Robust Know Your Customer using Distributed Ledger Technology

Matúš Drgoň

MInf Project (Part 1) Report

Master of Informatics
School of Informatics
University of Edinburgh

2020

Abstract

The Know Your Customer (KYC) is an expensive process financial institutions need to execute to start conducting business with a new customer. Distributed ledger technology (DLT) offers improvements in the way this process is executed and can reduce operational costs this process incurs. Whilst it could positively impact financial firms, their customers and the regulator, there are security challenges that need to be faced in order for this technology to be put in practice for this process. In this work we propose a DLT-based KYC solution that effectively deals with these security challenges. Our solution simultaneously increases security of the system and reduces operational costs incurred by the KYC process in comparison to the currently used KYC procedure mechanism. There is naturally a trade-off between the operational costs reduction and improved security of the system. We give a smart-contract implementation of this solution in Solidity programming language and provide a mathematical analysis of our model to quantify the benefits it presents.

Acknowledgements

I would like to thank to my supervisor, Aggelos Kiayias, for his precious guidance and feedback throughout this project, and to Lamprini Georgiou, a research assistant at the Blockchain Technology Laboratory at the University of Edinburgh, for her valuable advice on the legal side of this project.

Table of Contents

1 Introduction 8
1.1 Research objectives 8
1.2 Regulatory background 8
1.3 Financial incentive to optimise KYC process 10
1.4 Current KYC 11
1.5 Benefits of using DLT-based KYC 11
2 Related work 13
3 Distributed Ledger Technology Background 15
3.1 Public and private blockchain 17
3.2 Ethereum blockchain 18
3.3 Hyperledger Fabric blockchain 18
3.4 Corda blockchain 19
3.5 Our blockchain pick 19
4 A Robust DLT-KYC system 21
4.1 Basic DLT-KYC 21
4.2 The brittleness of the Basic DLT-KYC 23
4.3 Our Solution 24
4.4 Role of the Central Authority 27
4.5 Meeting KYC conditions 27
4.6 Implementation details 32
5 Security Analysis: DLT-KYC Robustness 34
5.1 Mathematical background 34
5.2 Results of the analysis 36
6 One-Tier problem 40
7 Future Work 43
8 Conclusion 45

Bibliography **47**

1 Introduction

Know Your Customer (KYC) is a process that a company needs to execute to verify a potential new customer before they can start conducting business with them. This process occurs in financial sector and is a costly challenge due to tight regulations and complexity of the process. An institution executing this process needs to establish identity of the customer, verify that source of the customer's funds is legitimate and assess potential money laundering risks associated with the customer [31].

The traditional approach to deal with tighter regulations and increased complexity of the process was to increase staffing. The increased KYC costs 'make this operating model no longer sustainable' [45]. On the other hand, executing the KYC adequately with respect to a legislative framework is for financial institutions one of the top priorities. When a financial institution fails at some aspect of executing the KYC process and on-boards a malicious customer who uses their account for illicit activities, such as money laundering or for financing terrorist activities, the financial institution may face legal consequences. These normally represent lawsuits and a fine the institution has to pay. This negatively affects the institution's reputation that has a further adversary impact on its business.

1.1 Research objectives

This thesis brings the following research objectives:

- Explain KYC compliance and significance of the regulatory framework.
- Introduce the current KYC process, indicate incentive to use the distributed ledger technology (DLT) and describe benefits it would bring.
- Establish and give a smart-contract implementation for a DLT-based KYC solution which utilises full sharing of executing the KYC process. We call this model *Basic DLT-KYC*. Identify security flaws in this process.
- Establish and give a smart-contract implementation for an improved DLT-based KYC solution that deals with the security flaws of the Basic DLT-KYC and builds on top of it. We name this model *Robust DLT-KYC*.
- Give a mathematical model describing security of the Robust DLT-KYC and provide results of the mathematical analysis.

1.2 Regulatory background

The origins of KYC date back to the Patriot Act 2001[1] that first introduced regulations for this process. Title III of the Patriot Act specifies two KYC requirements: the Customer Identification Program (CIP) and Customer Due Diligence (CDD). The CIP process identifies the customer and verifies that they are a real person. The customer might be asked to provide documents such as passport, national identification

[1] https://www.justice.gov/archive/ll/highlights.htm

card or driver's licence. The CDD process is more complicated as its objective is to estimate what type of transactions the customer will be conducting and what level of risk they will represent to the institution. This makes it easier for the institution to track suspicious transactions as well as dedicate a considerate level of monitoring of the customer's transactions depending on the level of risk the customer possesses. The customer might be asked for additional information, such as the purpose of the account, financial statements, banking references, description of business operations et cetera [57]. For instance, before a financial institution starts operating with a politically exposed person, it is recommended that this is checked by the senior management of the institution [15].
Regulations for KYC process are closely related to regulations regarding Anti-Money Laundering (AML) and Counter Financing of Terrorism (CFT). The Patriot Act 2001 was influenced by several previous acts, perhaps the most notable ones being Bank Secrecy Act of 1970[2] and Money Laundering Control Act of 1986[3]. The Bank Secretary Act was 'designed to help identify the source, volume, and movement of currency and other monetary instruments transported or transmitted into or out of the United States or deposited in financial institutions' [23, 22]. It required banks to report cash transactions over $10 000 and properly identify people conducting such transactions. The Money Laundering Control Act of 1986 was the first regulation that introduced money laundering as a federal crime and was the building block for further legislation in this field.

The connection to KYC is that these regulations, and subsequent regulations in this field, all require proper verification of a customer before a financial institution can open an account for this customer and start conducting business with them. By thorough verification of the customer, the risk associated with illicit activities, such as money laundering or financing of terrorist activities, is mitigated.
While these regulations have attracted most attention in the US, there are also international regulations related to money laundering and KYC, such as Anti-Money Laundering and Countering Financing of Terrorism Act 2009 [35] and international recommendations for good KYC practices [20].
The regulations are strict and complex due to the serious adversary impacts money laundering, financing of terrorist activities, and other illicit activities would bring if the regulations were not in place. There is an equally strong incentive for the financial institutions to conform to these regulations. The Basel Committee on Banking Supervision [15] identified that 'inadequacy or absence of KYC standards can subject banks to serious customer and counter-party risks'. It identified the following possible risks:

- Reputational - a significant thread to financial institutions, as the nature of this industry requires strong level of trust from customers' side in the institution. This risk is defined as 'the potential that adverse publicity regarding a bank's business practices and associations, whether accurate or not, will cause a loss of confidence in the integrity of the institution.'

[2] https://www.irs.gov/businesses/small-businesses-self-employed/bank-secrecy-act

[3] https://www.congress.gov/bill/99th-congress/house-bill/5077

- Operational - defined as 'the risk of direct or indirect loss resulting from inadequate or failed internal processes, people and system or from external events'. This risk relates to improper verification of the customer and inadequate execution of customer due diligence. If it is perceived by the public that a financial institution fails at executing its internal processes accurately, this might present further disrupt on the institution's business.

- Legal - is defined as 'the possibility that lawsuits, adverse judgements or contracts that turn out to be unenforceable can disrupt or adversely affect the operations or condition of a bank'. It can be perceived as a direct consequence of operational risk due to an improper execution of customer verification and due diligence. As a consequence, financial institutions can get into lawsuits and face expensive fines. For instance, in 2019, the Financial Conduct Authority fined Standard Chartered Bank £102.2 million for AML breaches in areas of its business [21]. In global, fines related to poor money laundering controls issued in 2019 accounted for a total of $8.14 billion [25]. Over the past 17 years, AML fines averaged a sum of $155m and median of $2.8m [41]. According to Wayne Johnson, the CEO of Encompass - a company providing KYC solutions: "Since 2015, annual AML penalty figures have been steadily rising each year. Multi-million dollar fines have been commonplace for a while, but we are now seeing more penalties of one billion dollars or over, with two in 2019 alone." While it is hard to analyse the exact causes of the circumstances that lead to these lawsuits and fines, KYC process is a significant factor that, when executed thoroughly, mitigates the aspects that would escalate to scenarios with these legal consequences.

1.3 Financial incentive to optimise KYC process

According to a survey conducted by Thompson Reuters in 2016, a financial firm would annually spend on average $60 million on KYC and this process would take 24 days. For larger businesses, the annual costs associated with KYC would represent as much as $500 million. The length of this process is another problem associated with this process. Out of the surveyed customers, 89% responded they had negative experience with this process and 13% decided to change their relationship with the financial institution [46]. The complexity and costs of this process have been rising. In 2017, financial institutions with a revenue of $10 billion or more spent annually $150 million on KYC-related procedures, which represents an increase of $8 million in comparison to the previous year. Average execution time of this process also rose to 26 days from 24 days in 2016 [47]. According to equity research by Goldman Sachs, designing a solution with a single proof of KYC that is compliant with Anti-Money Laundering regulations could annually save across the industry $\$3 - \$5$ billion [30].
In addition to the potential negative consequences a financial institution would face if it did not execute the KYC process adequately, it also faces a direct incentive to minimise the operational costs associated with this process. We propose a new solution that could decrease financial costs incurred from executing this process and speed up the process to increase customers' satisfaction rate.

1.4 Current KYC

KYC process begins when a customer wants to work with a financial institution. The customer and the financial institution define and agree on a set of rules that specify their relationship. The customer consequently sends personal documents to the financial institution that verifies them. As per the Customer Identification Program (CIP) part, this could be either a passport, an ID card, or some other official state document. This uniquely identifies the customer and provides information about the customer, such as their name, date of birth, nationality, address etc. In order to execute Customer Due Diligence (CDD), additional information about the customer, such as previous bank statements, references etc. could be required from the customer. This would naturally depend on the type of customer and the level of risk this customer could potentially represent. After reviewing the relevant documents from the customer, the financial institution then decides to either accept or decline the customer. If accepted, the customer and the financial institution may start conducting business.

It is important to note here that one of the challenges is that the KYC process might be executed differently by distinct financial institutions. The process is also dependent on the regulatory framework which varies across countries with different legislation. However, certain aspects of the KYC process, both in CIP and CDD part, are executed the same way. Parra-moyano et al. [42] introduces the *core* KYC process. This represents the minimal KYC verification, as required by the regulatory framework, that all financial institutions conforming to this legislative framework need to execute. We can split the KYC process into its *core* part and additional checks and controls of a customer, that are specific to a financial firm executing the verification process.

KYC process (as a whole) is currently executed for a customer individually by each financial institution this customer would like to operate with. If we define the average cost of this process for a customer to be c, and the number of financial institutions the customer is operating with to be n, the overall cost these multiple KYC processes would represent is $c * n$.
The basic DLT-KYC process, introduced in 4.1, proposes that the core KYC is executed only once and its result is shared across all FIs that operate with the customer. The process is only executed by the first FI to operate with the customer and is equally shared by all FIs that operate with the customer over time. This solution provides for an overall cost capped at c, independent of how many financial institutions operate with the customer. For a single FI, the KYC operational costs would decrease from c to c/n, where n is the number of all financial institutions operating with the customer. While this scenario provides for minimising the cost associated with KYC, it puts trust of all financial institutions in the first financial institution that executes this process. We analyse this in more detail in our Robust DLT-KYC (4.3) and in 5.1.

1.5 Benefits of using DLT-based KYC

We have outlined the financial incentive to optimize the KYC process and how this could be achieved by DLT - the main impetus for implementing this technology. There are additional advantages that DLT presents for the core KYC process, summarized in

the following list:

- Increased security. When FIs execute the KYC for a customer independently, and share the results using DLT, the overall scheme is more secure. The probability that a FI overlooked certain aspect of the process, and was notified about this later, is mitigated by having another institution do this process independently. This is thoroughly described in 5.1.
- Improved screening. [45] identifies that '.. a unique ID created for a client as part of the identification and verification process could reduce false positives by accurately identifying the entity or individual. It could also connect a greater number of sources of information relating to an individual client to build a richer picture of their behavior and relationship network, potentially uncovering hidden risks.' This criterion is important because we would be able to tell more about a customer's behavior due to using shared information between the institutions operating with the customer, while maintaining private identity of each institution. In section 4.3, we introduce the idea of a customer rating to achieve this benefit. It provides a way to gather anonymous feedback on a customer from financial institutions operating with the customer. This feedback is shared between the financial institutions. The customer does not have access to this feedback. This can help identify what level of risk a customer presents and track it over time in case the customer's behavior unexpectedly changes.
- Increased prevention against money laundering and financing of terrorism. If a customer is revealed to have used his/her account for illicit activities, such as money laundering or financing terrorism, the distributed system enables the regulator to quickly report this to other institutions operating with that customer. Financial institutions can promptly react to this by freezing account of the customer. It enhances collaboration between the regulator and FIs which helps combat money laundering and financing of terrorist activities.
- Increased customer satisfaction. The results of the core KYC process are shared between the FIs operating with the customer, which is likely to lead to a decrease in the average waiting time on this verification process.

