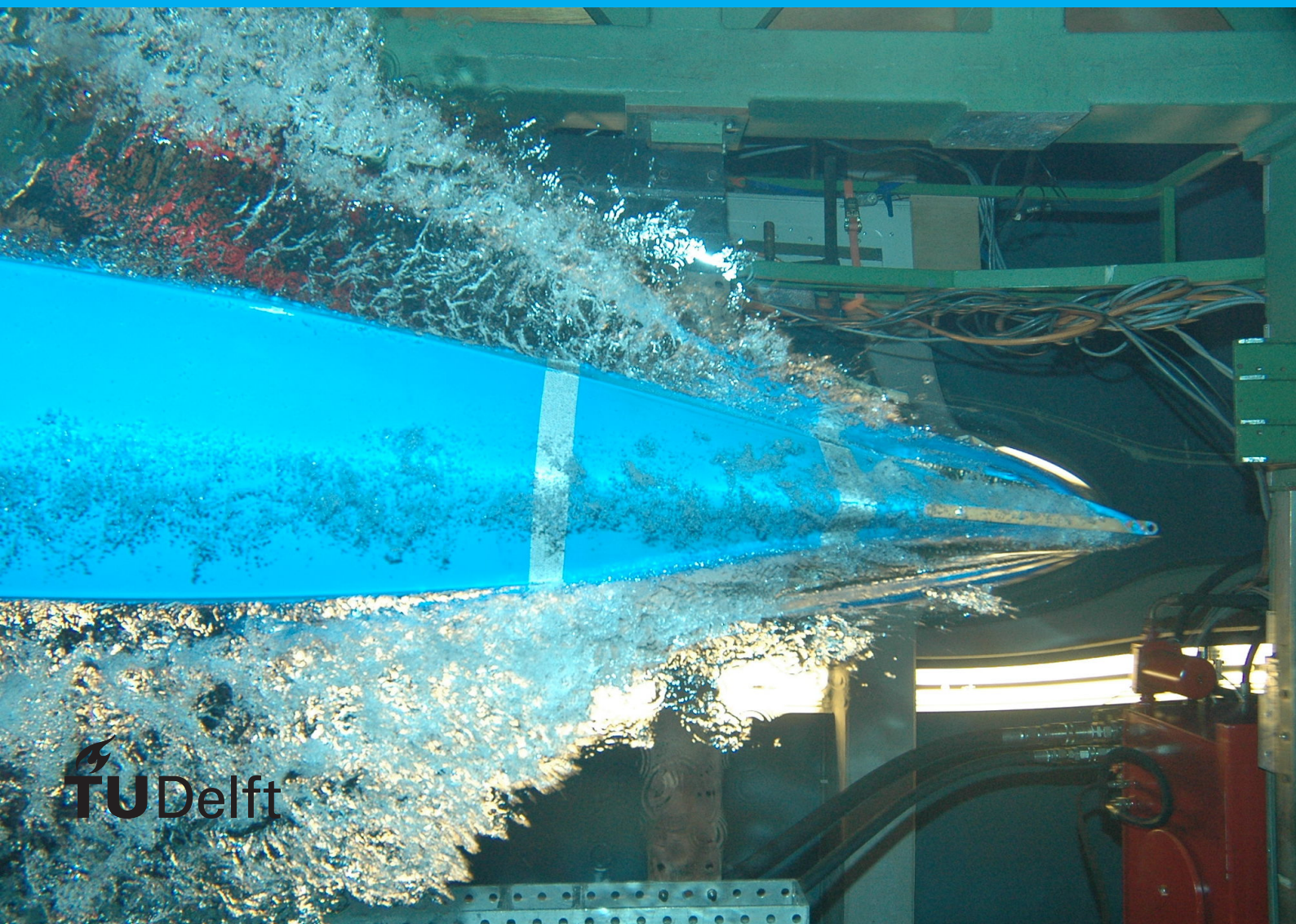
EuroToken

A Central Bank Digital Currency (CBDC) with off-line transfers

R. W. Blokzijl

- Stablecoin
- Blockchain
- Cryptocurrencies
- TrustChain
- CBDC

R.W.Blokzijl@student.tudelft.nl



EuroToken

A Central Bank Digital Currency (CBDC) with off-line transfers

by

R. W. Blokzijl

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on TODO.

Student number: 4269519
Project duration: November 11, 2020 – TODO
Thesis committee: Dr.ir. J.A. Pouwelse, TU Delft, supervisor
Member 1, TU Delft

This thesis is confidential and cannot be made public until TODO.

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

TODO: Add preface

R. W. Blokzijl
Delft, TODO

Contents

1 Introduction

Since the Bitcoin [46] white paper was published in 2008, the world has been speculating on how decentralized ledger technologies could be used to restructure the financial infrastructure of the world to enhance its transparency, digital efficiency, and feature set. 9 years later, Facebook announced a private currency controlled by a group of corporations [37]. 3 years after that the Chinese government announced that they had reached 92,771 transactions per second with their new Central Bank Digital Currency (CBDC) [39]. And the Eurosystem is set to make a decision on whether to start a digital euro project in mid 2021.

The direction of crypto currencies is no longer only determined by eccentric visionaries imagining a financial system that gives power back to the people. Governments and large corporations are joining the competition to create the worlds leading digital coin. Whether for profit, their national interests, or the good of humanity, the winner will be left controlling and overseeing all the worlds transactions.

Currently very few decentralized currencies are in a position to challenge the upcoming central coins. The most well known crypto currencies, including Bitcoin and Ethereum [26], lack the price stability necessary to be a reliable store of value. There have been attempts to create fully decentralized stablecoins, but none have been proven to work in practice on a large scale just yet.

If fully decentralized currencies don't come up with a solution to scalability and stability soon. The future of the financial system might come down to a competition between the large governments of the world and the private sector.

Whether and how these parties succeed will have large implications for the future of financial markets of the world, and might determine the level of freedom of the societies of the future. Where China and Facebook are making rapid progress. Meanwhile, the Eurozone is still deliberating while they might play a vital role in the future of financial markets, by including democratic the process.

This thesis aims to provide a design for a digital euro that builds on, and recombines mechanisms used by today's stablecoin, and existing Distributed Ledger Technologies, in order to create a digital euro analog called EuroToken. EuroToken is a scalable, secure and off-line capable payment system that allows peer-to-peer transfer of funds, while maintaining price stability.

We show how the EuroToken system can be used to create a scalable CBDC as well as serve as a private money alternative to current banks and provide the benefits of programmable money with the price stability of the euro.

The financial world of 2021 finds itself in a turbulent period in history. With the requirements of money shifting faster than ever before, the monetary systems we inherited from previous generations are increasingly struggling to meet current demands. In response, various new forms of monetary solutions are racing to fill the expectations of a modernising population and acquire A position in the middle of the worlds money flow.

In this chapter we illustrate some challenges for Central Banks have to overcome to maintain an inclusive economy, while staving off geopolitical challengers and private corporations.

1.1. The decline of cash

The eurozone system relies on two main types of money. Private money, managed by private banks, and public money, managed by the European Central Bank ECB [TODO CITE public vs private money].

Public money is the money we have in our physical wallets. It consists of bank notes and coins and is often referred to as cash. Once upon a time it was possible to exchange this money for gold directly at the central bank of a country and thus derived its value from gold directly. Today however, the value of this money is guaranteed by the reputation and trustworthiness of the central bank [32].

Private money is the digital money in our bank account. It derives its value from the banks reliability and reputation and is only usable through the bank itself. Without permission from the bank, storage, transfer and withdrawal are not possible.

A person in the eurozone would weigh the risks and benefits of these two types of money. On the one hand the ECB is a large, historically trustworthy, and democratically controlled institution. The only way in which the persons cash would lose its value is the broad failure of the European government and central bank. On the other hand storing money in banks has the benefits of the digital age. Storing money in the form of cash means exposure to the risk of having money stolen or being robbed. Since both the impact and the risk grow with the amount of money held in cash, the proclivity to store money in digital forms grows with the amount of money owned.

