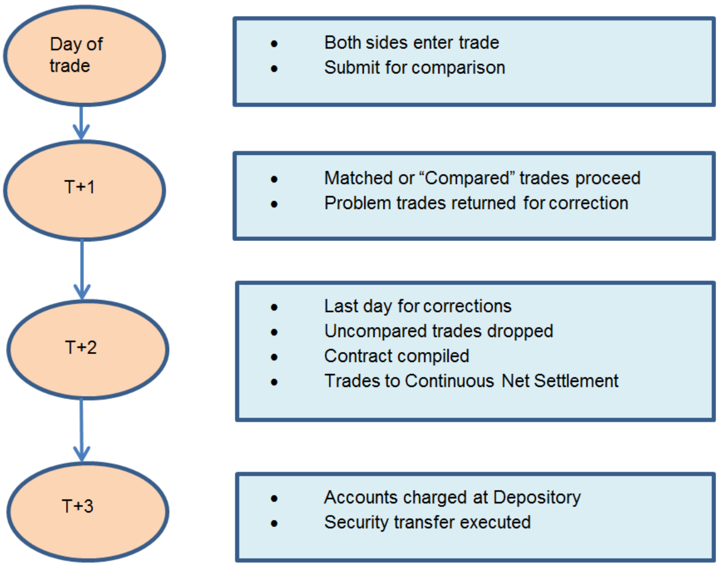
Day of trade
Both sides enter trade
Submit for comparison
T+1
Matched or "Compared" trades proceed
Problem trades returned for correction
T+2
Last day for corrections
Uncompared trades dropped
Contract compiled
Trades to Continuous Net Settlement
T+3
Accounts charged at Depository
Security transfer executed

Figure 3.8: Trade settlement workflow.

3.4.1 Conservative solution

Blockchain technology together with smart contracts allows the creation of any kind of token into the blockchain itself so that bonds, shares and other financial assets can be represented. One possibility to simplify existing process is to represent securities and perform post-trade lifecycle on blockchain. Clearing and settlement will happen on blockchain, which will also be the depository for the assets. In this view, order matching is still done off-chain and then a clearinghouse can perform novation and split orders in a sell order with the original buyer and a buy order with the original seller. The entire process, as well as informations and rules representing securities will be encoded in smart contracts.

Let us consider a smart contract between a clearinghouse and a buyer. The clearinghouse will issue this contract once the trade is matched which will have the following methods to interact with the buyer:

- **Get information method**: information about type, amount and securities rules can be requested by the buyer to review the order.
- **Delivery versus payment method (DVP)**: the buyer can call this method to pay the clearinghouse and receive securities. Settlement is triggered only if the buyer has the necessary funds.

In order to perform the payment, the buyer must have an account on the blockchain with adequate amount of funds. In case the buyer has not got enough funds on his account, a call to the DVP method would return an error and securities would not be settled. Other features can be added: the clearinghouse can for example set a timeframe in which DVP can be executed or create margin accounts for clearing members, with custom rules defined in smart contracts. This solution reduces clearing and settlement time from days to minutes.

Furthermore, blockchain acts either as both clearing and settlement platform eliminating the need for reconciliation among different actors. Contract terms execution throughout code reduces back office workload and risk of errors.

3.4.2 Disruptive solution

A more sophisticated solution can also consider trade matching and execution on blockchain. In this case, the following participants would be involved:

- **Brokers** - which would have blockchain accounts and place orders for their clients in form of smart contracts.
- **Clearing firms** - which can manage client identity through a KYC registry and would request brokers the margin requirements for trading activity.
- **Clients** among which payments can be done directly throughout accounts registered on the blockchain.

A private permissioned blockchain in which nodes are controlled by a consortium composed by brokers and clearing firms is considered. The brokers can put orders on the blockchain as smart contracts, which automatically handle trading matching and execution. Clearing firms role will remain similar to the conservative scenario one. Even if this solution will be more complex and require coordination between a large number of participants, it will bring most benefits in terms of time reduction for trading lifecycle.

Chapter 4

Discussion

In this article, blockchain technology is introduced both from a technological and functional point of view. The role of cryptography and the different consensus mechanisms for blockchain technologies are discussed. Cryptocurrencies features and next generation blockchains with smart contracts are explained. Limitations about scalability, privacy and security as well as possible technological solutions were presented. Then some relevant financial use cases are analysed and applications based on blockchain technology are proposed showing potential benefits for the banking industry.
Successful blockchain implementation in financial industry depends on the overcoming of technological downsides shown in Section 2.2. Nevertheless, the use cases analysis of Chapter 3 shows that blockchain can bring great benefits in terms of efficiency increase and cost reduction.