Parra-Moyano et al.[42], Valkanov et al. [55] and a study conducted by Refinitiv [45] identify a possible improvement in reporting and monitoring. Creating a single profile for each customer on the blockchain can help the regulator or financial institutions better track the customer's behavior. The record of customer's activities would be immutable and ordered, which can effectively help in resolving any potential issues. The increased prevention against money laundering and financing of terrorism can be perceived as one aspect of improved monitoring and reporting. Additional benefits of reporting and monitoring of the customer would be brought if the DLT were also used for tracking future transactions between customers and financial institutions. Whilst blockchain can be leveraged by financial institutions to improve tracking of customers' transactions [10, 34], each institution would use this internally and would not share the blockchain with other financial institutions. Delivering a DLT-based solution that would achieve this across financial institutions would rise many privacy challenges. We have not found a study that would specify how the technology could be used for

this purpose. We acknowledge the benefits it could bring may be significant, but it is beyond the scope of this study - which is to use DLT only for the KYC process - and is not included in the list.

A natural question that might arise is whether these benefits could not be brought using some other technology and what makes the DLT so suitable for this use-case. The DLT enables financial institutions to agree on a set of given rules, the legal framework specified by the regulator, and conform to these rules while maintaining anonymity. Anonymity of financial institutions is a necessary condition. Revealing any information about an institution operating with a customer are highly undesirable for both parties. By conforming to the same regulatory framework, all institutions leveraging the DLT follow specific rules they agreed on without having to explicitly trust each other. The DLT also achieves equality in the institutions' view of the blockchain which provides good grounds for ensuring that no financial institution would be put in an advantageous position in comparison to other institutions.

2 Related work

The DLT, due to its immutability of records and distributed nature, could "deliver to the digital world a new level of objectivity and trust that even known reputable trustees will not be able to match" [8]. It is debatable to what extent this claim applies to the financial sector, where the level of trust financial institutions require from their customers is very high, but it accurately recognizes that immutable records is one of the core aspects in finance and for KYC.

Britton et al. [11] outline that an efficient solution for KYC would allow each financial institution access to customer data that would be stored in a distributed database - a solution based on the distributed ledger technology. It identifies that the advantage of using blockchain is the digital identity it would enable for each customer. The digital identity would be assigned to each transaction associated with this customer and would store all relevant information about the customer which could be used during AML/transaction monitoring and increase efficiency of AML checks. More specifically, it could decrease the false positive rate of AML checks, which is currently ¿99% and brings a lot of inefficiency to the regulator. It further outlines that a DLT-based solutioon would enhance customer experience, reduce operational costs for the financial institutions, increase security and transparency for regulators.

The Hong Kong Monetary Authority[4] researched the use case of KYC for the distributed ledger technology and identified that it could improve customer experience and cut down operational costs incurred by the process for financial institutions due to avoiding repetitive tasks. This refers to the KYC checks that currently have to be executed by each financial institution separately and represent a lot of repetitive work. Their whitepaper [5] further outlines the following challenges of such a solution: cyber security, legal and regulatory requirements. The DLT-based solution would have

[4]https://www.hkma.gov.hk/eng/

to be executed in cooperation with regulators to comply with a legal framework and ensure customer privacy. The solution has to ensure that ”cyber attacks will not result in network damage and data loss.”
While there is a general consensus on the direct benefits of using the DLT for the KYC process for customers, financial institutions and even the regulator, there is a lack of work done that would outline specific steps how this could be achieved.

Pachaiyappan et al. [40] introduce a smart-contract implementation of the KYC process on Ethereum blockchain. It very well describes what the KYC process consists of and outlines the possible use of Corda blockchain by R3[5] instead. However, the contract length is rather short, it lacks sufficient analysis of how the contract ensures customer privacy, distribution of the cost, how it prevents possible contract manipulation by financial institutions or third parties, how the institutions should operate with it etc.

Sinha et al. [51] propose a system architecture with a sample smart contract utlizing IPFS [6] on Ethereum blockchain. It suggests a fully decentralized KYC system on public Ethereum network and provides a gas cost analysis of executing and operating with the smart contract. While this implementation can be used a base for further work, it lacks sufficient argumentation on the same issues outlined for [40]. The problem with the public Ethereum network is that it is accessible by anyone on the internet and simply cannot be used for this use case due to the sensitivity of information. It would breach customer privacy which is one of the core aspects of financial world.

Parra-Moyano et al. [42] offer the most complex study on this topic. Their work was executed in close collaboration with financial experts from Nordea Bank[7] and when presented at the hackathon organized at the IT University of Copenhagen won the first prize. It proposes a design for how the KYC process can be executed using the DLT and outlines a more centralised, as well as a fully decentralized solution. It provides an architecture of the system, but does not provide a specific implementation.
The paper outlines the following conditions that are equally important and apply to our implementation too:

- Proportionality - it is necessary to distribute the costs between all financial institutions equally. The cost of executing KYC process should not bring any financial institution an advantage or a disadvantage.
- Irrelevance - it should not be relevant which institution executes the core KYC process. A financial institution should not have incentive to either execute core KYC, or not execute KYC and let another institution do so.
- Privacy - a financial institution operating with a customer should not be aware of which other institutions are operating with this customer. Financial institutions often compete for customer’s monetary funds. It is important to respect privacy of the financial institutions operating with a customer, as revealing this informa-

[5] https://www.r3.com/
[6] https://ipfs.io/
[7] https://www.nordea.com/en/

tion could negatively impact both the institutions whose identity was revealed, as well as the customer who was affected by this.

- No-minting - this condition ensures that a financial institution cannot simulate having executed the core KYC verification without actually doing so. This is essential as it could jeopardise the trust other institutions have not only in the customer and between each other, but also in the whole solution.

Parra-Moyano et al. [43] introduce a research paper based on the previous version by Parra-Moyano and Ross [42]. The new research paper outlines several aspects that can be improved in the original paper and proposes an smart-contract implementation in Solidity programming language on Ethereum blockchain. The first aspect outlines the need of a trusted third party (TTP) that has to periodically check that the financial institutions have paid their proportion for executing the core KYC to keep the system fair. The second aspect identifies the possible need to change status of the customer in a decentralized manner. The third aspect identifies the need to update the KYC process over time due to possible changes in the legal and regulatory framework. The last aspect concerns the storage of customers' documents as their original paper proposed a complex database that would need to be maintained by the regulator and introduce a high cost.

3 Distributed Ledger Technology Background

DLT is "a type of database that is spread across multiple sites, countries or institutions. Records are stored one after the other in a continuous ledger, rather than sorted into blocks, but they can only be added when the participants reach a quorum" [56]. DLT is stored across several locations, also called nodes. The nodes can both act as clients, operating with the DLT, and be responsible for maintaining the state of the database. The records refer to transactions that take place in the network and are stored on this ledger. The transactions are grouped into blocks which form a continuous chain. This describes the notion of *blockchain*.

The definition of DLT further states that the participants need to reach a quorum. This is a communal consensus specifying the state of the ledger. It ensures the ledger has a unique state and all participants operate with the same ledger.

The terms blockchain and distributed ledger (technology) are often used interchangeably. Distributed ledger and blockchain are both a subset and a particular example of the DLT [7]. The main distinction is that unlike blockchain, the records stored on the distributed ledger do not need to be structured in blocks [9]. The underlying property shared between the distributed ledger technology and blockchain is that they both are a type of distributed database spread across multiple participants. All operational benefits 1.5 and technological qualities of the distributed ledger technology 3 equally apply to blockchain and thus our implementation. From now on, we will use the terms DLT, blockchain and distributed ledger interchangeably unless explicitly said otherwise. They all will refer to an instance of the DLT where transactions are stored in blocks that form a chain.

The DLT is based on a *peer-to-peer* (P2P) [6, 49] architecture which makes it rather different from a standard centralised *client-server* [39] architecture. We illustrate the key difference by the following analogy. Please note that this analogy does not try to propose a different solution for this existing system, it only uses this famous example to outline the fundamental difference between the two architectures.

If we execute a simple google search, our browser needs to send a request to a server controlled by Google. This request is processed by the Google's server that eventually sends a response to our browser that renders it. Google acts as the central authority that is in charge of its servers. As a client, we can only access the Google's servers in a limited way - for instance, we can query them and retrieve the results. We are not responsible for a functional Google service. For instance, we are not expected to and cannot participate in giving responses to other clients' requests. There is a clear distinction between us, the clients, and Google, the (central) service provider.

In the case of a distributed ledger, we can use the distributed database as a client, who wants to retrieve some information or perform a transaction, but also as a service provider, who is responsible for maintaining the state of the database. In our analogy, this is as if we were able to use Google both as a client for ordinary searches, and to operate on the server side and take responsibility for adequate functioning of the system. The notion of a central authority would diminish and it would be responsibility of the participants on the network (the distributed ledger) to provide the service and maintain its functional and correct state.

Paraphrasing [36], "users can use the distributed ledger (blockchain) for online transactions that are rendered immutable and transparent, yet resistant to censorship and manipulation due to the technology's cryptographic and distributed foundations." The transactions are used for exchange of goods and services, or simply to store money on the ledger in a currency it offers. These technological qualities are the key to meet our KYC conditions outlined in 2 and operational benefits of the DLT outlined in 1.5. Let us have a closer look at each of these:

- Transparency - ensures that once a transaction is on the ledger, any entity that has access to the ledger can view meta-data of this transaction. This is the building stone for improved monitoring, screening and reporting of transactions and clients on the network. The meta-data includes information such as the source, target and timestamp of each transaction.

- Resistancy to censorship - ensures that each party that should have an access to information about transactions has that access in reality. This means that the ledger is transparent to each party on the network and two parties with the same role on the ledger, such as two clients, have equal views and possibilities to operate with it.

- Immutability/Resistancy to manipulation - ensure that once a transaction is on the ledger, it cannot be tampered with or manipulated in any way.

3.1 Public and private blockchain

Blockchain systems introduce a trade-off between decentralization, security, and scalability. The trilemma problem identifies that a blockchain can only have two out of the three [1]. Blockchain has two fundamental categories - it can either be public or private.

Public blockchain enables anyone from the internet become a participant and read the blockchain's content, use it to carry out transactions and also participate in the process of creating the consensus. It assumes that a participant can do so at any time without providing any form of identification or asking for permission [32]. This accounts for a high-level of anonymity on the ledger. Based on the meta-data of a transaction, it is possible to identify the public key of a person executing the transaction, but the real identity behind this public key is not revealed and practically almost impossible to unravel. A public ledger is a fully decentralised system.

Due to the fact that a participant is not verified in any way to join the blockchain, it might have malicious intentions and could try to manipulate the blockchain. The security of a public blockchain is ensured due to the cryptographic foundations and distributed consensus based on incentive mechanism. This incentive mechanism gives a reward to parties on the blockchain that participate in creating the consensus while conforming to the rules and acting fairly. These parties have to solve a cryptographic puzzle that requires a lot of computational effort. Hence, they are also referred to as *miners* [38].

Private (or permissoned) distributed ledgers are not accessible by a third party from the internet unless they are given a permission. A private ledger "assumes that all actors on the network are known and trusted; belonging to a controlled membership" [32]. This permission can be given by a central authority or consortium of parties that already operate on the ledger. There are requirements that a party needs to fulfil before becoming a client or service provider on the ledger. There might be additional roles, dependent on the specific implementation of the ledger, each serving a different purpose and with various responsibilities and permissions. Security of the private blockchain is enhanced, as only a party that is identified can be given a permission to join the ledger. It comes at the cost of lacking the fully decentralised nature and anonymity the paries have on the ledger. The ledger still operates in a distrbuted manner, but it is no longer fully decentralised due to the need of central authority or consortium. The anonymity can be accomplished between the clients operating on the ledger, but the client is no longer anonymous to the central authority.

The consensus process may only need to be achieved by a limited, predefined number of participants, instead of being achieved by each party on the ledger, as in a public blockchain. This means a private ledger might not require mining, proof of work and remuneration for adding blocks to the chain [26]. This improves the blockchain's scalability and the rate of transactions that can be executed on the ledger. The private blockchain can also be configured with specific parameters, such as the time passing between two consecutive blocks, the size of blocks, the hardware of the nodes running the blockchain software, or simply the size of the network [48]. Altering these parameters for a specific use-case further makes the private blockchain more efficient.

3.2 Ethereum blockchain

Ethereum [13, 58] is a public blockchain with a built-in Turing-complete programming language that allows anyone to write smart contracts and decentralized applications that can be built on top of this blockchain. The smart contracts and decentralized applications can define their own arbitrary rules for ownership, transaction formats and state transition functions. Similar to bitcoin blockchain, Ethereum is based on a public-key infrastructure and remains secure due to the cryptographic standards and consensus protocol [24].

A decentralized app (dapp) is a mobile or web application that has access to a blockchain through a single node that is on the blockchain. The node has a copy of the blockchain and when the dapp makes a request, such as to execute a transaction, it is propagated through this node on the blockchain where the transaction is stored. The application itself is running on a single device and is not distributed, giving the user same experience as in case of a standard centralized application [4].

A smart contract can be envisioned as a real contract that is deployed on the blockchain in form of code that describes this contract's functionality. It is an "automatable and enforceable agreement" [17]. Automatable by computer, as it is digitally deployed on the blockchain, and enforceable by code that defines the contract.