In addition to people storing their money in private banks for security reasons, the digitalisation of society is a powerful motivator. Transacting using private money as opposed to public money allows for transfer of funds all over the world. Private digital money can be used to perform purchases online, as well as transfer money to persons far away, while public money is only useful when it can be physically transfered to the receiver.

As a result of digitalisation, the world is moving from cash to cards. In the year 2000, less than 22 percent of transaction in the EU were card transactions. In 2019 this is over 47 percent [51]. This decline of central bank money lead to a number of unfavorable scenarios [38]. The decline of open and off-line money can lead to financial exclusion for the unranked and vulnerable in our society. In 2017, 3.6 percent of Europe's household had no registered bank account [45]. As more and more businesses move online or become pin only. These people see their means of payment decrease. Additionally the increased reliance in private institutions can leave the entire euro system less transparent and more vulnerable to corruption.

To solve these issues, an open digital coin with value guaranteed by the Central Bank would be needed. It needs to be open and easy to use, online capable, but not be reliant on any private institution to function.

1.2. The need for a competitive Euro

Commerce is moving online - Instant transactions

The banking system is insufficient to handle todays international trade - national payment systems suck -

People in other parts of the world are moving into crypto

If the eurozone does not get its own digital currency the ECB, and thus our democratic process, loses

If decentralized currencies win, individuals become increasingly dependent on the

It might lose the ability to

1.3. Rise of challengers to traditional currencies

The creation of of Bitcoin in 2008 [46] kicked off the race to create a currency that would solve the problems plaguing the current monetary system. While a complete solution has not emerged, today we see a number of alternatives that present various benefits and trade-offs. While Bitcoin and similar fully-decentralized currencies have proven that a global digital token of value can be realised, one of their main challenges remains the price stability of the asset and the scalability of the network.

To address the stability issue, stablecoins have risen in popularity. While decentralized stablecoins do exist [49], it is so called centralised stablecoins that gain their reputation as a digital alternatives to the dollar. Tether [43] is the most prominent example of this. With a market cap of 40 billion dollars they are the 5th largest crypto-currency by market cap [31]. In order to achieve the stability and dependability

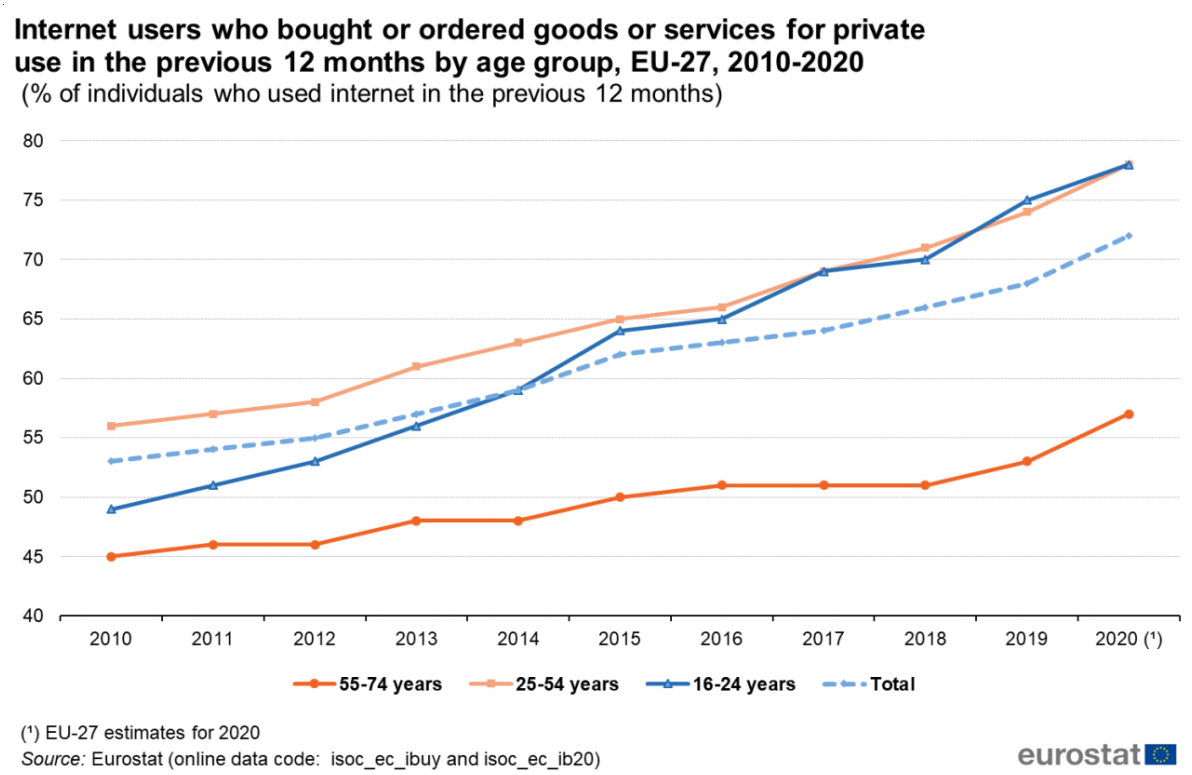


Figure 1.1: E-commerce 2010-2020 [4]

of their coin, Tether Holdings Limited acts as a centralised middle-man exchanging 1 tether for 1 dollar. Centralised stablecoins are often seen as an intermediary solution that provides a wrapper over the old monetary system in order to extend it with the features of digital currencies. These coins are essentially financial derivatives that depend on already existing currencies.

On June 18, 2019, a new currency conceived by a group of Facebook engineers was announced under the brand name “Libra” [37]. Later renamed to Diem, it would be a new free floating currency managed and governed by a consortium of multi-national companies united under the banner of the Diem association. While Diem presents itself as a solution for the worlds 1.7 billion unbanked, it is essentially a private world currency that would be controlled by corporations who will not be accountable to democratic processes.

The battle for the future control of monetary systems is a world wide phenomenon. While distributed open-source communities of engineers are trying to create a system free of corruption and private interest groups are trying to extend their reach, governments around the world are beginning to realise the threat to the established order. In order to not lose their influence of over their respective economies, governments around the world are looking into new digital versions of their currencies. The Federal Reserve has published their “Preconditions for a general-purpose central bank digital currency” [50]. The ECB has published a report specifying a number of Reasons to issue a digital euro, scenarios and implied requirements [38]. Meanwhile, the government that has perhaps progressed the farthest is the Chinese government. With successful public trails [47] they seem to be closest to a working digital currency.

1.4. The technical debt of traditional finance

People have been trading various commodities as a store of value nearly since 6000 BC [33]. Since then money has taken various forms, slowly moving up layers of abstraction, but the function of money has always stayed the same: acting as a medium of exchange [44].