A large participation into blockchain applications will maximize financial industry advantages. Furthermore a relationship with fintech environment based not only on competition but also on collaboration will bring great advantages to both financial institutions and fintech startups. UniCredit is investing in blockchain technology throughout internal research, experimentation and participation to DLG banking consortium [35]. This contribution will play an important role in the global blockchain initiative.

Appendix A

Cryptography

Cryptography is the science of securely communicating secret messages in presence of third party adversaries in the communication channel. Blockchain transactions are secured by cryptography. Different types of blockchains use different cryptographic algorithm based on the same foundation. In this article, we present the general cryptographic primitives used in the Bitcoin blockchain, even if these concepts can be applied to a broader class of blockchain.

Public key cryptography

The basic method of securing the communication of a secret message throughout an insecure channel is symmetric cryptography: in this scheme, two parties (Alice and Bob) want to send private message to each other and they share a common secret called private key. When Alice wants to send a message to Bob, she encrypts the message using an encryption algorithm and the private key producing a cyphertext that can be sent through the insecure communication channel. This process is shown in Figure A.1. The encryption algorithm is designed to produce cyphertext from which it is infeasible to recover the original message without the associated private key, so that an adversary who intercepts it through the communication channel cannot recover any information about the original message.

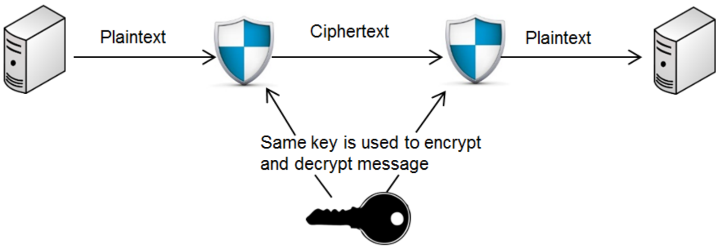

Figure A.1: Symmetric key cryptography.

Since keys exchange cannot be executed throughout an insecure channel due to the presence of third party adversaries, it is difficult to achieve for two parties that are not at close distance. Public key cryptography was born in the '70s to address the problem of key distribution. In this scheme, Alice and Bob have two different pair of keys. Each pair is composed by a private key, which must not be revealed and a public key, which can be sent to an insecure channel. Both Alice and Bob can generate their key pair using a key generation algorithm. The key advantage of the public key cryptography scheme is that the two parties need only to exchange their public keys to securely communicate within each other. The public key scheme is shown in Figure A.2. In this process, first Bob sends his public key to Alice throughout the insecure channel. Then Alice uses Bob's public key to encrypt the message and produce the cyphertext. In the end, Alice sends the cyphertext to Bob, which can recover the original message, decrypting the cyphertext using his private key. In real world application, this process is more complex since other mechanisms are involved in order to prevent other types of attack. A complete explanation of public key cryptography goes beyond the scope of this paper, but the interested reader can refer to [12].

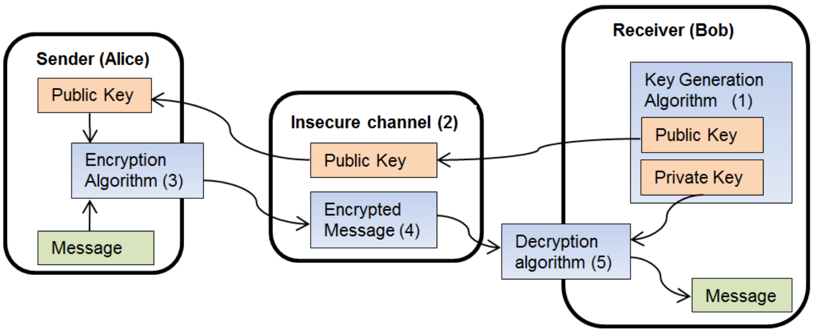

Figure A.2: Public key cryptography.

Digital signature

Digital signature is a cryptographic technology applied to a file that replicate the same functions of an ink signature on a paper document. It guarantees that the signed digital document was generated by the singer, has not been tampered with and that the signature cannot be rejected. In the Figure A.3, a usage scheme of digital signature is shown. In this process, first Alice produces a key pair and sends her public key to Bob throughout an insecure channel. Then she signs the message throughout a signing algorithm and her private key. In the end she sends the message to Bob, who can use the public key and a verification algorithm to verify the message.