Stark et al. [52] give two abstractions of how we can perceive a smart contract:

- Smart contract code - code of the contract can be viewed as a verified program that is deployed on the blockchain. Its functionality and capability is dependent on the programming language this contract code was made in [2].
- Smart legal contract - defines the contract from a legal, political and business perspective.

Ethereum is a public blockchain that anyone can join without having to obtain any permission. Each node on the ledger only reveals its public key, maintaining anonymity to any other node on the ledger. A functional DLT-based KYC system has to be implemented on a permissioned blockchain to keep customers' details safe. There are several tutorials that outline how the main Ethereum network can be forked into a private permissioned blockchain [28, 18]. This was done by JPMorgan Chase when creating *Quorum*, a private blockchain that leverages the existing codebase and support of the public Ethereum blockchain and introduces additional enhancements to make it usable for enterprise purposes [16].

3.3 Hyperledger Fabric blockchain

Hyperledger Fabric [3], further referred to as fabric, is a permissioned blockchain that can be tailored to fit various use cases. In contrast to Ethereum, it runs without dependency on a native cryptocurrency and applications. Smart contracts can be written in standard programming languages which can be used by an institution that can leverage its existing code base and know-how when implementing the DLT-based core KYC.
Fabric introduces *modular consensus protocols* that come with a concept that enables

a consensus protocol to be chosen based on a use case, in contrast to using a single uniformed consensus protocol. A consensus protocol used in a public blockchain, such as Ethereum or Bitcoin, falls into category referred to as *byzantine fault tolerant* (BFT) [33] protocol. A BFT consensus protocol provides the decentralised nature of a public blockchain, but requires a lot of computing effort. It is operationally less efficient. It requires consensus across the whole network, which limits the rate of transactions that can be executed on the blockchain. There is no ”one size fits all” BFT protocol that could be applied to any use-case of blockchain [50]. The modular consensus protocols enable a consensus protocol to be chosen based on a particular use case, which makes it appealing to our implementation.

In the context of financial services, such as for executing the KYC process, *transaction finality* is an essential requirement. Transaction finality means that once a transaction is successully executed, it will inevitably appear on the distributed ledger. This does not immediately hold true for a public blockchain and is one of the obstacles for their usage by enterprises [53].
Fabric proposes the following types of nodes in the blockchain: clients, peers and orderers. Clients represent the end-users and submit transactions. Peers commit transactions and maintain the state and a copy of the ledger. They receive ordered update messages from orderers for committing new transactions to the ledger. A peer can take up a special role called endorser. An endorser can endorse a transaction before it is committed to the chain. The last role, orderer, provides a communication channel with a delivery guarantee. The communication channel is shared between clients and peers and can be used for broadcasting messages containing transaction information.
Clients only see the messages and associated transaction on channels they are connected to. Restricting access to transactions to only involved parties means that consensus needs to be reached only at the transaction level, instead of the ledger level. This means that the consensus is reached by only parties associated with the transaction, in contrast to consensus that would have to be reached by all parties on the blockchain. It increases transaction flow and makes Fabric more scalable [54, 19].

3.4 Corda blockchain

Corda [12] is a permissioned distributed ledger platform designed for financial services industry. A node on the ledger does not see contents of the entire ledger, but has only access to transactions it is a part of. The fundamental buidling block is a state object, represting a specific instance of a an agreement, which may be thought of as representing a real-world contract or section of a contract. State object is a digital document which records the existence, content and current state of an agreement between two or more parties. It is shared only between those who have a legitimate reason too see it.

3.5 Our blockchain pick

We decided to select Ethereum blockchain for implementing the core KYC process using distributed ledger technology for the following reasons:

- It is an open source project with strong support and a lot of available resources.

While Fabric and Corda are also open-source projects, Ethereum has a longer history and the strongest developers' community. There are several frameworks (e.g. Remix, Truffle) that can be used for easy implementation, deployment and testing of the smart contracts. This gives it an advantage over Hyperledger Fabric and Corda that seem promising, but currently cannot compete with the level of existing infrastructure for Ethereum.

- It is usable for enterprise environment. We acknowledge that the main Ethereum network is a public ledger and cannot be simply used in an enterprise environment, but it is possible to fork it and create a private Ethereum blockchain based on the public main network, leveraging the already existing code-base. There are several tutorials that outline how this can be achieved [28, 18]. JPMorgan Chase did this when creating Quorum, a private blockchain that combines the public Ethereum and introduces additional enhancements to make it usable for their internal purposes [16].

4 A Robust DLT-KYC system

This section starts with a basic DLT-KYC system inspired by [42]. We modified architecture of the solution and provide a smart-contract implementation that was missing in the research paper. We outline a security flaw, called the brittleness, in this scheme and introduce our improved solution that effectively tackles this flaw. A smart-contract implementation and architecture of the solution is again provided. We define minimal requirements that the central authority needs to meet in our solution and explain how our solution fulfills conditions outlined in 2. Finally, we give a little insight into our smart contract implementation of the improved DTL-KYC system.

4.1 Basic DLT-KYC

When a customer approaches the first financial institution, he/she needs to provide the institution with necessary documents that the institution needs for executing the core KYC process. The institution stores these documents in its local database and executes the core KYC. The institution now needs to register on the blockchain, using an interface provided by the central authority (CA). When doing so, the CA stores the identity of the institution in its private database. The institution needs an account to operate with the customer on the blockchain. The number of accounts an institution can open is not limited, and it is recommended that the institution opens a new account for each customer it is interacting with. The CA's private database stores the institution's real identity behind each account it opens on the blockchain. This information is only available to the CA and each account appears anonymous to other parties on the blockchain.
When the institution decides to on-board the customer, it creates a simple customer profile on the blockchain. This profile serves as a digital identity of the customer and is specified by a unique ID. This profile is only created after the core KYC process was executed, complying to the regulatory framework. The institution gives the customer his/her ID and the two entities can start conducting business together.
Note that the institution only had to execute the *core* KYC process. We mentioned in 1.4 that each institution might want to execute the KYC process differently, with a various level of complexity. The core process that this solution simulates is the basic legal requirement on the customer verification and due diligence. The institution might want to require additional documents and background checks on the customer and store these in its local database. This is decided by the institution and is executable without any interaction with the blockchain.
The digital identity of the customer does not reveal any personal details about the customer. The blockchain only stores the number of institutions the customer is operating with and hash of the customer's documents that were used during the core KYC. These documents are also referred to as document package. The described process is shown in the figure below.

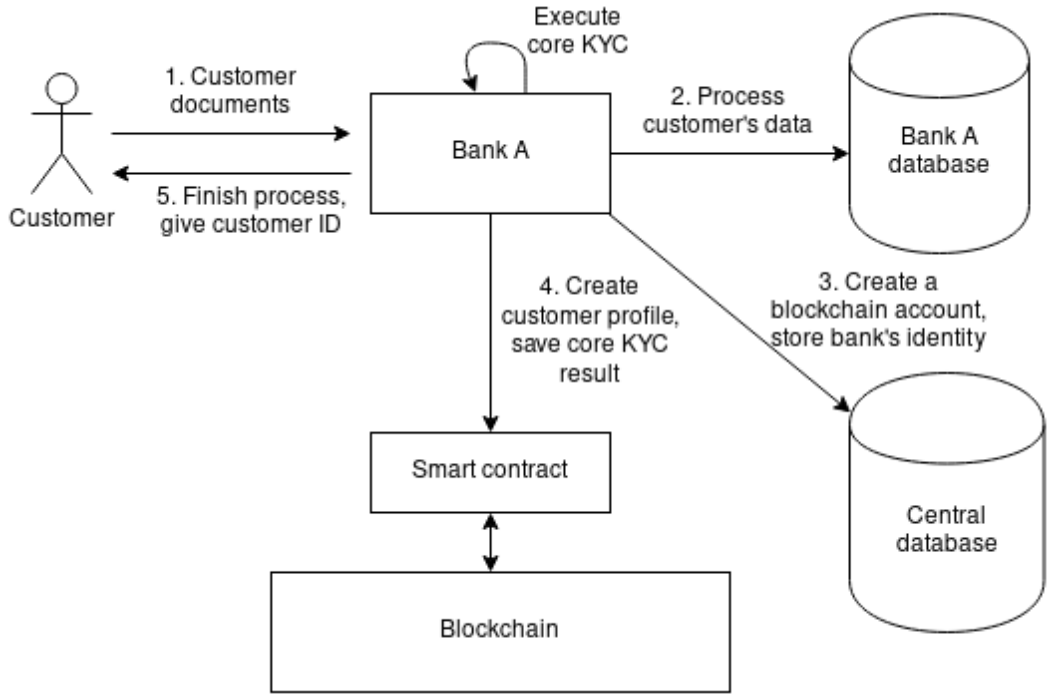

Figure 1: Customer approaches the first financial institution. The central database and blockchain are in control of a trusted third party, such as the regulator.

When the customer wants to operate with another financial institution (`Bank X`), the core KYC process does not need to be repeated because it was executed by Bank A. The customer provides `Bank X` with his/her customer ID. In case the customer forgets their ID, they can always retrieve it from any financial institution they have been conducting business with. `Bank X` can retrieve from the blockchain how many institutions are already operating with this customer. This assumes that `Bank X` already operates on the blockchain. If it does not, the institution would firstly need to be verified by the CA and obtain a permission to access the blockchain. The institution pays an appropriate fee that keeps the system fair and is added to the list of institutions the customer is operating with.

The regulatory framework could develop and change over time. When a new legislative requirement is introduced into the process, the core KYC process for the customer has to be updated. `Bank X` then needs to obtain required customer documents, update this process and store the updated version on the blockchain. Naturally, this update incurs additional cost. To keep the system fair, this cost is also equally distributed by all financial institutions operating with the customer. This procedure is explained in more detail in section 4.5.1.

`Bank X` can now start operating with the customer. `Bank X` may similarly want to execute additional checks of the customer before starting to operate with them. These checks and information required from the customer would be stored in the bank's local database. We call this implementation the *basic* DLT-KYC scheme. It requires only a single execution of the core KYC process whose result is shared by all financial institutions. The figure below outlines this process.

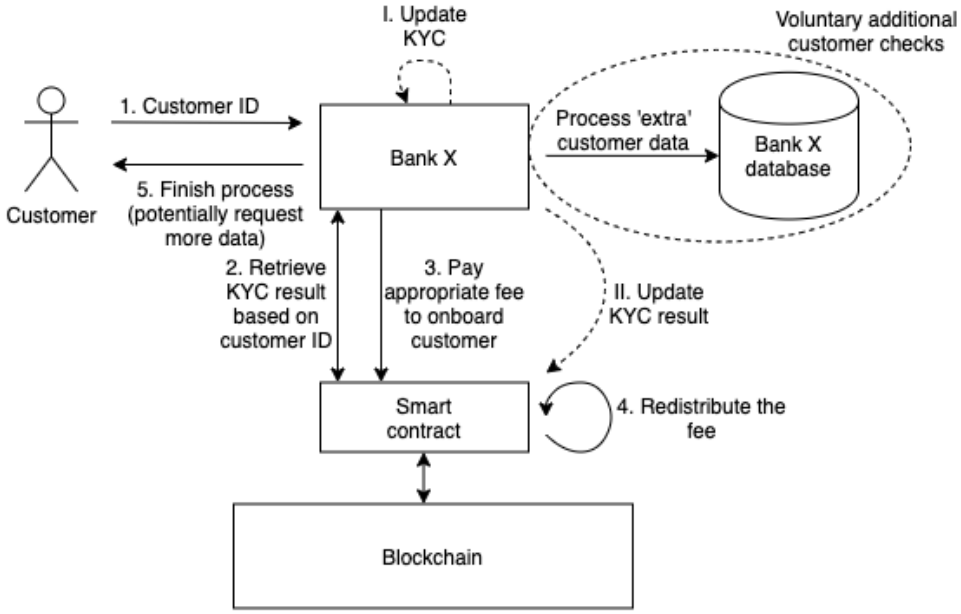

Figure 2: *Basic* DLT-KYC. Customer approaches a new financial institution. The dashed lines and roman numerals indicate a process that might have to be executed only when a legislative change was introduced in the core KYC.

4.2 The brittleness of the Basic DLT-KYC

In this section we illustrate a significant shortcoming in the basic DLT KYC scheme that, to the best of our knowledge, has so far remained unaddressed. Consider a customer who uses their account for malicious activities that are in breach of the regulatory framework. The customer may be operating with several financial institutions, having an account with each of them. Let us call the institution where the customer was using their account for these activities `Bank Z`. `Bank Z` is now exposed to the negative financial and reputational consequences that are implied by operating with this customer. The core KYC process was executed by the first financial institution this customer started operating with. Let us call this institution `Bank A`. The problem arises when `Bank A` and `Bank Z` are not the same institution. In addition, the customer can use their account at `Bank A`, and at any other institution he/she operates with, for completely legal activities. The question is whether the financial and reputational consequences should be faced by `Bank A`, that executed the core KYC process, or `Bank Z`, where the customer used their account for illicit activities.
Further steps in this direction would have be taken in close cooperation with both financial institutions and the regulator before this system could be put in practice. We outline the following possible causes for this problem:

1. A problem occurred in executing the core KYC process. The regulator knows the real identity of all institutions operating on the blockchain and can identify `Bank A`. `Bank A` is likely to face further financial and reputational impacts consequent to this issue.