The first known true standardised gold coins have attributed to Lydian society back in 640 BC. Slowly over time the gold and silver contents of the coins became less important, and currencies as a proxy for trust, became slowly more dependent trust in the system rather than the real value of the

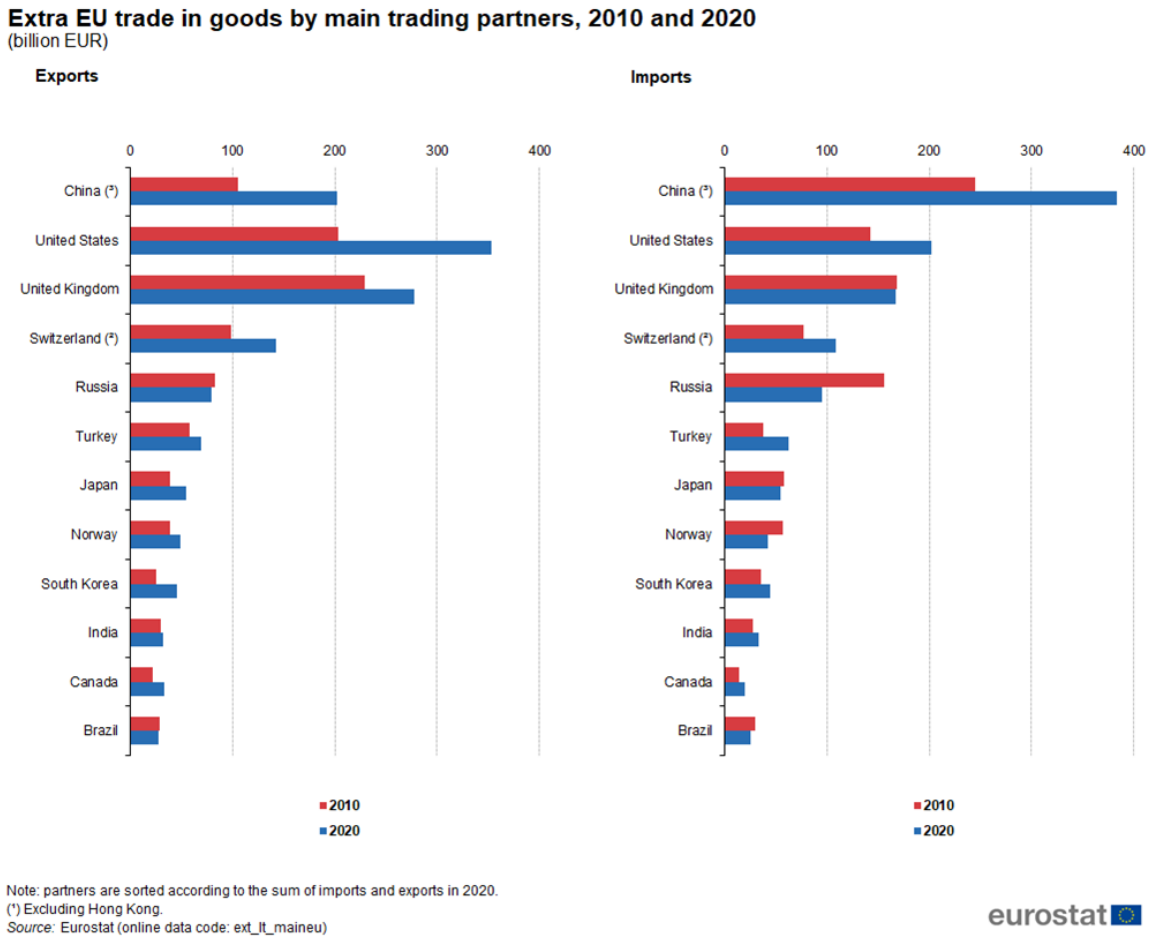


Figure 1.2: E-commerce 2010-2020 [4]

currency. In 806 AD, this culminated in the first use of paper money in China. Since these events, the form of money has varied based on societal conditions. Because of its functional utility, money in the form of bank notes backed by gold has been a popular form of money that has been used in European society since 1440 [32].

At the end of the 1900s Europe had a system of many national currencies, denominated per country. The value of the currency was maintained by different countries, often using the gold standard [32]. Meanwhile private banks would allow people to store their money in a safe institution, while they could lend that money out. For international and cross-currency trade, people would swap different currencies in exchanges when needed, but because of the localised nature of society this was infrequent.

This system worked fine in an era where most exchange was done by cash, and most trade was done within national borders. But as the world became more connected and digital, different solutions had to be built on top of the old system to respond to the changing demands of the population. Private banks, who were once used only for large transactions and money storage, got a more and more central role in day to day transactions. In effect the system as we see it today has accrued a lot of technical debt as the its requirements changed over time. As a result a number of inefficiencies have emerged from this technical debt.

First, international money transfer. Even within the euro-system this can often take up to a week. Second, banks as private institutions are vulnerable to bankruptcy. This makes them a “financial”on their semi-central point of failure, that take a good portion of the financial system with it. Third, transactions are dependent on bank IT systems. This makes them a “technical” semi-central point of failure, potentially leaving people unable to purchase their essentials. Fourth, people are tightly coupled to their banks. While having multiple bank accounts is possible, a lack of standardisation and interoperability makes people easily dependent on their one bank and its features.

2

Problem description

Over last century the world of finance has had to cope with significant paradigm shifts. Currencies traditionally started as a bottom up distributed system based on some commodity of value. While currencies like this seem to need no institution backing them, the fact of their physical nature makes physical projection an unfortunate necessity. This naturally leads to institutions that centralise some authority in order to secure the assets.

This age old solution is mirrored in the digitally accounted financial system of today. While the world is getting more more dependent on digital payments for every day transactions, the control of access rests in the hands of large opaque corporations. With private banks at the heart of the financial system, all innovation has depended on, and thus been limited to, the current gatekeepers of the financial world. While traditional physical currencies are still around as an option, the recent decline of their usage makes the need for a new open payment system ever more evident.

In order to rectify the offset in the balance of power and to promote productive financial innovation, a new open medium of exchange is required. Such a currency needs to be digitally efficient, transparent, accountable, and be fit for global transacting in the 21st century. In this chapter we explore the challenges in the creation of such a currency and specify the requirements for such a payment system.

2.1. The difficulty of modern digital payment solutions

The search for reliable, digital, money can be traced back as far as David Chaum in 1983 when he first released his paper on Blind Signatures for Untraceable Payments [28]. In this paper Chaum does not specify a design for a fully decentralized currency, but a mechanism for preserving user privacy against third parties in digital transactions. Since then many implementations have been attempted, including Chaum's own eCash [24] [27]. ECash had the potential to become a standardized digital payment system right from the start. However, the financial institutions of the time had their sights set on another digital payment system: Credit Cards. The adoption of credit cards as the predominant method of online payment lead to a steep rise in credit card fraud [22]. Along with theft of social security numbers, theft of credit card numbers is the predominant form of identify theft. While we leave the issue of a digital identity out of scope, the inadequacy of the credit card system at protecting users has been proven again and again. In its competition with Credit Cards, eCash went bankrupt in 1998 [16].

Over the years, alternative payment systems have been developed in response to the lack of good payment solutions. Direct consumer to merchant payment systems like PayPal [19], Venmo [20] and Skrill [12] have emerged to fill the gap left by credit cards. However, the success of these services is dependent on the ability to integrate their solutions with traditional banks in order to make the payment process seamless. Because of the lack of standardisation in the industry, integration has to be done for every bank individually. When this integration is lacking, users are stuck with these services as an additional bank account which has to be maintained only for their payment features. These solutions don't solve the underlying problem: the tight coupling between a user and their banking/payment provider.

The problem of decoupling users from their banks has been approached from many different angles. Perhaps the most famous solution in recent years is Bitcoin [46]. While criticising a number of

issues that lie at the heart of value accounting in our current system, the Bitcoin white paper proposes a digital payment infrastructure with associated currency that completely moves the very core of value accounting to a distributed and open system. 10 years after its inception the crypto-currency is extremely popular as an investment vehicle. Any significant payment volumes have yet to be demonstrated however. With transaction fees at around 59 US dollars [**?**] around April 21 2021, Bitcoin is not ready to be a direct consumer facing payment method.

Other solutions to the tight coupling problem have been successful on national levels. In the Netherlands the iDEAL system has succeeded in integrating most dutch banks under a single online payment system. In Norway a similar product called Vipps exists that integrates all Norwegian banks into a single payments app that enables users to transparently send money. Similar systems exist in many European countries [2]. The problem with these solutions is the lack of cross border payments as a result of the lack of international standardisation.