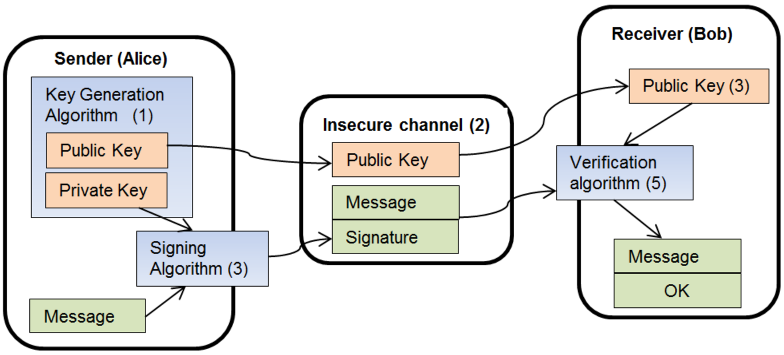

Figure A.3: Digital signature scheme.

Account, wallet and transactions

In Bitcoin, users' accounts are identified and managed through cryptographic key pairs. A user employs a wallet software to receive and spend his bitcoins. The public key is associated with the address of the account, so that the user can give it to anyone in order to receive funds. The private key is used to sign transactions to spend bitcoins in the user's wallet. A transaction is a message signed with the private key, which contains the following data[1]:

- The address from which bitcoins are taken
- The amount of bitcoin to be transferred
- The address of the receiver of the funds

A digital signature protocol is employed to ensures that only the owner of the account, who knows the private keys, can unlock the funds.

Cryptocurrency blockchain workflow

To understand the basic functioning of the blockchain it is useful to refer to Bitcoin. In this context, each user has an account with the associated amount of bitcoins registered on the blockchain. The funds are controlled throughout a cryptographic key pair created by the user with a wallet software. The wallet software is used to manage the keys and to sign transactions. A typical transaction workflow in Bitcoin involves the following steps:

- Alice creates and signs a transaction from his account to the one of Bob
- The transaction is broadcasted into the peer-to-peer Bitcoin network
- Validator nodes check if the signature is valid and if the sender's account have enough bitcoins: if the verification is successful, nodes include the transaction in the current block

[1] This is a simplification of the real message [12].

- Every validator node participates to the consensus process applying the proof-of-work algorithm: once a block is validated, it transmitted through the network
- All the nodes check the validity of the block and, if the verification is successful, they add it to the blockchain.

Appendix B

Smart contracts

In this section, a fixed rate bond smart contract is implemented in the Solidity programming language for the Ethereum Blockchain [13]. For simplicity, the only two actors in the contract are the bank, which issues the bond, and the customer. The `Bond` contract will use two libraries:

- `CurrencyManager`: handles the transaction and customers' balances verification.
- `KYCRegistry`: provides a method for the bank to check if the customer is autorizhed to buy the bond.

For the sake of simplicity, we present only the declaration of these libraries.

```
contract CurrencyManager{
function transfer(address from, address to, string currency, uint value)
                 constant returns (bool);
function getBalance(address from, string currency) constant returns (uint);
}

contract KYCRegistry{
function isAuthorized(address addr) constant returns (bool);
}
```

In Solidity smart contracts are similar to objects in object oriented languages. The data are encoded in object's members, while interactions with the objects are described through methods. The contract has the following features:

- **Issuing rule**: definition of bond features and amounts issued.
- **Buying rule**: only identities allowed by the KYC registry can buy bonds.
- **Coupon payment**: payment function with internal check for date correctness.

```
contract CurrencyManager{
function transfer(address from, address to, string currency, uint value) constant returns (bool);
function getBalance(address from, string currency) constant returns (uint);
}

contract KYCRegistry{
```

```
function isAuthorized(address addr) constant returns (bool);
}

contract Bond{

// Global parameters

uint bondNumberSold;
uint bondNumberAvailable;
mapping(uint => addressContainer) bondDistribution;
uint numberOfAddress;
address currManAddress;
address KYCRegistryAddress;
uint onHold = 0;

// Bonnd Structure

struct BondData {
string isin;
uint[] paymentDates;
uint singleIssuePrice;
uint interestRate;
uint repayment;
string currency;
}

BondData bondData;

// Address container structure

struct addressContainer {
address addr;
uint amountBond;
uint amountBondToSell;
uint priceBondToSell;
uint amountBondToBuy;
uint priceBondToBuy;
}

// Events

event ConfirmationBondSale(address buyer, uint amount, uint valuePaid);
event ConfirmationPayment(address buyer, uint amount, uint valuePaid, uint referenceDate);
event ErrorTransactionMessage(uint reason);
event ErrorNotEnoughBond(uint amount, uint bondNumberAvailable);
event ErrorNotEnoughMoney(uint amount, string currency, uint balance);
event ConfirmationBondSaleBetweenPrivates(address seller, address buyer, uint amount, uint valuePaid);

// Bond constructor

function Bond(string _isin, uint amount, uint[] _paymentDates, uint _singleIssuePrice, uint _interestRate,
 uint _repayment, string _currency, address addr, address _KYCRegistryAddress) {
bondData.isin = _isin;
bondData.singleIssuePrice=_singleIssuePrice;
bondData.interestRate=_interestRate;
bondData.repayment=_repayment;
bondNumberAvailable=amount;
bondNumberSold=0;
numberOfAddress=0;
bondData.paymentDates=_paymentDates;
bondData.currency = _currency;
currManAddress = addr;
```

```
KYCRegistryAddress = _KYCRegistryAddress;
}

// Bond methods

function getBondPrice(uint amount) constant returns (uint){
if(amount > bondNumberAvailable){
ErrorNotEnoughBond(amount, bondNumberAvailable);
return 0;
}

uint etherAmount = amount * bondData.singleIssuePrice;
return etherAmount;

}

function getMyBondAmount() constant returns (uint){
for(uint i = 0; i < numberOfAddress; i++){
if (bondDistribution[i].addr == msg.sender) {
return bondDistribution[i].amountBond;
}
}

return 0;

}

function getIsin() constant returns (string){

return bondData.isin;

}

function buyBondFromIssuer(uint initAmount, uint amount) {

KYCRegistry kycRegistry = KYCRegistry(KYCRegistryAddress);
if(!kycRegistry.isAuthorized(tx.origin) || !kycRegistry.isAuthorized(msg.sender)){
ErrorTransactionMessage(500);
return;
}

if(amount != initAmount / bondData.singleIssuePrice){
ErrorTransactionMessage(100);
return;
}

if(amount > bondNumberAvailable){
ErrorNotEnoughBond(amount, bondNumberAvailable);
return;
}

CurrencyManager currManager = CurrencyManager(currManAddress);

bool isTramsactionDone = currManager.transfer(msg.sender, address(this), bondData.currency, initAmount);
if(!isTramsactionDone){
ErrorNotEnoughMoney(initAmount, bondData.currency, currManager.getBalance(msg.sender, bondData.currency));
return;
}

uint foundAddress = 0;
for(uint i = 0; i < numberOfAddress; i++){
if (bondDistribution[i].addr == msg.sender) {
```

```
foundAddress = 1;
uint currentAmount = bondDistribution[i].amountBond;
currentAmount = currentAmount + amount;
bondDistribution[i].amountBond = currentAmount;
bondNumberSold += amount;
bondNumberAvailable -= amount;
}
}

if(foundAddress == 0){
bondDistribution[numberOfAddress].addr = msg.sender;
bondDistribution[numberOfAddress].amountBond = amount;
numberOfAddress++;
bondNumberSold += amount;
bondNumberAvailable -= amount;
}

ConfirmationBondSale(msg.sender, amount, initAmount);

}


function makePayments() {

if(onHold == 1){
ErrorTransactionMessage(1);
return;
}

uint currentTime = now;
uint index = 0;
uint leng = bondData.paymentDates.length-1;
while (index < leng && bondData.paymentDates[index] == 0){
index++;
}

if(currentTime < bondData.paymentDates[index] || bondData.paymentDates[index] == 0){
ErrorTransactionMessage(1);
return;
}

onHold = 1;

uint referenceDate = bondData.paymentDates[index];
bondData.paymentDates[index] = 0;

onHold = 0;

for(uint i = 0; i < numberOfAddress; i++){
uint payment = bondData.interestRate * bondDistribution[i].amountBond;
if(index == leng){
payment += bondData.repayment * bondDistribution[i].amountBond;
}

CurrencyManager currManager = CurrencyManager(currManAddress);

bool isTramsactionDone = currManager.transfer(address(this), bondDistribution[i].addr, bondData.currency,
 payment);
if(!isTramsactionDone){
uint availableAmount=currManager.getBalance(address(this), bondData.currency);
ErrorNotEnoughMoney(payment, bondData.currency, availableAmount);
return;
```

```
}

ConfirmationPayment(bondDistribution[i].addr, bondDistribution[i].amountBond, payment, referenceDate);
}


}
}
```

Bibliography

[1] David Chaum. Blind signatures for untraceable payments. In *Advances in cryptology*, pages 199–203. Springer, 1983.

[2] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[3] Dominic Frisby. Bitcoin: The future of money, 2014.