2. A problem did not occur in the core KYC process and each institution operating

with this customer should have done additional controls of this customer before starting to operate with him/her. These controls should have prevented malicious activities executed by the client and the controls would not be included within the core KYC process. These controls should have been executed outside the distributed ledger. In this scenario, `Bank X` is likely to face most financial and reputational consequences.

3. The customer's illicit activities could not be prevented by the KYC procedure. This scenario is not relevant for our implementation as it does not matter whether the current KYC scheme or the proposed DLT-based would be used to on-board the customer.

Although the regulator's approach is dependent on the specifics of the situation, it is certain that mitigating this problem at the basic DLT-KYC is hihgly desirable. In fact, as it can be easily seen, it is enough for a single institution to fail in performing the KYC process diligently for the whole system to collapse! Indeed, if it becomes known that `Bank A` is not diligent enough in conducting the KYC process, it will become the single point of entry for all individuals wishing to engage in criminal activity. This points to the fact that the basic DLT-KYC system is *brittle*. We present how we rectify this problem in the following section.

4.3 Our Solution

We introduce a modified DLT-based KYC scheme which mitigates the probability of operating with a malicious customer and enhances security of the KYC process. Our modified system uses a probability with which the execution of the core KYC process needs to be repeated. When a customer wants to start operating with a new institution, `Bank X`, this institution pays the appropriate fee to on-board the customer and the fee is again redistributed to keep the system fair. In case `Bank X` does not have to repeat the core KYC, no additional action has to be taken. Otherwise, `Bank X` has to obtain the customer's documents required for the core KYC process and independently repeats the process.
When repeating the process, it is possible that `Bank X` would get to a different result based on the customer verification. It could decide not to operate with the customer, even though the customer can be already operating with several other institutions. For instance, `Bank X` could have found some transaction details that would make the customer too risky to operate with and were overlooked by `Bank A` that initially executed the process. `Bank X` can raise an alert that would inform the regulator and/or other financial institutions operating with the customer about this situation. The relevant parties would be alarmed immediately, opening up space for a more in-depth investigation if it is required. Further actions would depend on the seriousness of the situation and the reason why the two independent core KYC controls lead to different results. If necessary, these actions would be taken in cooperation with the regulator.
Note that `Bank X` might have to both repeat and update the core KYC. In this case, `Bank X` would re-execute the core KYC process according to the newest regulatory framework and update hash of the document package on the distributed ledger accordingly.

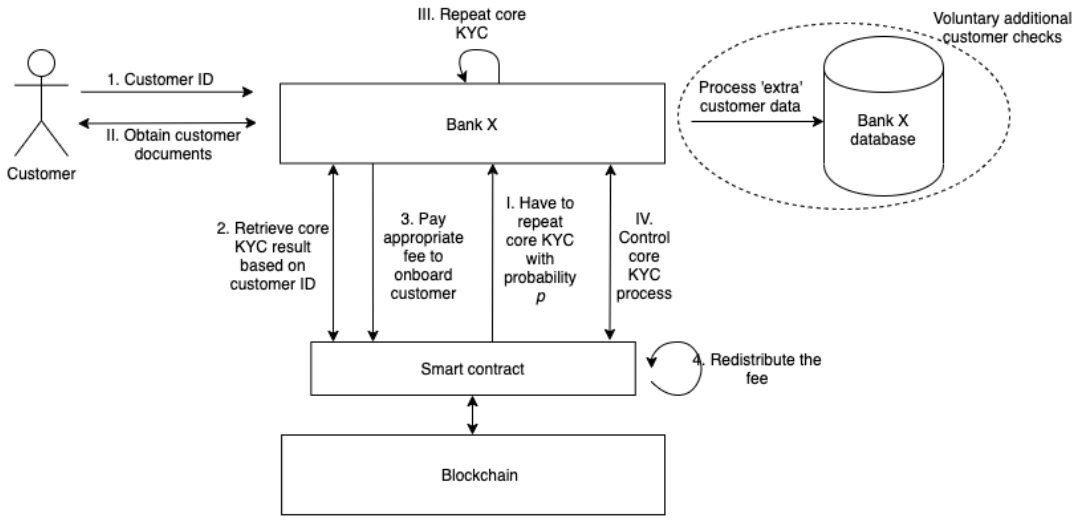

Figure 3: Customer approaches a new financial institution. This institution might have to repeat the core KYC, indicated by roman numerals.

The key aspect of this modification is that any financial institution operating with a customer might be asked to execute the core KYC process and does so independently. It does not represent an additional burden to the institutions, as they already have operating teams that currently always need to execute the process for a new customer. We provide a mathematical analysis of the impact this modification has on the security of the system in 5.1. It is important to note that in comparison to the current KYC scheme, this modification accounts for both increased security and reduced operational costs incurred by the process.

A study published by Refinitiv [45] identifies that a unique ID created for a client as part of the identification and verification process could connect a greater number of sources of information relating to an individual client to build a richer picture of their behavior and potentially uncover hidden risks. The customer's ID uniquely specifies the customer's profile. The profile provides information about the number of financial institutions the customer is operating with, but does not include further details on customer transactions or other information about the customer. Building a richer picture of a customer's behavior and assessing hidden risks the customer could possess is not an explicit step that can be currently achieved from the customer profile. This would require a DLT-based platform for recording all customers' transactions, not only transactions related to the KYC process.
Our implementation models a customer's behavior and potential risks the customer represents by introducing a customer rating scheme. Using the smart contract, each financial institution is able to rate each customer it is operating with as a form of feedback on cooperation with this customer. This rating is not available to the customer, but it is available to each financial institution operating on the blockchain. If a customer approaches a new financial institution, this institution can view the average rating of this customer as determined by other institutions operating with him/her. A very low rating value could be a reason for the institution to reject the customer. A moderately

low rating would indicate that the institution should consider additional KYC controls, outside the core process, to ensure proper customer diligence was executed before onboarding the customer. A high average rating would indicate easy cooperation and low level of risk present from the customer. The rating can be only assigned by an institution that operates with the customer. Naturally, institutions can decide to change the rating they assigned to a customer over time.

Figure 4 outlines how the new institution, `Bank X`, can view a customer profile before it decides to start operating with the customer. The customer profile includes additional fields, but the most relevant ones for determining the potential risk of operating with the customer are presented.

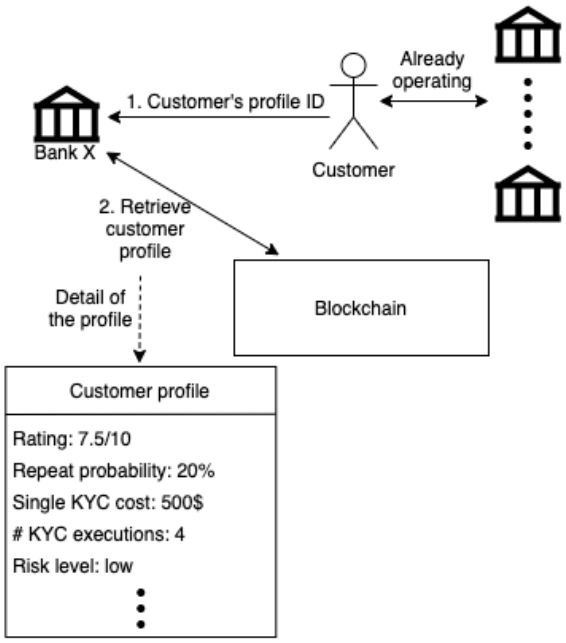

Figure 4: Customer approaches a new financial institution (`Bank X`). The institution views the customer's profile in order to see associated risk of operating with the customer. This provides grounds for further controls the institution may decide to execute.

We outlined the general benefits of using DLT-based KYC solution in section 1.5. Here, we present the unique benefits of our implementation:

- Increased security. Due to introducing the probability of repeating the core KYC process, the security of this scheme is enhanced. The obtained level of security is on average higher than the current or the basic DLT-based KYC scheme proposes. We quantify this in section 5.1.
- Improved cooperation between the financial institutions and the regulator. When a financial institution repeats or updates the core KYC for a customer, it may come across illicit past transactions or other activities of the customer. In such a case, the institution can immediately raise an alert that would notify the regulator. This brings a benefit to both the institutions and the regulator. The regulator

would be informed of a potential issue directly by an institution and could further investigate it immediately. The institutions, in case it is confirmed they were operating with a malicious customer that used their account for illicit activities, would obtain valuable time and could cancel operation with this customer immediately. The advantage is that having an additional institution independently control the customer, as is the case in our improved scheme, increases the chances of observing any potential risk the customer presents.

- Improved cooperation in between the financial institutions. There are two main indicators of a risk that customer presents: price of a single core KYC execution and the customer's rating. If a financial institution finds that a customer presents a higher level of risk than the two indicators imply, but has not found any compromising information about the customer, it can raise an alert that would notify other institutions operating with this customer of a potential hidden risk he/she presents.

4.4 Role of the Central Authority

Our solution outlined the necessity of a central authority (CA). This CA could be the regulator, but it could also be a verified outsourced external company or a consortium of the financial institutions that decide to use the proposed solution. Introducing a CA can increase initial costs of a DLT-based KYC solution and can represent a main hindrance in putting it to practice. In order to have an estimate of these initial costs, we need to define the scope of the central authority's privileges and responsibilities. We try to minimize the responsibilities of the CA to minimize the incurred cost by this and identify the following requirements from the CA:

- Entrance controls of an entity that would want to join the blockchain. Our solution is based on a permissioned private blockchain which only a verified institution could enter. As the institutions appear anonymous to each other on the blockchain, it is important that each institution present on the blockchain has been thoroughly verified.
- A database that would maintain a mapping between accounts on the blockchain and real identities of the institutions behind those accounts. This requirement ensures that the identity of any entity on the blockchain can always be obtained by the regulator when the regulator needs to proceed to any legal actions.

4.5 Meeting KYC conditions

In order for our improved smart contract to be usable, it is important that it meets KYC conditions outlined in 2.

4.5.1 Proportionality

The proportionality condition ensures equal cost distribution between all institutions operating with a customer. It does so as follows: when the first financial institution executes the core KYC process for a new customer, it has an associated cost with

this process. Let us call c the average cost of executing the core KYC process for a customer. In our repeated scheme, when a customer would like to operate with a new financial institution, `Bank X`, this institution has to repeat the core KYC with probability p_{rep}. As the process might need to be repeated, updated, or both, we create a new variable c_{agg} which is the aggregate cost incurred by executing possibly several core KYC processes for this customer. If there are k institutions operating with a customer, in order for the proportionality condition to hold, each must have had a cost of c_{agg}/k. There are four possible scenarios `Bank X` could be facing before operating with the customer.

1. Neither KYC update, nor repetition of the process is required.
2. Update the core KYC without repeating the process.
3. Repeat the core KYC without any update.
4. Repeat the core KYC and execute an update.

Let us start with the first scenario. Assume there is only one institution operating with the customer, `Bank A`, which executed the core KYC. This costed c and so the aggregate cost is the same, $c_{agg} = c$. In order to on-board the customer, `Bank X` has to pay a fee of c_{agg}/k, where k is the number of financial institutions already operating with the customer plus this new institution, `Bank X`. In this case, $k = 2$. After `Bank X` pays this fee, it is equally distributed between all of the $k-1$ institutions that were already operating with the customer. In our scenario, this means only `Bank A`. Both institutions have paid $c_{agg}/2$ and `Bank X` can now on-board the customer with no additional costs. The system is fair.

Assume now there are already $k-1$ institutions operating with the customer and that all these $k-1$ institutions have equally shared the cost of executing the core KYC. This means that each institution has paid $c_{agg}/(k-1)$. Note that aggregate cost may no longer be equal to the cost of a single core KYC. In order for `Bank X` to on-board the customer, the institution needs to pay c_{agg}/k. This new contribution of value c_{agg}/k is then equally distributed between the $k-1$ institutions that are already operating with the customer. The cost that a financial institution already operating with the customer will face after this new institution joins in is the following:

$$\begin{aligned} \frac{c_{agg}}{k-1} - \frac{c_{agg}}{k}\frac{1}{k-1} &= \frac{c_{agg} * k - c}{k(k-1)} \\ &= \frac{c_{agg}(k-1)}{k(k-1)} \\ &= \frac{c_{agg}}{k} \end{aligned} \tag{1}$$

We can see that `Bank X` pays c_{agg}/k and the system remains fair for each institution also after `Bank X` joins in.