The European Union is actively trying to integrate their financial system across borders. With payment-integration initiatives like SEPA, PSD2 and the European Payments Council, the EU is slowly moving towards a better integrated euro zone, however much work is left to be done. With this research we aim to provide a design for a central bank backed payment solution that works across the entire EU. It presents a standardized payment processing back-end that opens the door to digital innovation in all areas of finance.

2.2. Requirements for a digital euro by the ECB

In October 2020 the European Central Bank published a report detailing a number of scenarios where a new digital euro could provide a benefit [38]. Associated with these a number of requirements are provided.

1. **Enhanced digital efficiency**
2. **Cash-like features**
3. **Competitive features**
4. **Monetary policy option**
5. **Back-up system**
6. **International use**
7. Minimise ecological footprint (cost saving and environmentally friendly)
8. **Ability to control the amount of digital euro in circulation.**
9. Cooperation with market participants
10. Compliance with the regulatory framework
11. Safety and efficiency in the fulfilment of the Eurosystem's goals
12. Easy accessibility throughout the euro area
13. Conditional use by non-euro area residents

In this project we aim to conform to these requirements as best we can and we evaluate our solution by these requirements. The technical requirements are emboldened in the list, as they will be guiding in our design. The rest will only be speculated on as they do not pertain to the topic of computer science and fall outside of our area of expertise. Therefore a technical solution to these problems has to conform to the following requirements.

1. Be a secure system of accounting
2. Scale to the size of the European union
3. Preventing unsanctioned money creation
4. Price stability
5. Disaster resilience through off-line transfer ability

We aim to create a payment system that is secure, scalable, off-line transferable, stable, secure and digitally capable.

2.3. Trade-offs around double spending, scalability and decentralisation

When designing a modern digital payment systems with the ability to transfer funds off-line peer-to-peer systems and distributed ledger technologies are an worthwhile case study. Since the Bitcoin in 2009

various crypto-currencies have iterated on the idea of a fully decentralized currency. After 12 years of development a number of trade-offs are becoming visible that show the limitations of the technically.

Ideally a payment system has no central points of control. By keeping a payment system decentralized the failure of one part does not affect the functioning of the ability of the users to make payments. Such a system also has inherent scalability. However, keeping a currency secure from unsanctioned money creation is not a trivial problem in a fully distributed system.

The primary problem of unsanctioned money creation can be split into 2 different problems. First, for any transaction to be valid the payer has to have received the funds in the past. And second, they have not already spent the money.

The first of these is relatively easy to solve. To do this all transactions received in the past can be digitally "signed" by the sender. The signature of the sender on the transaction proves the transfer of funds from one party to another. When receiving funds, the transaction can be trusted by verifying that the sender has received the money in the past from someone else. This way every transaction can be recursively validated back to their creation point.

The second problem is famously called the "double spending problem" and requires some trade-offs to solve. The goal is to construct a way to prove that for any given transaction, the balance of that transaction has not previously been spend by that user. In the first problem a transaction can be rejected if not enough information is available to verify the fund availability. In the second the goal is to prove the non-existence of a transaction.

Solving the double spending problem without a centralised party like a bank that keeps track of all historic transactions is a difficult task. Blockchain based systems have had some success. The most well known distributed ledger technologies are Ethereum [26] and Bitcoin [46]. We will refer to these as single blockchain networks. Both of these networks maintain a single blockchain of transactions which gives transactions a total order.

Blockchain bases DLTs group all transactions into blocks, which are in turn organised in a linked list where every block refers to the block before it. Accounts are identified by a public key. Every transaction is a transfer from one public key to another, and is signed with the associated private key of the sender. Every transaction references previous transactions where the sender received money, thus ensuring the funds are available. Every transaction thus has "input" transactions. In order to allow a user to send less than the output of a previous transaction, a transaction can have multiple "outputs", sending to multiple public keys including the senders. To solve the double spending problem, any "output" of any transaction can only be spent once, and any transaction that spends an already spent output is rejected. If a user has spent that output before, that transaction would exist somewhere in the chain, thus proving the new transaction to be fraudulent.

The problem with single blockchain systems is their inherent lack of scalability. Since the entire chain has to be checked for any conflicting transactions, the entire blockchain, and thus the entire history of the network, has to be kept by everyone that wants to validate a transaction. This means all new blocks have to be distributed to all nodes in the network periodically to redistribute the new transactions made. The speed of the network to propagate transactions, and the limited puts a practical limit on the amount of transactions a single blockchain network can handle since the whole world has to be informed.

To improve the scalability of this system Ethereum2.0 [**?**] proposes a network upgrade that adds multiple parallel blockchains called shards. These shards will be responsible for their own fraction of the transactions on the network, only interacting with one another when necessary. While this does increase the capacity of the network, the extent of the scalability gained has yet to prove itself.

Other attempts at a more scalable ledger exist. [8] builds on the concept of the Tangle. Instead of a blockchain, transactions reference each other in a Directed Acyclic Graph model. Users are not required to maintain the entire block-DAG but are required to validate a number of existing transactions before their own transaction is validated. This lowers the change of a double spend significantly, but requires of their users to so quite some work in order to be able to transact. While public nodes are available to validate transactions for you, these are currently run by enthusiasts and altruists. A criticism is that there is no reason the altruistic processing of transactions will continue into the future.

Another example is Nano [10]. Nano employs a block DAG that resembles a lattice structure which assigns a personal blockchain on a per user basis. Transactions happen between users and are incorporated into both chains. Double spending is prevented by having blocks broadcast into the network and representatives nodes validating the transactions. These representative nodes are elected thought a delegated proof of stake mechanism.

The double spending problem has many potential solutions, all with their own set of trade-offs. A payments solution that aims to scale to a global level using distributed technologies must careful balance double-spending, scalability and decentralization.

2.4. The problem of off-line digital payments

The ability to provide off-line payments is an unsolved problem in the world of digital payment solutions. Solving the double spending problem in a Peer-to-Peer network is a challenge on its own. Doing so without a live connection to that network increases the complexity even further.

The problem of double spending and off-line payments is best understood using the CAP theorem. Consider the total set of transactions to be the database, and reads and writes to be updates to the balances of any user. If we decide we want off-line payment, we implicitly choose the value of *partition tolerance*, thus creating a trade-off between *availability* and *consistency*. We can either read a users balance, **or** be sure we know the correct balance of the user.

An “off-line transaction” is thus a transaction done without access to a sufficient a sufficient set of peers to validate consistency. This means that if any user accepts an off-line transaction is **not** possible to know whether a conflicting transaction exists in another part of the network.

Any network wanting to prevent double spending will accept only 1 in a set of conflicting transactions, usually the first to arrive. Anyone accepting an off-line transaction is therefore at risk until they check in with the network.

For this reason a transaction is usually not accepted by the receiving user until we have some guarantee that all conflicting transactions will be dropped by the rest of the network. This is the concept of transaction finality. A transaction is considered *final* is the rest of the world will reject all conflicting transactions.

Most blockchain based systems achieve this guarantee by having a globally distributed blockchain storing all transactions that *finalises* any transaction with some *eventually negligible* probability of conflict.

In all of these systems, the receiving user still chooses when they accept the transaction. While its possible to wait until the network can be reached online before goods are exchanges, any users that know each other can defer the validation. Instead of relying on the network based, the exchange is based on the trust of the receiver in the sender.

Of course the trust between users is hard to quantify, and shouldn’t be relied on fully. However, increasing the trust between two transacting users in the period between the transaction and the check-in with the network is the key to implementing dependable off-line transactions.

2.5. The price stability problem

The primary purpose of a currency as a payment solution is to be an “intermediary store of value”. This means that anyone that chooses to exchange their assets or services for a currency, can later trade it in for something else of a same, or similar value. If a currency cannot keep its value stable over time it will fail to be a good intermediary store of value. In order to maintain the viability of a currency as a good option for payment, the price thus has to remain stable over time. This concept alone eliminates nearly all of the new crypto currencies as good payment solutions as their price is dependent on daily market fluctuations.