[4] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1(2012):28, 2008.

[5] Melanie Swan. *Blockchain: Blueprint for a New Economy.* ” O’Reilly Media, Inc.”, 2015.

[6] UK Government Chief Scientific Advisor. Distributed ledger technology: beyond block chain.

[7] Crypto-currency market capitalizations. URL http://coinmarketcap.com/.

[8] Startup management update to the global landscape of blockchain companies in financial services. URL http://startupmanagement.org/2015/12/08/update-to-the-global-landscape-of-blockchain-companies-in-financial-services/.

[9] Coindesk. State of Bitcoin and blockchain Q3 2015. URL http://www.slideshare.net/CoinDesk/state-of-bitcoin-2015?ref=http://www.coindesk.com/state-bitcoin-2015-ecosystem-grows-despite-price-decline/.

[10] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

[11] Vitalik Buterin. Proof of stake: How I learned to love weak subjectivity. URL https://blog.ethereum.org/2014/11/25/proof-stake-learned-love-weak-subjectivity/.

[12] Pedro Franco. *Understanding Bitcoin: Cryptography, Engineering and Economics*. John Wiley & Sons, 2014.

[13] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. 2014. URL http://gavwood.com/paper.pdf.

[14] Ripple. URL https://ripple.com/.

[15] Jae Kwon. Tendermint: Consensus without mining. 2014. URL http://tendermint.com/docs/tendermint.pdf.

[16] Vlad Zamfir. Introducing casper "the friendly ghost". URL https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/.

[17] David Mazières. The stellar consensus protocol: A federated model for internet-level consensus. URL https://www.stellar.org/papers/stellar-consensus-protocol.pdf.

[18] Solidity. URL https://solidity.readthedocs.org/en/latest/.

[19] What is the bitcoin block size debate and why does it matter? URL http://www.coindesk.com/what-is-the-bitcoin-block-size-debate-and-why-does-it-matter/.

[20] David Schwartz, Noah Youngs, and Arthur Britto. The ripple protocol consensus algorithm. *Ripple Labs Inc White Paper*, 2014. URL https://ripple.com/files/ripple_consensus_whitepaper.pdf.

[21] Karl J O'Dwyer and David Malone. Bitcoin mining and its energy footprint. 2014.

[22] Guy Zyskind, Oz Nathan, and Alex Pentland. Enigma: Decentralized computation platform with guaranteed privacy. *arXiv preprint arXiv:1506.03471*, 2015.

[23] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 459–474. IEEE, 2014.

[24] Bitcoin Blockchain Size. URL https://blockchain.info/it/charts/blocks-size.

[25] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments. Technical report, 2015. URL https://lightning.network/lightning-network-paper.pdf.

[26] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security*, pages 436–454. Springer, 2014.

[27] Coindesk. How to store your bitcoins - bitcoin wallets. *CoinDesk*. URL http://www.coindesk.com/information/how-to-store-your-bitcoins/.

[28] Applied blockchain. URL http://appliedblockchain.com/.

[29] Banca d'Italia. Comunicazione del 30 gennaio 2015 - valute virtuali. URL https://www.bancaditalia.it/pubblicazioni/bollettino-vigilanza/2015-01/20150130_II15.pdf.

[30] European Central Bank. Virtual currency scheme - a further analysis. 2015. URL https://www.ecb.europa.eu/pub/pdf/other/virtualcurrencyschemesen.pdf.

[31] New York State Department of Financial Services. Regulations of the superintendent of financial services - Part 200. virtual currencies. URL http://www.dfs.ny.gov/legal/regulations/adoptions/dfsp200t.pdf.

[32] Credits testing KYC blockchain on the Isle of Man, September 2015. URL http://www.ibtimes.co.uk/credits-testing-kyc-blockchain-isle-man-1520923.

[33] Stefan Thomas and Evan Schwartz. A protocol for interledger payments. URL https://interledger.org/interledger.pdf.

[34] New DTCC White Paper Calls for Leveraging Distributed Ledger Technology to Solve Certain Long-Standing Operational Challenges | DTCC. URL http://www.dtcc.com/news/2016/january/25/new-dtcc-white-paper-calls-for-leveraging-distributed-ledger-technology.

[35] R3CEV. R3's distributed ledger initiative grows to 42 bank members and looks to extend reach to broader financial services community. URL http://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/t/56727a13e0327c81fa9361ab/1450342931449/PRESS+RELEASE+R3+distributed+ledger+initiative+grows+to+42+bank+members+and+extends+reach+FINAL+.pdf.

Websites last checked: January 2016