The second scenario from the list requires `Bank X` to update the core KYC process without repeating the part that was already executed. This update incurs an additional cost c_{upd} for `Bank X`. For brevity, let us assume there are already $k-1$ institutions operating with the customer that have so far equally shared the price, by having paid

$c_{agg}/(k-1)$. In order to on-board the customer, `Bank X` first pays c_{agg}/k which is distributed equally between the $k-1$ institutions. After it executes the update of the core KYC, it uploads hash of the updated document package on the distributed ledger. Putting the hash on the ledger does not represent any cost, but executing the process itself incurred cost c_{upd} that was entirely covered by `Bank X`. When the hash of the document package gets updated on the ledger, the contract automatically creates a debt of value c_{upd}/k for all $k-1$ institutions that were already operating with the customer, not including `Bank X`. The overall cost for `Bank X` then is:

$$\begin{aligned} \frac{c_{agg}}{k} + c_{upd} - \frac{c_{upd}(k-1)}{k} &= \frac{c_{agg}}{k} + \frac{c_{upd} * k - c_{upd} * k + c_{upd}}{k} \\ &= \frac{c_{agg} + c_{upd}}{k} \end{aligned} \tag{2}$$

All other institutions had paid c_{agg}/k before the update was executed and c_{upd}/k afterwards. This shows that the proportionality condition is met. In order to meet the proportionality condition in the future, it is necessary to update the aggregate cost and the cost of executing a single core KYC for the customer. Written as an expression, $c_{agg} \leftarrow c_{agg} + c_{upd}$ and $c \leftarrow c + c_{upd}$.

The third scenario is mathematically identical to the second one. The difference is that instead of the cost of update, c_{upd}, `Bank X` has to cover the cost of executing the core KYC c. Proof of meeting this condition follows equation , with the update cost c_{upd} replaced by c. One difference is that `Bank X` does not need to update hash of the document package, only compare the result it obtains with the result stored on the distributed ledger. The aggregate cost is increased by c.However, c - the cost of a single core KYC execution - remains the same.

The last scenario requires both an update of the core KYC and re-execution of the process in its previous form. This is equivalent to re-execution of the core KYC from the scratch with respect to the newest legislative framework.

Assume that the customer is currently only operating with one financial institution, `Bank A`, when it approaches `Bank X`. `Bank A` executed the core KYC which incurred price c. The aggregate cost is currently also c. In order to on-board the customer, `Bank X` has to pay $c/2$ that is received by `Bank A`. It then needs to repeat the core KYC up to the updated legislative framework. This covers the cost c of repeating the core KYC up to the previous standards plus the cost of the update c_{upd}. When `Bank X` updates hash of the document package, the contract creates a debt of value equal to the cost incurred by `Bank X` for re-executing the updated core KYC, which is $(c + c_{upd})/2$. `Bank A` owes this debt to `Bank X`. After these transactions take place and `Bank A` pays to `Bank X` the debt, the total cost incurred for `Bank A` is:

$$c + \frac{c + c_{upd}}{2} - \frac{c}{2} = c + \frac{c_{upd}}{2} \tag{3}$$

The incurred cost for `Bank X` is:

$$c + c_{upd} + \frac{c}{2} - \frac{c + c_{upd}}{2} = c + \frac{c_{upd}}{2} \tag{4}$$

We can see the two costs are equivalent and the proportionality condition holds true. The aggregate cost and the cost of executing a single core KYC would need to be updated again accordingly: $c_{agg} \leftarrow c_{agg} + c + c_{upd}$, $c \leftarrow c + c_{upd}$.

To prove this condition in general, let us now assume the customer is already operating with $k-1$ institutions and that each institution has so far equally distributed this cost. When the customer wants to operate with a new institution, `Bank X`, this institution has to pay c_{agg}/k to on-board the customer. It obtains access to the customer's documents and requests additional documents required for the updated part of the core KYC. `Bank X` re-executes the core KYC up to the newest legislative framework which incurs cost $(c + c_{upd})$. It updates hash of the document package for the customer, by which it creates a debt of value $(c + c_{upd})/k$ to each of the $(k-1)$ institutions already operating with the customer. When all institutions pay this debt back, the total cost for `Bank X` is the following:

$$\begin{aligned} \frac{c_{agg}}{k} + c + c_{upd} - \frac{(c + c_{upd})(k-1)}{k} &= \frac{c_{agg} + k(c + c_{upd}) - (c + c_{upd})(k-1)}{k} \\ &= \frac{c_{agg} + c + c_{upd}}{k} \end{aligned} \tag{5}$$

The cost for an institution that was already operating with the customer after `Bank X` joined in is:

$$\begin{aligned} \frac{c_{agg}}{k-1} + \frac{c + c_{upd}}{k} - \frac{c_{agg}}{k(k-1)} &= \frac{k * c_{agg} + (k-1)(c + c_{upd}) - c_{agg}}{k(k-1)} \\ &= \frac{(k-1)(c_{agg} + c + c_{upd})}{k(k-1)} \\ &= \frac{c_{agg} + c + c_{upd}}{k} \end{aligned} \tag{6}$$

We can see the two costs are equal and the proportionality condition is met.

4.5.2 Privacy

The privacy condition has two aspects: privacy of the financial institutions and privacy of the customers. The privacy of financial institutions requires the following: First, each institution on the distributed ledger is anonymous to other institutions operating on the ledger and only known to the regulator. Second, an institution cannot know what customer ID's another institution is operating with.

The first criterion is accomplished by not storing the identity of an institution on the distributed ledger, but in a private database only accessible by the regulator.

The second criterion is fulfilled by requiring that each institution uses a unique blockchain account to operate with each customer. The following situation outlines why the second criterion is important. Let us assume `Bank A` and `Bank B` are two financial institution operating on the blockchain with multiple customers. If `Bank B` could identify the blockchain accounts `Bank A` uses to operate with its customers, there would be a large overlap in the customers `Bank A` and `Bank B` are operating with, it could reveal a pattern that would enable the institutions to identify each other. By ensuring that `Bank B`

cannot identify the blockchain accounts `Bank A` uses for operating with its customers, this scenario is avoided.

The privacy of customers ensures that the customer's identity and personal details and documents the customer submitted for the core KYC process cannot be compromised. In addition, it needs to be ensured that a financial institution cannot identify a customer based on their customer profile. The distributed ledger only stores a hash of the document package of the customer. This is accessible to entities on the blockchain - financial institutions and the regulator. It does not store any documents about the customer, which guarantees there cannot be a leak of the customer's personal information. The customer holds their ID and when they want to start operating with a new institution, they give the institution this ID. Only institution that was given this ID knows the real customer's identity behind a digital profile. Otherwise, an institution can find out how many institutions operate with a customer based on the digital profile, but the identity is not revealed. The regulator is the only party that has access to the real identity of each account on the blockchain. This fulfills the privacy condition.

4.5.3 Irrelevance

The irrelevance condition is about establishing there is no extra incentive for any institution to either execute the core KYC or let another institution do so. When a customer approaches the first financial institution, this institution needs to execute the core KYC unconditionally to comply with the regulatory framework. When the customer approaches another financial institution, there is no certainty on whether the institution has to re-execute the process or not. There is only an existing probability that the institution has to repeat the process, but the outcome depends on a random number generator. A financial institution cannot be certain about a single customer, but when operating with multiple customers, it can estimate the number of core KYC processes it needs to execute according to the law of large numbers[8]. Similarly, a financial institution has no impact on whether another institution needs to do the core KYC process for a customer. If a financial institution refuses to do the process when required, it is unable to operate with the customer. This fulfills the irrelevance condition.

4.5.4 No-minting

The no-minting ensures that a financial institution cannot simulate having executed the core KYC verification process without actually doing so. This condition is partially fulfilled by requiring an institution to pay appropriate fee before it can on-board a customer. Any time an institution has to update, repeat the core KYC, or do both, there is a digital footprint left on the distributed ledger signifying this occurred. This can only occur when there was a previous request for this from the smart contract. Otherwise, the smart contract does not allow an institution to simulate it executed or updated the core KYC.

The challenge is to assure that when the core KYC has to be repeated, updated, or both, this is adequately executed. Controlling the adequacy of the process can not be

[8] `https://en.wikipedia.org/wiki/Law_of_large_numbers`

directly done using the distributed ledger, because the ledger only stores the hash of the customer's document package. This hash does not reveal any information about the process.

This condition is fulfilled implicitly by an incentive mechanism. A financial institution has a clear incentive to execute the core KYC adequately in compliance with the regulatory framework. It could otherwise later face serious legal and reputational costs. After all, this is the mechanism that is used in the current, non-DLT KYC process.

We propose that the regulator could do controls of the institutions that had to execute or update the core KYC. These controls can become more efficient due to the lower number of cases where they would be required. The number of cases would decrease because the DLT-KYC scheme does not require the core KYC to be executed by each institution a customer operates with, but allows for mutual cooperation and sharing of the result of the process between the institutions.

4.6 Implementation details

We propose a single smart contract, with full code publicly available[9], that implements the DLT-based core KYC process. The contract needs to be deployed only once, by the regulator, and can be used by numerous financial institutions and serve for multiple customers. In order to operate with the deployed contract, an institution needs an account on the ledger. It needs to be initially verified by the regulator to obtain a permission to create an account on the blockchain. After it is verified, the institution would be able to create multiple accounts as it is expected to use a unique account for each of its customers.

The following list summarizes the main fields included within the customer's profile:

- `document_package_hash` - hash of the customer's document package
- `require_update` - identifies that an update in the core KYC is required
- `update_in_progress` - specifies whether an institution is currently working on an update of the core KYC for a customer
- `single_kyc_price` - price of executing a single core KYC process for the customer with respect to the current legislative framework
- `cumulative_kyc_cost` - the cumulative cost incurred by the core KYC process for the customer
- `repeat_probability` - probability with which the core KYC has to be repeated
- `institution_count` - the count of institutions the customer is operating with
- `kyc_count` - the count of the core KYC processes that have been executed for the customer
- `rating_average` - average rating as assigned by institutions operating with the customer

[9] `https://github.com/Matus23/KYC-Ethereum-smart-contracts`

Our smart contract allows the core KYC process to be updated - an important aspect the basic DLT-KYC did not provide. When a change that requires this update is introduced in the legislative framework, the regulator can simply set the flag `require_update` to true for every customer it applies to. The advantage is that this process is automated and executed directly by the regulator that introduces the change in the legislative framework. Each institution is alerted and knows that an action needs to be taken within a certain time period.

When the core KYC for a customer needs to be updated, two or more financial institutions might want to simultaneously execute this. This might not always be required and would hinder the proportionality condition. Hence, we introduce an additional field, `update_in_progress` that can be set by any institution operating with the customer to incorporate this update. It indicates that an institution is working on this update for the customer and can be only set to true when the `require_update` is true.

The combination of these two fields ensures that all institutions are notified when an update in the core KYC is required and that only one institution executes this process at a time. Of course, if an institution would like to do this update outside the ledger, it can always do so. When the update for a customer is executed, both flags are set to false.

The `repeat_probability` specifies the probability with which the core KYC has to be repeated. A higher value increases security of the system, but reduces the saved costs introduced by using the DLT. A lower value would be recommended to use for customers that possess a low level of risk, while a higher probability can be used for customers with a relatively high level of risk. Currently, it is up to the first institution that operates with the customer to set this probability. This could be later changed based on a proposal that would be reached in cooperation with financial professionals.

A customer with a high level of risk, such as a politically exposed person, would almost certainly require additional controls outside the DLT-based core solution. However, the details of the core KYC would still be dependent on the individual customer and the process would not incur equal cost for all customers. This cost is therefore individual for each customer and is currently specified by the first institution executing it. Note that in order for this solution to be put in practice, a more sophisticated approach would have to be taken, otherwise the institution could artificially increase the cost so that it would receive a higher financial compensation in the future. On the other hand, when the customer is not operating with any other institution yet, it is more difficult to identify the relevant parties that should determine the cost of the core KYC.

The `cumulative_kyc_cost` specifies the overall cost incurred by possibly multiple core KYC processes and updates for the customer.

The `institution_count` field specifies how many financial institutions a customer is operating with. `kyc_count` is a similar field specifying how many of these institutions had to execute the core KYC process for the customer. The higher the `kyc_count`, the more secure it is to operate with this customer and the institution might decide to execute less additional checks, further reducing the compliance costs.

5 Security Analysis: DLT-KYC Robustness

5.1 Mathematical background

This section provides mathematical analysis of the security of our improved DLT-KYC system where the core KYC has to be repeated with a certain probability. We give probabilistic models to answer the following question:

> What is the probability that the whole system, for a given customer, is working correctly with respect to the regulatory framework? In other words, what is the probability that the system, for a given customer, implies no risk to any of the financial institutions?

We need to introduce a few variables we will be operating with. Let us call the average probability that a financial institution executes the core KYC adequately with respect to a regulatory framework p_{ok}. The probability that a future financial institution will have to repeat the core KYC process is p_{rep}. Finally, let $list\_size$, also abbreviated as LS, be the number of institutions operating with a customer.

This model is based on the following two assumptions.

1. For a given customer, every financial institution that wants to operate with the customer and has to execute the core KYC process would do so independently of the other financial institutions.
2. All financial institutions would adequately execute the core KYC process, with respect to a regulatory framework, with the same probability. We call this probability p_{ok}. While this scenario is quite unlikely, p_{ok} can be chosen as the average probability of all institutions executing the process adequately.