This raises the question: How do we keep a method of payment solutions truly price stable? This is really a question for economists to answer, however, in order to create a digital payment solution, we do need some method of stabilizing the currency.

One problem is the difficulty of measuring stability. Stability means that what you can purchase today is about the same as you can purchase tomorrow. However, tracking this is extremely difficult. The “value” of any item is constantly shifting based on supply and demand, thus getting any measure of the value of anything is very complex. This is why many stablecoins outsource the problem of defining value to an external system, usually an existing fiat currency. The system works by “pegging” the stablecoin against some collateral.

Say a “token” is always directly exchangeable at central exchange A for the US dollar at a 1:1 ratio. The price of the token in the market will tend to follow the price of the dollar. When the price of the token in the market dips *below* 1 dollar, anyone can buy the dollar on the market, and directly sell it at exchange A for 1 dollar, making an instantaneous profit. This decreases the amount of tokens in the

market. Because of this reduced supply the price in the market will increase. If the price int he market is *above* 1 dollar instead, any investor can buy the token at exchange A for 1 dollar, and sell it on the market for a profit, thus increasing the supply and decreasing the price.

This principle is behind practically all stablecoins on the market. However, this immediately leads to the next problem: How to run a 1:1 exchange. The simplest solution is also the most successful at the moment. Tether [43], currently the 5th largest crypto-currency by market cap, uses a central exchange to ensure a 1:1 exchange ratio between the US dollar and its crypto-token USDT. The obvious problem with this is the fact that the system has a critical centralised element. Without proper oversight such a system could be secretly severely under-collateralized. Other centralised stablecoins like USDC [15] and the Stasis Euro [13] utilise audits by private auditing firms to increase transparency, however the system remains centralised.

Fully distributed stablecoins like MakerDAO [49] and EOSDT [3] do exist. These are kept at a stable price by providing an exchange of 1 token for 1 dollars worth of “collateral”. This collateral is some blockchain accounted token of value. To make sure the system does not get under-collateralized when the price of the collateral drops, the system is over-collateralized at all times. While these systems have seem some success already, they derive their notion of value from the dollar, and are dependent on the value of their collateral.

2.6. Research Focus and Structure

While a complete redesign of Europe’s monetary system is obviously out of scope for this thesis, the previously described problems and requirements lead us to the following question:

Can we create a digital payment system that combines the functionality of world-wide online payments, and local off-line payments in a single solution.

This document describes the motivation, design, implementation and evaluation of the EuroToken system. The EuroToken system is a conceptual design that aims to fit the requirements stated in the previous section, as well as a limited proof of concept design testing certain aspects of the design. The structure of this work is as follows, in the next chapter we describe the design of the EuroToken system. The design is approached from the fundamental questions of a currency and answers the fundamental questions first. What is a digital currency? What is the double spending problem? And how to design a system that is scalable while not compromising the principle of double-spend prevention? The design aims to provide the following features:

1. Be a fully functional system of accounting
2. Preventing unsanctioned money creation
3. Scale to the size of the European union
4. Be off-line transferable

In order to not be limited in the same way as Bitcoin and similar currencies, we choose to sacrifice the following feature: Decentralization. This give us the required leeway to create a scalable and off-line capable system. However, we do attempt to provide the necessary tools to overcome the downsides of the centralisation.

3

Design

Any payment system that aims to replace public money while being able to operate at the scale of the euro system needs to conform to a number of requirements. Such a system needs to be scalable, privacy aware, allow peer to peer transactions off-line. In needs to be price stable, exchangeable for euros, and most importantly, it needs to be secure and cheating resistant. In this chapter we first describe how a distributed block-lattice provides a good basis for a scalable, private, and off-line friendly transaction system. We then explain how we position the system in relation to the euro, how the price can remain stable, and how a system can mimic the properties of cash. We then go in to the details of how the system is secured, and how we prevent double spending while still remaining scalable and allowing off-line transactions. Finally, we explain how the system could be expanded upon by legal frameworks that can provide varying risk vs privacy trade-offs and how certain guarantees could be enforced in the system.

3.1. Distributed accounting and networking

The possibilities and limitations of any virtual currency are dependent on its system of accounting. In order to conform to the off-line, scalability and transparency requirements, a system of distributed accounting is chosen. As the fundamental building block for the EuroToken system we use a Hyper-Sharded block-lattice that keeps track of every users transaction history on their own edge device. By storing all information required for transacting at the physical end points of transactions, we create the possibility of direct off-line transaction between users, without any link to the outside world.

While the EuroToken system design is independent of the underlying communication technology, the off-line requirement leads to there being some limitations on the way users interact. Since off-line users cannot connect to servers we choose to work with a Peer-to-Peer system that allows users to find each-other based on personal identifiers.

We build on Peer-to-Peer networking, that provides a mechanism to discover the network location of users based on the same public key that is used to identify their wallet. This allows us to almost completely abstract away from locating users using IP addresses and ports. As a result we only have to worry about maintaining a users public key to identify and communicate with them across time. Our peer to peer network does not only abstract away from IP addresses, bus also from the IP network completely. Namely, it provides communication over Bluetooth without the need for any internet connection. This becomes very useful for demonstrating the off-line capabilities of the EuroToken system.

3.2. Block-lattice accounting

As mentioned for our distributed accounting system we choose to build on a block-lattice structure. As illustrated in figure 3.1 every user has a personal blockchain structured as a chronological, one-dimensional string of “blocks”. Every block will include a cryptographically secure hash identifying what block preceded it. Because of the trapdoor effect of the hash, any block will uniquely identify all blocks that come before it. This allows anyone to verify the validity of entire history of another user, given the last block in this history. Every block will contain a single transaction that specifies the transfer

of funds from one user to another, as well as a reference to a corresponding block in the chain of the transaction counterparty. This effectively creates a system of double accounting.

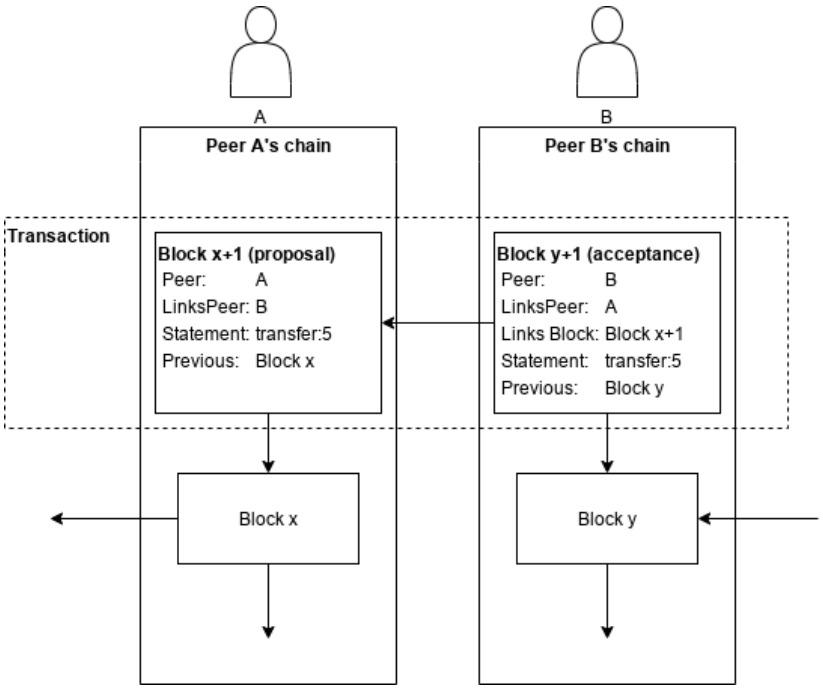


Figure 3.1: Block-lattice structure

Every block can contains a “declaration” by the user, or a reference to the declaration of another party. These declarations are digitally signed by the declaring party and form the base of any transaction. To do a transaction the sending user (Alice) will create a new (half)block with a declaration stating “I transfer 1 EuroToken to Bob”.

When Bob receives this block from Alice, he can accept it by creating a block in his own chain and returning it Alice. Before Bob accepts the block, he first validates the history of Alice by requesting enough of her chain make sure that Alice doesn’t validate any of the network rules that would invalidate Bob’s receiving of the money. Once Bob is satisfied with the correctness of Alice’s transaction history he incorporates a new block declaring the acceptance of Alice’s transaction. This block includes the hash of Alice’s block, thus entangling the chains of Alice and Bob together. Bob now has a signed proof by Alice that the transaction happened. He can use this to prove the transaction happened at any point in the future.

3.3. Gateways: Euro to EuroToken exchange

The viability of any currency as a store of value over a given time frame is dependent on the stability of its price over that time frame. This is an issue that has plagued decentralised crypto currencies from the very beginning. The hope is that the currency will stabilise itself when it reaches a critical adoption level. However even currencies like the euro and US dollar don’t remain stable without periodic interventions of their respective central banks.

The euro has long served as the core of the financial infrastructure of the European economy. It has essentially done this using two consumer facing versions of money: the euro as a publicly accepted, physical item of value (the public euro), and the euro as a digital, privately managed, unit of account (the private euro). These public, and private types of money serve citizens in different ways. The public euro is the most stable store of value since its guaranteed by the central bank, it also has the advantaged of requiring no internet connection to use. While the private euro has digital advantages in usability and security, but derive their value from the “reliability” of private banks, and are only insured by governments up to 100.000 euros [CITE]. With the declining usage of public money in favor of digital money, the need for a new type of euro to fill the gap of public money is getting stronger.

For these reasons we present the EuroToken system as a 3rd type of money. Instead of reinventing

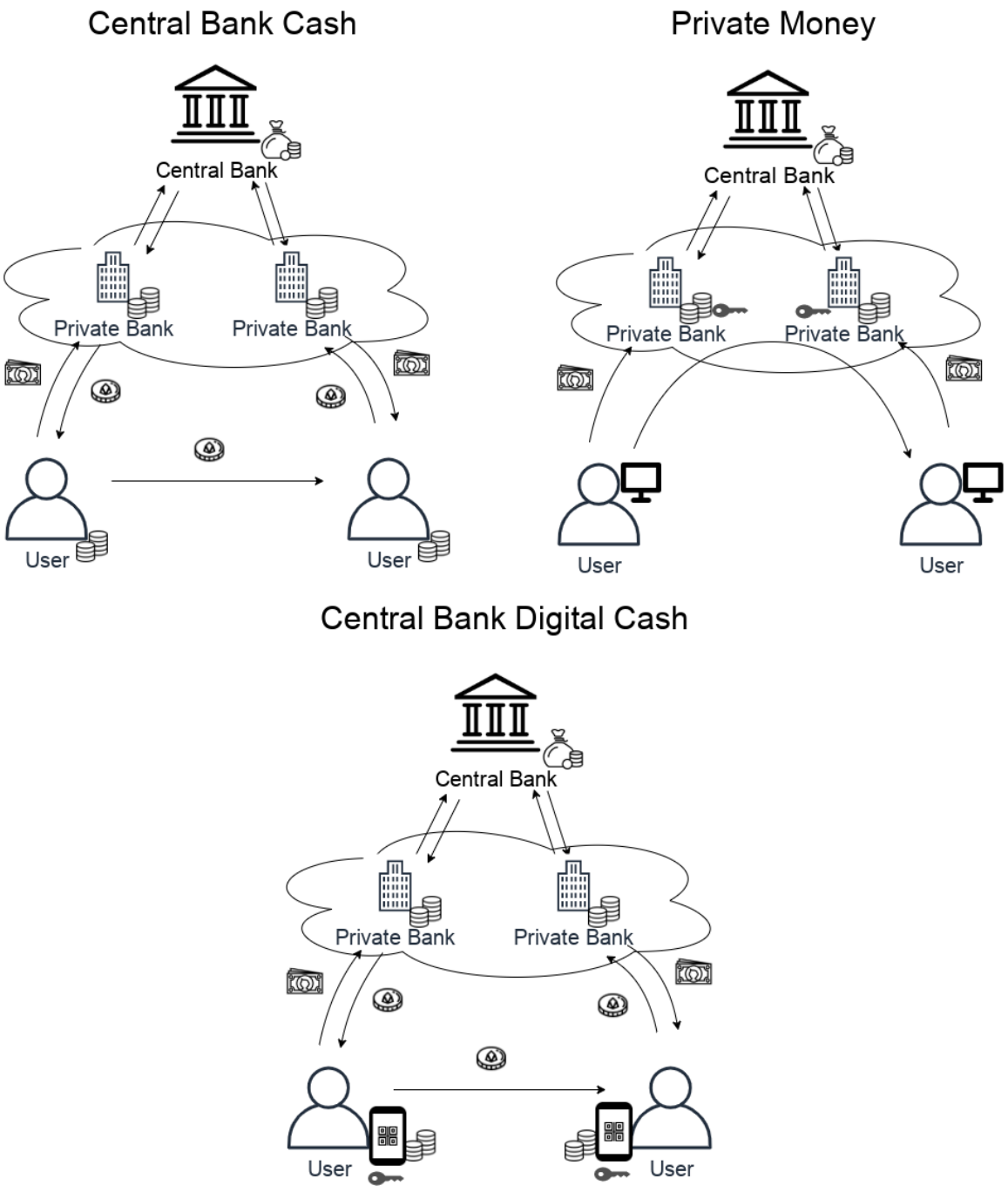


Figure 3.2: Usage flow of cash, private money, and EuroToken.

the wheel of “stability” we connect the EuroToken system directly to the euro system, while providing extra features on top of the current euro system.

In order to properly connect EuroTokens to the euro system, an easy and value-transparent method of exchange is required. Just like private and public euros are exchangeable though local banks, a mechanism is needed to exchange between euros and EuroTokens at a 1:1 ratio. To do this, we implement a “gateway” between the private euro system and the digital euro. This gateway implements the EuroToken protocol on the one hand, and interfaces with banks on the other.

In our current design, the gateways are designed to be run by public parties associated with the central bank. Any detailed speculation on the best way to connect such a system to the established euro is best left to economists. However, we envision a possible future where multiple gateways are run by existing private money institutions who perform the heavy lifting of day to day exchange. In such a system private banks would act as an accounting system for the EuroToken exchange without being allowed to leverage their EuroToken position. The Central bank would allow these private institutions to conform to reserve requirements in the form of EuroToken holdings rather than only cash. By not allowing private banks to mint new EuroTokens, but only exchange them, the central bank can control the amount of EuroToken in circulation in a similar way to current public money. This would insulate the EuroToken from the impact of a failing euro or bank, in the same way as physical public money is currently insulated from such failing.

This way of connecting to the euro could allow for a smooth transition to a digital form of public money, while the established and regulated financial institutions are still positioned properly in a place where financial services can be provided.

3.4. Transaction finality and Double-spending

In order to remain a viable store of value, a currency needs to provide protection against any non-sanctioned creation of that currency. If a network allows its users to “create” new money in any significant way, the value of the coin will drop as the supply increases, thus undermining one of the most fundamental function of the currency. The structure of the blockchain provides an immutable and signed history of any transactions, thus enabling users to prove that the funds they are attempting to send actually exist. However the blockchain does not inherently allow users to prove that they have not spent, and will not spend, the same balance again.

In this section we explain how the network prevents unsanctioned creation of currency.