After the core KYC for a customer is executed by the first financial institution, we know that each financial institution that would like to start operating with the customer in the future will either have to repeat the core KYC process with probability p_{rep}, or just enter the list without repeating it with probability $1 - p_{rep}$. Given that the $list\_size$ has size $n+1$, we know that the probability that the core KYC process was repeated a certain number of times behaves according to the Bernoulli distribution[10]. We introduce R as a random variable expressing how many times the process was repeated. LS is a random variable expressing the number of institutions operating with the customer, also called the size of the institution list of the customer.

The relation expressing probability of how many times the process was repeated as a function of the size of the customer's institution list is described by the following equation:

$$P(R = k \mid LS = n+1) = p_{rep}^{k}(1 - p_{rep})^{n-k}\binom{n}{k} \tag{7}$$

The reasoning is the following: the first financial institution has to execute the core KYC. Hence, there are n additional financial institutions that need to re-execute the core KYC with probability p_{rep}. The first term of the product, p_{rep}^{k}, expresses the

[10] https://en.wikipedia.org/wiki/Bernoulli_distribution

probability that the process was repeated k times. This would mean that the remaining $n-k$ times, the process was not repeated. This is expressed by the second term of the product $(1-p_{rep})^{n-k}$. The extra term $\binom{n}{k}$ expresses the number of possible combinations when k core KYC processes would be repeated out of n trials.

Let us introduce a new random variable, OK, that expresses how many times the core KYC was adequately repeated with respect to the regulatory framework. We need to consider this as there is always a small possibility that a financial institution either intentionally, or unintentionally, does not do the customer verification in a way that would comply with the regulatory framework. The probability that the core KYC process was executed adequately i number of times, given that it was repeated k number of times, is given by the following equation:

$$P(OK = i \mid R = k) = p_{ok}^{i}(1-p_{ok})^{k-i}\binom{k}{i} \tag{8}$$

In order to find out if the first institution executed the core KYC adequately, there needs to be at least one financial institution repeating the core KYC that does so adequately. This institution would either obtain the same result, or it would spot the difference between the two processes. Given that there are k institutions repeating the core KYC, this probability can be written as:

$$\begin{aligned} P(OK >= 1 \mid R = k) &= 1 - P(OK = 0 \mid R = k) \\ &= 1 - (1-p_{ok})^{k} \end{aligned} \tag{9}$$

We need to obtain the probability that the core KYC was adequately repeated i times as a function of the size of the institution list (LS), not as just as a function of the number of core KYC repetitions (R). We need this expression because random variable R depends on p_{rep}, which is a hyper-paremeter that would be set by the institutions. On the other hand, the number of institutions operating with a customer is only dependent on the customer's incentive to operate with other institutions and cannot be tuned by the system.

The probability that the core KYC was adequately repeated i times out of k repeated attempts, where $i <= k$, when the customer operate with $n+1$ institutions is:

$$\begin{aligned} P(OK = i \mid LS = n+1) &= \sum_{k=0}^{n} P(OK = i, R = k \mid LS = n+1) \\ &= \sum_{k=0}^{n} P(OK = i \mid R = k, LS = n+1) P(R = k \mid LS = n+1) \\ &= \sum_{k=0}^{n} P(OK = i \mid R = k) P(R = k \mid LS = n+1) \\ &= \sum_{k=0}^{n} p_{ok}^{i}(1-p_{ok})^{k-i}\binom{k}{i} p_{rep}^{k}(1-p_{rep})^{n-k}\binom{n}{k} \end{aligned} \tag{10}$$

The above equation used the sum and product rule in probability theory [29]. An important observation is that $P(OK = i \mid R = k, list\_size = n+1) = P(OK = i \mid R =$

k), as when we know how many times the core KYC was repeated, the size of the institution list is no longer relevant.

We can simplify the above relation, as we need to find out the probability of repeating the core KYC successfully at least once, instead of exactly i times. Putting equation 9 and 10 together, we obtain:

$$P(OK >= 1 \mid LS = n+1) = \sum_{k=1}^{n} [1-(1-p_{ok})^k] p_{rep}^k (1-p_{rep})^{n-k} \binom{n}{k} \tag{11}$$

Note that the sum here goes from $k = 1$ instead of $k = 0$, as was the case in equation 10. In order to have at least one adequately repeated core KYC process, the process itself has to be repeated at least once in the first place. Thus, for the case $k = 0$, the probability $P(OK >= 1 \mid R = 0) = 0$ by definition.

In order to answer question outlined in the beginning of this section, we need to find a formula describing how secure the scheme is overall. If the first institution executed the core KYC inadequately, our scheme is only secure if there is at least one of the following institutions that adequately re-executed the core KYC. If the first institution executed the core KYC adequately, then the following institutions that repeated the process could make a mistake in the process and the system would still be secure. It would be secure as it would continue to rely on the the first execution of the process and the consequent ones would be recognised as incorrect. Using equation 11, the probability that the system is safe (s), given than the customer operates with $n+1$ institutions, is mathematically written as:

$$\begin{aligned} P(s \mid LS = n+1) &= P(OK \geq 0 \mid LS = n+1) p_{ok} + P(OK \geq 1 \mid LS = n+1)(1-p_{ok}) \\ &= p_{ok} + (1-p_{ok}) \sum_{k=1}^{n} [1-(1-p_{ok})^k] p_{rep}^k (1-p_{rep})^{n-k} \binom{n}{k} \end{aligned} \tag{12}$$

This holds true as $P(OK >= 0 \mid list\_size = n+1) = 1$ by definition.

5.2 Results of the analysis

We present some model situations coming from our mathematical analysis to demonstrate that our system is more cost-efficient and secure than the KYC scheme that is currently put in practice. The security is also enhanced in comparison to the basic DLT-KYC scheme, but there is a compromise between the cost-reduction and enhanced security our solution offers offers.

The first diagram simulates a situation in which a financial institution is assumed to adequately execute the core KYC process in 90% of the cases and inadequately in the remaining 10%. By inadequately, we mean that the institution could either intentionally or unintentionally underestimate a certain aspect of the core KYC process and start operating with the customer without proper verification as required by the legislative framework. The core KYC process has to be repeated with a 50% probability. This applies to each financial institution operating with the customer, except for the first

institution that has to execute the core KYC unconditionally. The y axis on the left shows the probability that the core KYC for a given customer was executed adequately and the system is secure. This is achieved when at least one of the institutions has executed the core KYC for the customer adequately. The y axis on the right shows the price of the core KYC for a single financial institution as a multiple of an average cost of executing the core KYC for the customer.

Note that the average probability of executing the core KYC adequately in 90% is only an estimate and is most likely well below the real value. However, this value well illustrates how the security of the system increases.

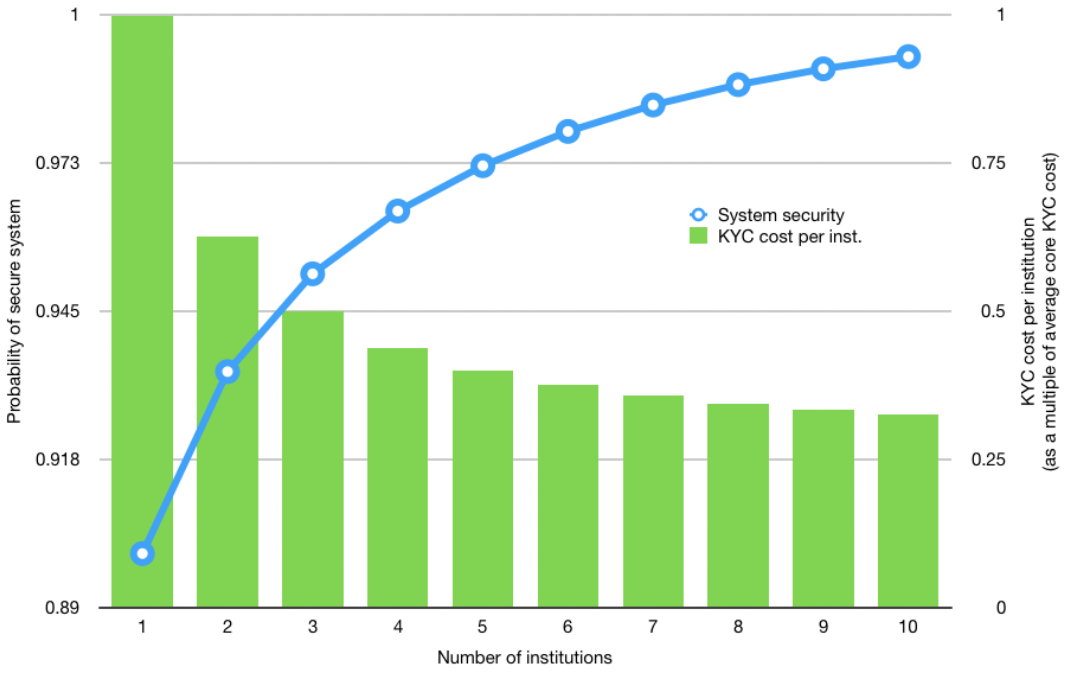

Figure 5: Security of the system for a customer. An institution executed on average core KYC adequately in 90% cases. The Core KYC needs to be repeated with 50% probability.

We can see that as the number of institutions a customer is operating with increases, the average cost of executing the core KYC for an institution decreases and the security of the system increases. This is really desirable, as it proves that our system is both more financially efficient and more secure than the current scheme. The current scheme, where the core KYC is executed by each financial institution individually, is independent of the number of institutions a customer is operating with. This means the diagram would show the same results as figure 5 shows for the case where the number of institutions is 1. This means that the security of the system would remain 0.90 and the cost per institution would be the average cost of executing the core KYC.

The probability of repeating the core KYC process is a hyper-parameter that would be selected based on opinions of professionals representing the financial institutions.

It could be different for each customer and be dependent on how much risk the customer represents. The second diagram illustrates a situation where the customer is operating with 5 financial institutions. As before, a financial institution executes the core KYC process adequately in 90% cases. The x axis now represents the different probabilities of repeating the core KYC process. The y axis on the left and on the right remain unchanged and represent the security of the system and the cost associated with the core KYC per institution respectively.

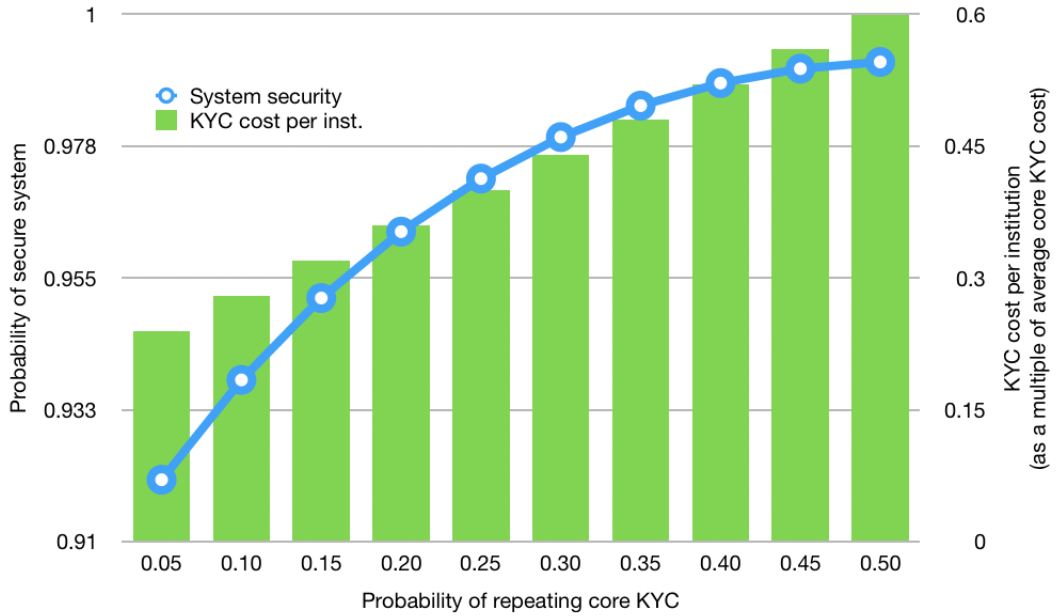

Figure 6: Security of the system for a customer. Institution on average executed core KYC adequately in 90% cases. Customer operates with 5 institutions.

We can see that the security of the system as well as the cost incurred for the core KYC per institution increases as the probability of repeating the core KYC increases. This is due to the fixed number of institutions the customer is operating with. Figure 6 clearly shows the compromise between increased security of the system and decreased cost reduction. Note that the cost per financial institution is in all cases lower and the security of the system is higher than if each institution were to execute the core KYC individually. If each institution executed the core KYC individually, the cost would be 1, which means it would be the average cost of executing the process. The security would be 0.9.

However, the extent to which the cost can be reduced depends on the probability of repeating the core KYC that has a negative effect on the security of the system. A compromise between the two needs to be made. For instance, for a customer with a higher level of risk, the core KYC could be repeated with probability 50%, where the cost per financial institution would be 0.6 of the average core KYC cost, but the security of the system would be 99.18%. If a customer represents a lower level of risk, the probability of repeating the core KYC could be set to a lower value, such as 10%. The cost per institution would be 0.26 of the average cost associated with the core KYC

and the security of the system would be 93.76%.