3.4.1. The double spending problem

In order to spend their money twice, a user has to create 2 blocks that are positioned in the same place in their blockchain. This is what is called a “double-spend attack”. This attack is only detectable if both of the conflicting blocks are found. Since we have opted for a distributed blockchain this detection becomes a non-trivial problem to solve. The transactions of 2 conflicting blocks might be re-spent many times by the time anyone sees the 2 conflicting blocks and notices that a double spend happened.

Bitcoin and similar currencies solve this problem using a global blockchain that everyone has access to. This allows users to check whether a given balance has already been spent by inspecting the global database of transactions. However, the global knowledge of the Bitcoin chain is inherently unscalable. Additionally, the details of the Proof of Work method of block generation leaves a certain measure of uncertainty with regards to the “finality” of any transaction in the newest blocks. This often requires users to wait up to an hour to be sufficiently confident their transaction really happened.

A solution to this problem in a network with a distributed block-lattice, starts with the realisation that the issue of detecting double-spending can be reduced to the issue of detecting “chain forking” in our network. The usage of the blockchain allows us to make sure that all transactions are ordered and consistent, this means that double-spend needs to be in 2 separate versions of that history. Thus requiring 2 blocks that refer back to the same historic block. This is a fork in the chain. We cannot “prevent” a user from creating 2 conflicting blocks in their chain as their chain is stored on their own device. But we can make sure that the rest of the network only accepts one of the 2 blocks, thus only accepting 1 “spending” of the balance. This choice between 2 conflicting blocks needs to be consistent so anyone in the network is working with the “same history”. Additionally, forks need to be detected and resolved before the balance is spent again by any of the 2 receiving parties. This way a double-spend will not propagate into the network and is limited to the users involved in the 2 transactions. To resolve

the conflict between blocks we define the concept of “transaction finality”. For a transaction to be final, it needs to be “validated” and “stored” in the network, while any conflicting transaction will be rejected by the network. Transaction finality the guarantee that a merchant needs before they can send their goods to a paying customer.

The transaction finality problem in our network has several possible solutions. In [25] Brouwer presents a method of distributing blocks to a randomly and fairly selected list of witnesses that would probabilistically detect any conflicting block before the receiver would accept them. In [40] Guerraoui Et. Al present a more theoretical method of block broadcast. These might be good candidates for future research. However since these solutions are inherently probabilistic, there is no hard guarantee that any double-spend will be detected in time.

3.4.2. Balance vs spendable balance

Currently lacking a good exact and distributed solution, we choose to utilize a decentralized network of trusted validators. These validators maintain the last transaction of users that register with them. Any user who receives money, can verify the non-existence of a conflicting block with the associated validator of the sender.

In the rest of this section, we define the concepts of “spendable balance” and specify the information requirements for marking a transaction as finalised.

In order for Alice verify if Bob is able to send her the money he is sending, she needs to know that Bob has sufficient funds. For this reason a rolling a balance across all transactions could be maintained across all blocks. Where the balance B for a given block with sequence i (B_i) is:

$$B_i = B_{i-1} + C_i$$

.

Where C_i is the change in balance for the block with sequence number i. This is negative when sending money. However the balance of a user does not take into account the concept of transaction finality. So instead we maintain the total “spendable balance” instead.

3.4.3. Finality statements

Before Alice can add the output of a block she received from Bob to her “spendable balance”, the transaction from Bob first has to be finalised. To achieve this a validation is performed with Bob’s associated validator. This is done by sending the validator a finality proposal.

3.3 The finality proposal block includes notes a list of hashes that point to transactions from Bob. Together with this block for the validator to sign, Alice will send all of Bobs blocks from the last transaction to validate to the last block the validator knows about. The way for Alice to determine what information this is, is explained in the section on checkpointing later in this chapter. In addition to Bob’s blocks, she will also send her “accepting blocks” that include the transaction in her chain. This is to make sure she can only claim a transaction from Bob once. Bob’s validator will then verify:

1. That there are no other transactions that conflict with the one to Alice.
2. That there are no other “accepting blocks” already linked to this transaction.
3. That Bob’s chain is valid up to the last transaction to verify.

If this is the case it will sign the proposal. If a later transaction from Bob is received that marks a fork in his chain, the fork from Alice becomes the only accepted fork, and the other one is rejected. Using this finality statements as proof of this, Alice is now allowed to spend the output of the transaction.

In the case that a different fork from Bob has arrived at the validator first, the fork where Alice receives money is rejected. Since Alice has already accepted the transaction in her chain and may have built other transactions after it (though not spent the output), she could be requested to submit a new finality proposal without this block. Since Alice is not permitted to spend the funds from Bob until it has been finalised this is the point where double spending is handled.

Note that the specific handling of this event might not involve the forfeiture of a transaction. We discuss this further in the section on off-line payments and conflict resolution.

3.4.4. Verification

For a block to be considered valid:

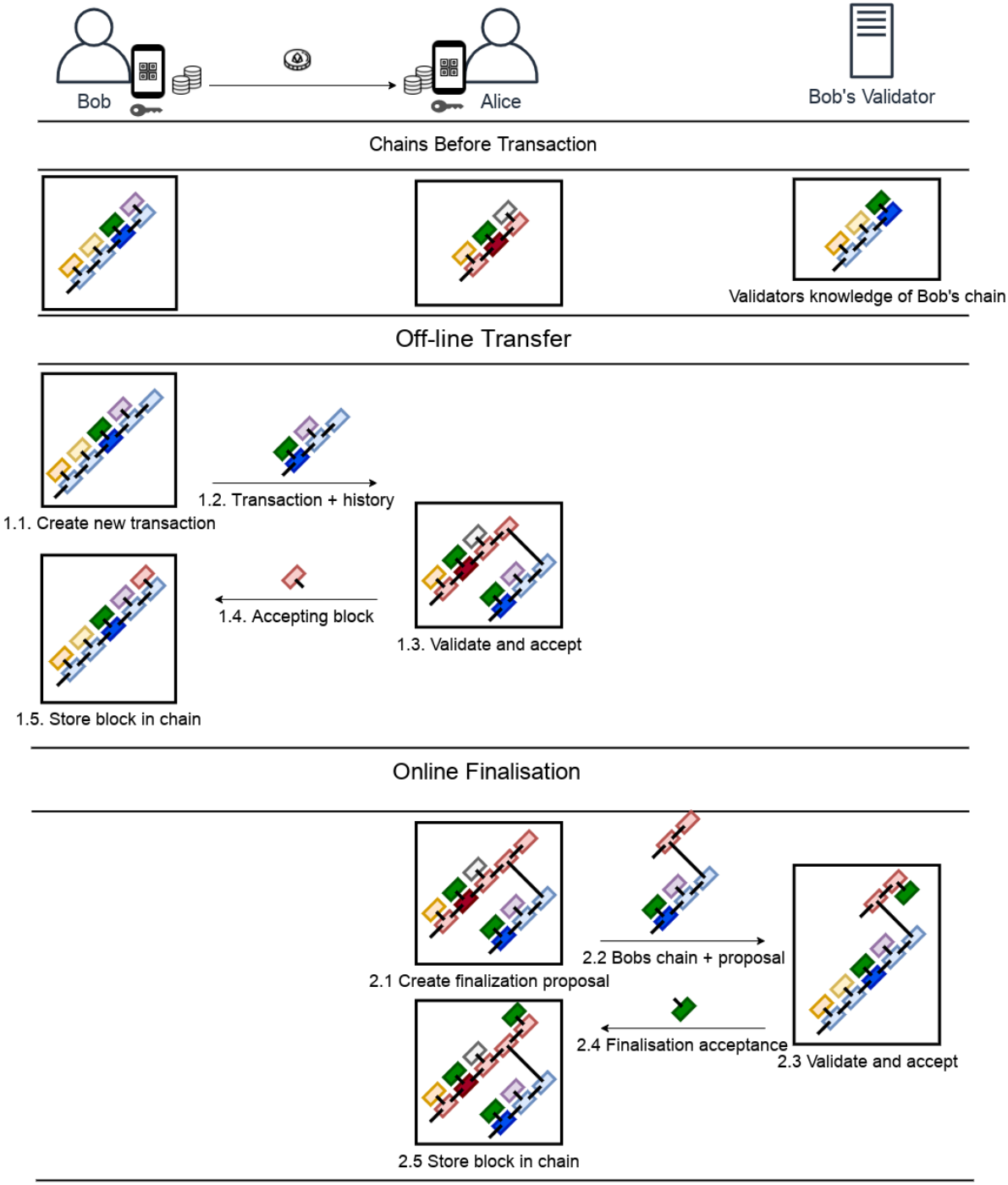


Figure 3.3: Off-line transfer and finalisation

1. All standard block-lattice invariants are maintained.
2. All blocks preceding it are verified to be valid
3. The total spent amount is to be less than the spendable balance.