The final figure compares the security of the system for various probabilities of repeating the core KYC process. We observe this as a function of the number of institutions the customer is operating with. The x axis thus shows the number of institutions and the y axis the security of the system.

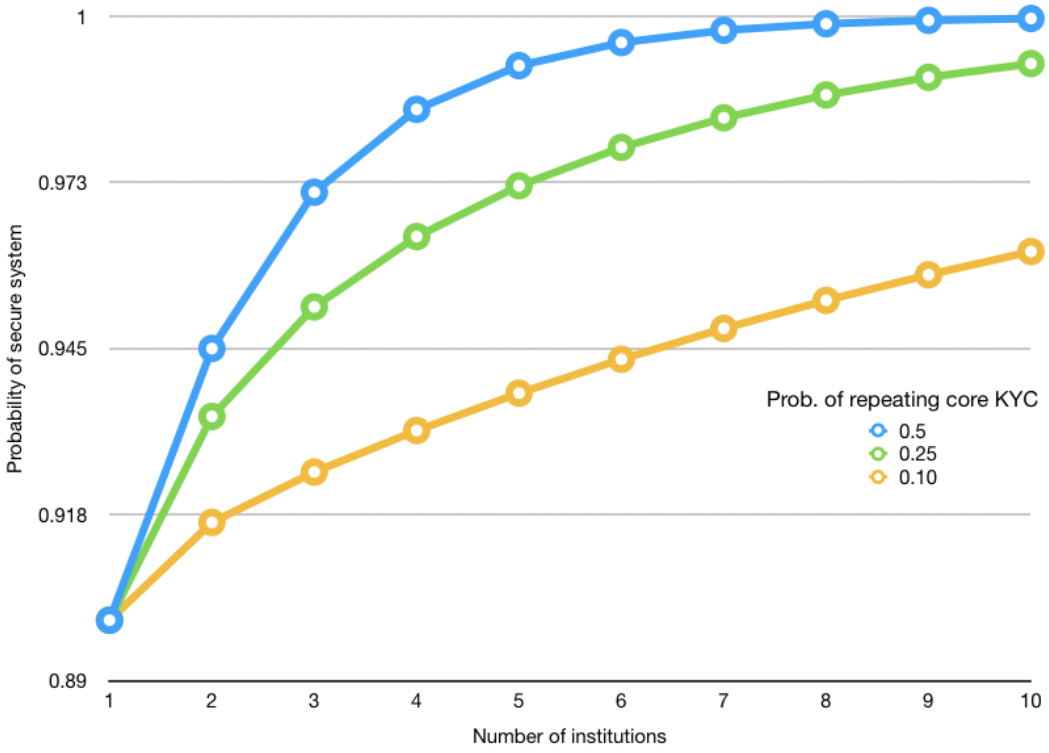

Figure 7: Security of the system for a customer. An institution executed on average core KYC adequately in 90% cases.

6 One-Tier problem

In section 4.2, we mentioned a key shortcoming of the Basic DLT-KYC system - the non-repeating nature of the core KYC process - and showed that it can be effectively tackled by our Robust DLT-KYC that introduced a probability of repeating the core KYC.
Another pitfall that, to the best of our knowledge, has not been addressed in any DLT-based KYC solution is that each financial institution is assumed to have equal capability of executing the core KYC process. The core KYC is a unified process specified by the regulator and each institution that would like to use our system needs to be verified by the central authority. However, this is not enough evidence for this assumption to hold true. We operate with this assumption in our mathematical analysis in 5.1 where we work with the average case of institutions' capability of executing the core KYC adequately. Although our analysis provides a good model on average, this might not suffice for a real implementation. We call this a *one-tier problem*, as it assumes that each institution is equally good at executing the core KYC process and all institutions belong to a single tier.

Some financial institutions might conduct business with more clients, be in charge of a larger capital, and have a long history on the market and strong reputation to uphold. Let us call such an institution `Bank X`. Imagine `Bank X` would like to start operating with a new customer using our proposed Robust DLT-KYC system. Assume that the customer was already operating with a financial institution, `Bank A`, that executed the core KYC and was the only institution to do so. It can happen that before `Bank X` starts operating with this customer, the institution neither needs to repeat, nor needs to update the core KYC process - it only has to pay a required fee to onboard the customer. In such a case, `Bank X` solely relies on the core KYC process as executed by `Bank A`. The problem is that the financial institutions are anonymous to each other - `Bank X` does not know anything about `Bank A`, except for the fact that it passed regulator controls before it joined the blockchain. However, these initial controls performed by the regulator may not provide reliable grounds for assuring that an institution will be infallible in executing KYC controls in the future. Potential financial or reputational consequences coming from operating with a possibly malicious customer may outweigh the benefits of our solution. This situation naturally presents a strong disincentive for `Bank X` to use a DLT-based KYC system.
In fact, our Robust DLT-KYC system partially solves this by introducing the probability of repeating the core KYC by a random institution. If a customer operates with many financial institutions, there is a good chance that institutions of different tiers (with different capability of executing the core KYC) executed the core KYC. This provides good operational grounds for our mathematical analysis of the Robust DLT-KYC model. The shortcoming is that in reality, many customers may not operate with sufficiently many financial institutions and we need a different way to work around the one-tier problem.

We introduce a financial institutions rating system. The rating represents an institution's quality of adequately executing the core KYC process with respect to a regula-

tory framework. Our last smart contract gives an implementation for this. When `Bank X` wants to now operate with a new customer, it can view the list of institutions that executed or updated the core KYC process for this customer, together with the ratings of these institutions. Based on this information, `Bank X` knows the level of trust it can have in the work of these institutions. If the institutions have a high rating, `Bank X` knows that this result is more reliable and can decide to execute less additional customer checks outside the core KYC process. On the contrary, if `Bank X` sees that the core KYC was executed only by a single FI, and this institution has a low rating, it can decide to repeat the core KYC process or execute additional controls before it starts operating with this customer. An example is illustrated in figure 8.

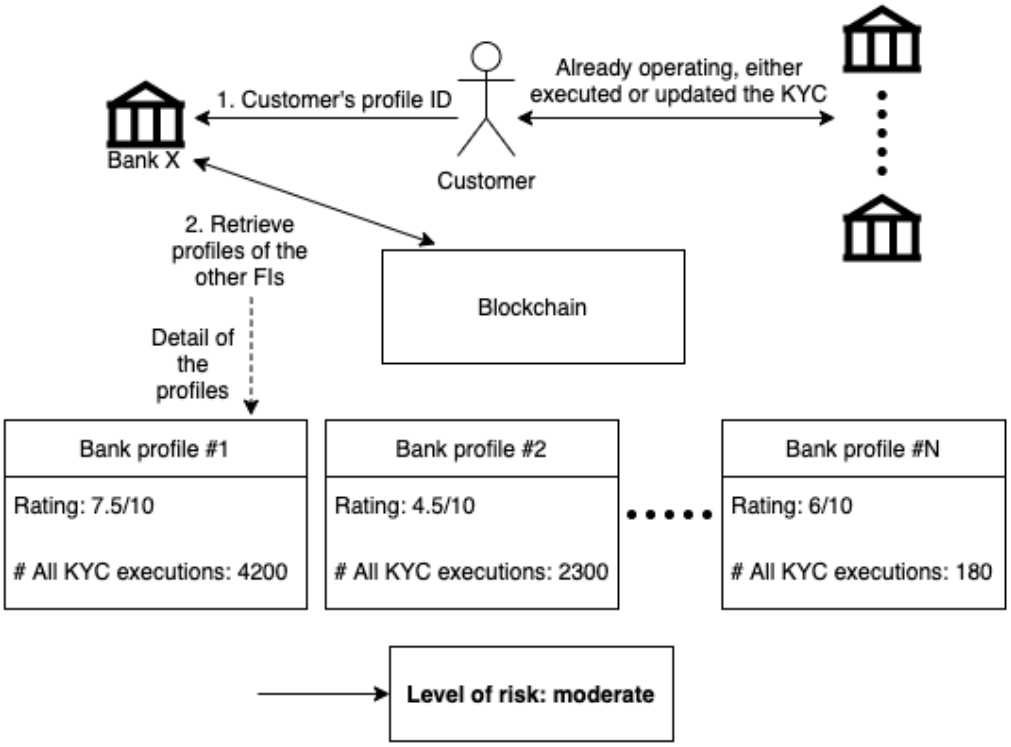

Figure 8: Customer approaches a new financial institution `Bank X`. `Bank X` can view which institutions executed or updated the core KYC for this customer and ratings of these institutions. Based on this information, it can forecast a level of trust it puts into the work of these institutions and decide on additional customer checks outside the blockchain solution.

The rating of a financial institution is the average of ratings that were assigned to this institution by other institutions operating on the blockchain. A financial institution (`Bank A`) can rate another financial institution (`Bank B`) under the following conditions:

- Bank A and Bank B must be operating with at least one mutual customer. Let us call the set of customers that operate with both institutions *S*. If the set S is an empty set, the rating cannot be assigned. The rating presents how satisfied

Bank A is with Bank B's execution of the core KYC for its customers - when S is empty, Bank A cannot assess this ability of Bank B.

- Bank B must have executed or updated the core KYC for at least a single member of set S. Set S represents customers that operate with both institutions, but does not specify whether Bank B executed or updated the core KYC process for any members of this set. Hence, it is required that for at least one of the members of S, Bank B executed or updated the core KYC.

- Bank A can only give Bank B a single rating, no matter what is the size of S. Bank A always has a possibility to change this rating if something unexpected happens. For example, a customer may turn out to be malicious after a longer period of time in which case it may be revealed only later that Bank B was fallible in executing or updating the KYC process.

In order to fulfill these conditions, we need to create a profile for each financial institution on the blockchain. This profile does not reveal real identity of an institution, a key concept which would breach our privacy condition outlined in section 2, but it provides a list of customer profiles the institution operates with. This list has to be provided in order to ensure appropriate checks the smart contract includes to avoid its misuse. The list is stored on the blockchain and is accessible to parties operating on the blockchain - the regulator and all financial institutions.

The remaining security challenge is the following: imagine a new FI, that does not yet operate with any customers, enters the blockchain. This institution is able to view the mappings between other FIs and their customers, but cannot deduce any further information based on this. However, any institution knows a customer's real identity behind their customer profile when it starts operating with them - an institution needs to know real identities of customers it operates with. Eventually, there will be two or more institutions operating with multiple mutual customers. These institutions will be able to use information retrieved from the blockchain, and thus our solution, to know of each other. This will not be sufficient to identify each other, but it must be considered that each institution, after some time of conducting business with a customer, would have additional information about its customers. The combination of internal information (e.g. the type of customer's transactions, investment portfolio etc.), information retrieved from the blockchain (identity of customers that mutually operate with this and other FIs), and external information about other FIs gained from being in the market (e.g. Bank X focuses on middle net-worth clients with mainly retail investment portfolio), could be sufficient for institutions to identify each other. This is illustred in figure 9.

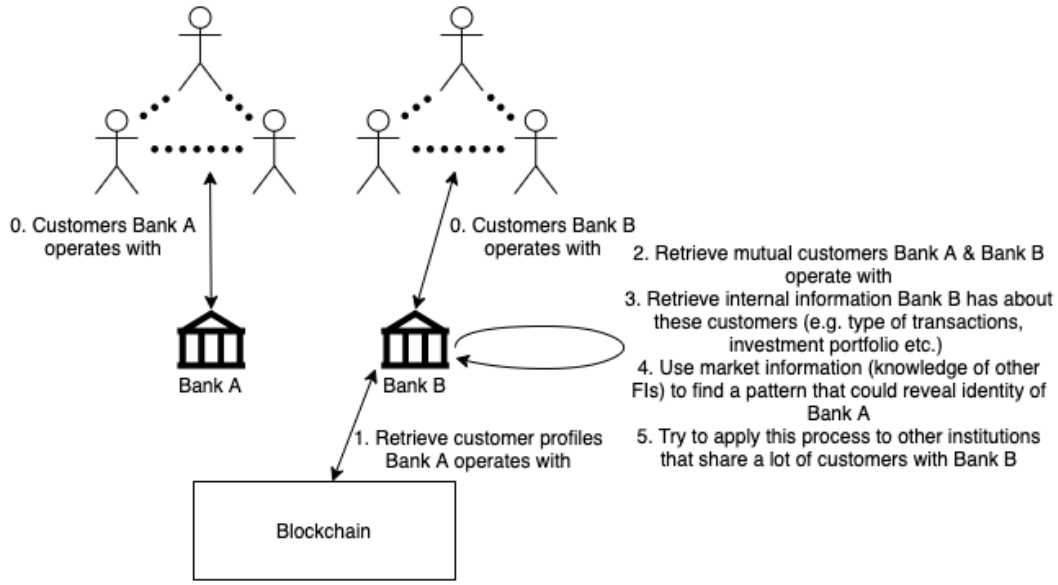

Figure 9: This figure illustrates how Bank B may try to unravel the real identity of Bank A based on its blockchain profile. It can try to unravel the identity of any other institution on the blockchain - the more customers, and the more unique the customers are, the higher the probability of success. Bank A, and any other institution on the blockchain, can try to unravel the identity of Bank B or another institution.

Whether this scenario would be real or just hypothetical has to be consulted with professionals from financial sector. Due to the potential of this risk, this solution is not included with our Robust DLT-KYC, but is outlined in this separate section and given its own smart-contract implementation.

7 Future Work

Several studies [37, 44, 45] outlined that using DLT could bring additional benefits in reporting and monitoring by tracking customers' transactions. Recording all customers' transactions on the distributed ledger is currently out of scope of this paper. Our DLT-KYC system stores transactions a financial institution needs to execute to start operating with a customer and to keep the system fair. It does not intend to store transactions between a customer and a financial institution, as these require an enhanced level of privacy and hence, if stored on the blockchain, they have to be suitably protected. Further research and work could be done to investigate an integrated system that would offer onchain transaction tracking and tracing.

We outlined a user profile that can be used to estimate a risk level the customer presents and does not require the customers' transactions to be stored on the distributed ledger. A financial institution that would like to start operating with the customer can use this risk level to determine the extent of additional checks it would execute. The institution might decide to lessen these controls when the customer presents a low level of risk, and dwell on these controls when the customer presents a high level of risk. However,

we did not specify how this level of risk would be calculated.
One simple way to offer this is to set the risk level as a numerical value on a scale of 1 to 10. A value of 10 would represent the maximal possible risk and the value 1 the lowest risk. This value would be dependent on the following factors: average rating of the customer, the probability of repeating the core KYC for the customer, cost of executing a single core KYC for the customer, and the number of times the core KYC was executed for the customer. The risk should increase with an increase in single core KYC cost and the probability of repeating this process. It should decrease with an increase of the average customer rating and the number of times the core KYC was executed. A precise mathematical expression would have to be formulated in cooperation with financial institutions that would be able to identify relevance of each factor and this topic remains open for future work.

In section 6, we used a bank rating scheme to deal with the false assumption of each institution having equal capability of executing the core KYC process. Our model offers a simple rating scheme where a FI's rating is the average of ratings it was assigned by other FIs. Additional factors, such as the real institution's reputation, number of customers and capital it is operating with, could contribute to this rating. Designing a more sophisticated rating scheme could be done in cooperation with professionals from a financial sector. However, one desirable improvement would be to keep the mapping between a financial institution profile and profiles of customers the institution operates with not visible to all parties on the blockchain. This could be either achieved by storing this off the chain, in a database maintained by the central authority, or by using a blockchain that would not reveal this information. A good potential candidate is Corda[11] and future work can be to re-implement our contracts using this blockchain.

Finally, our future work can be dedicated to designing and implementing a web application that would simulate how financial institutions and the regulator would operate with our solution. The web application needs to provide a graphical interface for seamless interaction with the smart contract deployed on the blockchain. Professionals responsible for the KYC process need to be able to easily pay onboarding fees, retrieve customer information and data, rate customers and other financial institutions, and retrieve some operational statistics. A non-exhaustive list includes the total cost incurred by using this solution in contrast to the current KYC scheme, change in customer satisfaction etc. FIs would get notifications when a change in the legislative framework would be made by the regulator and could monitor the potential risk of their customers. If a customer's rating would suddenly drop, for instance due to using their account at some other financial institution for illicit activities, all FIs the customer operates with would be swiftly notified of this. On the other hand, the central authority needs to have a clear overview of transactions executed on the blockchain, maintain a database identifying each entity on the blockchain and notify about changes in the legislative framework and what type of customers these changes would apply to.
Implementing a web application for the second part of my MInf report is a viable option as it is very scalable - the complexity would depend on the objectives we would

[11] https://www.r3.com/corda-platform/

specify. These could be either specified by us, based on the needs outlined in the previous paragraph and inspired by future research, or could be consulted with a professional in financial sector if there was a possibility for it.

8 Conclusion

The KYC process is one of emerging use-cases in the financial services setting that is being addressed via the use of DLT. When put to use, our solution can reduce compliance costs and increase security of the system at the same time. Naturally, there is a trade-off between reduction in the compliance costs and improvement in security. Our solution would benefit all parties - regulators could accomplish more efficient due diligence controls and customers' satisfaction with the service would increase due to shorter average waiting time for this service. These benefits are brought by the distributed nature of this technology and the way our system can leverage this. Institutions can share the result of their work and pay an appropriate fee instead of repeating the entire process, saving time and monetary funds. The customer profile in the distributed system enables the regulator to easily identify customers that possess a higher level of risk which can make future regulatory controls of the customers, that currently face a high false positive rate, more efficient.

Due to the complexity of this topic, we identified several research objectives and answered them in a chronological order. We started off with a legal background, putting our research in a broader perspective. We described the current KYC process and contrasted it with the benefits DLT can bring in this field. Building on top of reviewed literature, we provided a smart-contract implementation of work that was only researched on a conceptual level - the Basic DLT-KYC, as well as an implementation of our improved Robust DLT-KYC system. We gave a mathematical model to quantify benefits unique to our model and described its impact on a few illustrative examples. Finally, we identified the one-tier problem and outlined how it can be mitigated, providing a smart-contract implementation with the opportunity cost of decreased privacy it introduced.

There are several alternatives for our future exploration in this field. These include: (1) providing a deeper insight into fundamental aspects of the distributed ledger technology, how it can be applied for this use-case and what type of blockchain would be most suitable, (2) identifying additional conditions of the KYC process and how to effectively fulfill these, (3) what additional benefits can be brought by a DLT-based KYC solution, and (4) building an web application that would illustrate how the regulators and financial institutions would operate with our proposed smart-contract solution and could give further answers for the previous points.

The emergence of financial technology (FinTech) and regulatory technology (RegTech) companies brings a strong competition on the financial services market. Traditional financial institutions attempt to quickly adapt to new technological changes in order to provide the best services for their customers. The business impact the distributed ledger technology can deliver in this sector is significant and investing in its research

has been one of the objectives of financial firms for several years [27, 14]. A DLT-based KYC system is an exemplary use-case that has potential to significantly benefit the financial sector, targeting financial firms, their customers and the regulator, but requires a lot of future work and cooperation between the world of academia and industry. Our work builds on top of existing foundations in this field and uniquely presents, gives implementations for, and solutions to the *brittleness* and *one-tier* problems of the previous existing DLT-based KYC solutions. For the second part of this project, we intend to delve deeper into this topic and focus on researching the most interesting aspects of this field.

Bibliography

[1] Ethereum sharding, Accessed in March 2020.

[2] Maher Alharby and Aad Van Moorsel. Blockchain-based smart contracts: A systematic mapping study. *arXiv preprint arXiv:1710.06372*, 2017.

[3] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, pages 1–15, 2018.

[4] Paul Arssov. What is a dapp? and what it most definitely is not., September 2019.

[5] Hong Kong Monetary Authority. Whitepaper 2.0 on distributed ledger technology. 2017.

[6] David Barkai. An introduction to peer-to-peer computing. *Intel Developer update magazine*, pages 1–7, 2000.

[7] Imran Bashir. *Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained*. Packt Publishing Ltd, 2018.

[8] Roman Beck, Michel Avital, Matti Rossi, and Jason Bennett Thatcher. Blockchain technology in business and information systems research, 2017.

[9] Oliver Belin. The difference between blockchain and distributed ledger technology.

[10] Michael J Bordash, Michael J Hudson, and Chih-Hong Wong. Tracking transactions through a blockchain, February 22 2018. US Patent App. 15/239,639.

[11] Matthew Britton. Could blockchain solve the kyc/aml challenge?, September 2016.

[12] Richard Gendal Brown. The corda platform: An introduction. *Retrieved*, 27:2018, 2018.

[13] V Buterin. A next generation smart contract & decentralized application platform (2013) whitepaper. *Ethereum Foundation*.

[14] Michael Casey, Jonah Crane, Gary Gensler, Simon Johnson, and Neha Narula. *The impact of blockchain technology on finance: a catalyst for change*. ICMB, International Center for Monetary and Banking Studies, 2018.

[15] Colin Powell Charles Freeland. Customer due diligence for banks. 2001.

[16] JPMorgan Chase. Quorum, Accessed in March 2020.

[17] Christopher D Clack, Vikram A Bakshi, and Lee Braine. Smart contract templates: foundations, design landscape and research directions. *arXiv preprint arXiv:1608.00771*, 2016.

[18] Ethereum. Private network, Accessed in March 2020.

[19] Hyperledger Fabric. Architecture origins, Accessed on March 2020.

[20] FATF. Fatf 40 recommendations, 2004.

[21] FCA. Fca fines standard chartered bank £102.2 million for poor aml controls, 2019.

[22] FCEN. History of anti-money laundering laws, Accessed in March 2020.

[23] FDIC. Bank secrecy act, anti-money laundering, and office of foreign assets control, 1970.

[24] Florian Glaser. Pervasive decentralisation of digital infrastructures: a framework for blockchain enabled system and use case analysis. 2017.

[25] Pedro Goncalves. Money laundering fines total $8.14bn in 2019, 2020.

[26] Dominique Guegan. Public blockchain versus private blockhain. 2017.

[27] Ye Guo and Chen Liang. Blockchain application and outlook in the banking industry. *Financial Innovation*, 2(1):24, 2016.

[28] Omkar S. Hiremath. Ethereum private network – create your own ethereum blockchain!, May 2019.

[29] Arno Onken Iain Murray. Bayesian inference and prediction, 2019.

[30] Brian Lee Steven Kent Ingrid Groer Eric Beardsley James Schneider, Alexander Blostein. Profiles in innovation: Blockchain: Putting theory into practice, 2016.

[31] Shani Koren. Know your customer, 2018.

[32] Roy Lai and David LEE Kuo Chuen. Blockchain–from public to private. In *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 2*, pages 145–177. Elsevier, 2018.

[33] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226. 2019.

[34] Jenny Lin, Perry Haldenby, John Jong Suk Lee, Paul Mon-wah Chan, and Orin Del Vecchio. Systems and method for tracking enterprise events using hybrid public-private blockchain ledgers, September 3 2019. US Patent 10,402,792.

[35] YVONNE Lootsma. From fintech to regtech: The possible use of blockchain for kyc. *Fintech To Regtech Using block chain*, 2017.

[36] Roger Maull, Phil Godsiff, Catherine Mulligan, Alan Brown, and Beth Kewell. Distributed ledger technology: Applications and implications. *Strategic Change*, 26(5):481–489, 2017.

[37] Matthias Memminger, Mike Baxter, and Edmund Lin. Banking regtechs to the rescue. *URL www. bain. com/Images/BAIN_BRIEF_Banking_Regtechs_to_the_Rescue. pdf*, 2016.

[38] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies. *Curso elaborado pela*, 2015.

[39] Haroon Shakirat Oluwatosin. Client-server model. *IOSRJ Comput. Eng*, 16(1):2278–8727, 2014.

[40] Vimalkumar Pachaiyappan and R Kasturi. Block chain technology (dlt technique) for kyc in fintech domain: A survey. *International Journal of Pure and Applied Mathematics*, 119(10):2108, 2018.

[41] Matt Packer. Five minutes on... the surge in anti-money laundering fines, 2019.

[42] José Parra-Moyano and Omri Ross. Kyc optimization using distributed ledger technology. *Business & Information Systems Engineering*, 59(6):411–423, 2017.

[43] José Parra-Moyano, Tryggvi Thoroddsen, and Omri Ross. Optimized and dynamic kyc system based on blockchain technology. *Available at SSRN 3248913*, 2018.

[44] Neepa Patel. Blockchain kyc/aml utilities for international payments, 2017.

[45] Refinitiv. A blockchain enabled kyc solution: New horizon or falsedawn?, 2018.

[46] Thompson Reuters. Know your customer survey, 2016.

[47] Thompson Reuters. Know your customer survey, 2017.

[48] Sara Rouhani and Ralph Deters. Performance analysis of ethereum transactions in private blockchain. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 70–74. IEEE, 2017.

[49] Rüdiger Schollmeier. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Proceedings First International Conference on Peer-to-Peer Computing*, pages 101–102. IEEE, 2001.

[50] Atul Singh, Tathagata Das, Petros Maniatis, Peter Druschel, and Timothy Roscoe. Bft protocols under fire. In *NSDI*, volume 8, pages 189–204, 2008.

[51] Prince Sinha and Ayush Kaul. Decentralized kyc system. *International Research Journal of Engineering and Technology (IRJET)*, 5(8):1209–1210, 2018.

[52] Josh Stark. Making sense of blockchain smart contracts, Jun 2016.

[53] Corda Team. Corda pluggable consensus, September 2018.

[54] Martin Valenta and Philipp Sandner. Comparison of ethereum, hyperledger fabric and corda. *no. June*, pages 1–8, 2017.

[55] Nedyalko Valkanov et al. Smart compliance or how new technologies change customer identification mechanisms in banking. *Economics and computer science*, (2):12–19, 2019.

[56] M. Walport. Distributed ledger technology: Beyond block-chain, 2016.

[57] Iza Wojciechowska. What is kyc and why does it matter?, 2019.

[58] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.