For a transaction of a receiving block to be considered final:

1. A checkpoint from from the validator of the sender has to be exist in the chain of the user AFTER the transaction.

By introducing checkpoints, the required information at the point of transactions is reduced. When Alice and Bob set transact between them, Alice can determine the validity of Bob's transaction by inspecting only Bob's chain, down do his last checkpoint. However, Alice must also request all Bob's information down to the last Full checkpoint, in order to

3.4.5. Spendable balance

Once a transaction if finalised, "spendable balance" of Alice can be calculated. The spendable balance changes at two events, the finalisation of an earlier receiving transaction and when Alice spends her money. As such the spendable balance SB_i for a given block with sequence number i is:

$$SB_i = SB_{i-1} + F_i - S_i$$

Where S_i is the total amount spent in the block with sequence number i, F_i is the total amount finalised in the block with sequence number i.

3.4.6. Conclusion

In the future we envision the system to take one of three routes regarding transaction finality. First, system could be built on a future breakthrough in distributed transaction finality. Second the system could be built on a probabilistic but bounded transaction finality, where the rare double-spend is eventually detected and settled through the legal system. Or third, like in our solution, the system is build on trusted nodes that verify transactions for user. Like the gateways, these validators could be run by regulated financial institutions. Such a system would most resemble the current financial system, with the added benefits of off-line transactions, programmable money, a standardised system of accounting, instantaneous international transactions, etc.

3.5. Checkpointing

Because of transaction finality, when Alice receives the transaction from Bob, she can rely on the finality statements, rather than having to validate the chain of everyone he received money from. This reduces the validation load to only Bob's chain. However this still has some issues. First, Bob's chain will grow larger over time, thus slowly increasing the validation load. Second, all this information needs to be stored by Alice until it can be delivered to Bob's validator.

The way this problem has been solved in traditional blockchain systems is through the global blockchain and limited transactions per second. By having only miners or stakers being required to maintain the whole blockchain, only a few machines have to be able to know the entire chain and store all that data. But this is still inherently unscalable.

A second issue is one of privacy, when Bob has to send Alice all of his chain for verification, Alice can derive much from this information. Though we would like to see methods of privatization added to perhaps conceal transfered amounts, we still need a way to minimize the information leakage to 3rd parties.

To solve this issue of validation scalability, we define a form of checkpointing. We periodically create a checkpoint block in a users chain that , that includes a summary of the entire chain before it. This information is:

1. The total "spendable balance" at that point in the chain
2. The public key of the validator who is responsible for this wallet.
3. A statement that the validator has received all blocks before this point

Alice now knows the blocks that are already stored by the validator. When Alice is receiving money from Bob, she only requires Bob's blocks down to the his last checkpoint.

3.6. Off-line transactions and online validation

The EuroToken system has the intentional distinction between transactions and their finalisation. Because of this, the first step of transactions only require a direct connection between users. In theory, this allows to transact off-line, if they're willing to risk that a conflicting block already exists in the validator. Of course, in this case, the transfer of funds depends on the trustworthiness of the sending party.

In this section we discuss a few ways of interacting with the system that allows for different risk exposure to the parties.

3.6.1. Online transactions

When users are connected to the internet, a real life interaction can easily combine the finalisation step with the transaction, only transferring goods or services once the transaction is finalised. We envision this as the default way for users to interact, especially for large transactions, and transactions with strangers, since this reduces the risk to either party to zero.

3.6.2. Off-line transactions

Since money only becomes spendable after finalisation, the receiving user is the one that will lose funds when a double spend happens. To lower the risk and damage of this, certain systems might be put in place. For this, we build on the fact that transactions are always signed by both parties. This makes sure that a proof of double-spending always exists, and is obtained no later than the finalisation attempt.

A way to ensure a user that they will receive the funds is by allowing senders to register their identity with their validator. The validator would sign a statement that the identity of the sender is known and that they will take legal action in the event of a double spend. This then optionally allows the validator to accept the risk of double spending. In the case of a double spend the validator would sign a special statement with the receiver, that invalidates the double-spent transaction, but transfers and finalises the funds from the validator instead. The validator will then pursue legal action against the sender for fraud.

In the meantime the validator could block the sender to perform online transactions and checkpoints until they first settle the double spent funds. The details of what is both technically and legally possible here is a good subject for future research.

3.7. Regulation of validators

One could argue that system hasn't solved the issues of transaction finality and double-spending, and that we only defer the problem to a different point. It is entirely conceivable that trusted validators could cheat by allowing certain wallets to double-spend. To add to this, using the checkpoint functionality a validator can specify a higher spendable balance than is actually logged in the chain.

However, when comparing our system to the current way private institutions are regulated, the blockchain structure of transaction can provide a powerful method of maintaining the integrity of the institutions. While we cannot prevent fraud at the institutional level, we do provide an option for detection to allow for regulation.

In order for a regulator to check that a validator has done their job with integrity, they need to be sure of 2 things:

1. That all "statements" have been made consistently with the rules of the network.
2. That no other "statements" have been hidden from the regulator.

"Statements" in this context, describe anything that the rest of the network puts their trust in. These are:

1. Finality statements
2. Checkpoints

Both of these statements are created in the form of "accepting blocks" and are stored by users and their validators with an associated hash. We now propose a two round system for validating all transactions within a given time period. In round one, we validate that all statements have been made

correctly and publicly store the hashes. In the second round, once we have the hashes of all statements available, we validate that all statements from other validators exist.

In the first round all information in the database of the validator is processed for consistency. Since all statements by the validator are made in the form of blocks in their personal blockchain, they have an explicit order. The blocks of the validator, together with the blocks of all the users the validator is responsible for, are processed in the same way as the validator was responsible for processing them. This step in the process ensures that all statements are made correctly.

In the second round, we ensure that there is no statement withheld by the validator. This is done by publicly publishing a signed list of the hashes of all statements made by the validator. This allows regulators to cross-check that all inter-validator statements have been reviewed by a validator. To make this step more efficient, we propose that when checking a validators consistency, regulators generate a list of statements for each distinct validator to increase the efficiency of distributing these hashes to relevant parties.

A possibility also exists to allow the public access to these records to ensure the integrity of their institutions.

4

Implementation

In this section we describe the implementation of the EuroToken protocol, as well as the prototype we built to test and showcase the capabilities of the EuroToken system. The protocol is implanted on top of IPv8. In includes an android/kotlin implementation as well as a python implementation. We then built a Euro to EuroToken exchange and transaction validator on top of the python implementation. On top of the kotlin implementation we built a wallet app that is fully capable of securely transferring EuroTokens between wallets, as well as exchange them with the EuroToken exchange.

4.1. Architecture

The architecture of the EuroToken system has two main components. The gateway and the wallet. The gateway is managed by a central trusted party and fulfills two main functions from the design. These are to asynchronously validate transactions made by users, as well as handling the exchange between EuroToken and Euros. As such it maintains a bank account as well as its own wallet. The wallets are operated by each user, and they are fully capable of transferring funds between each-other without having to interact with anyone in the euro system.

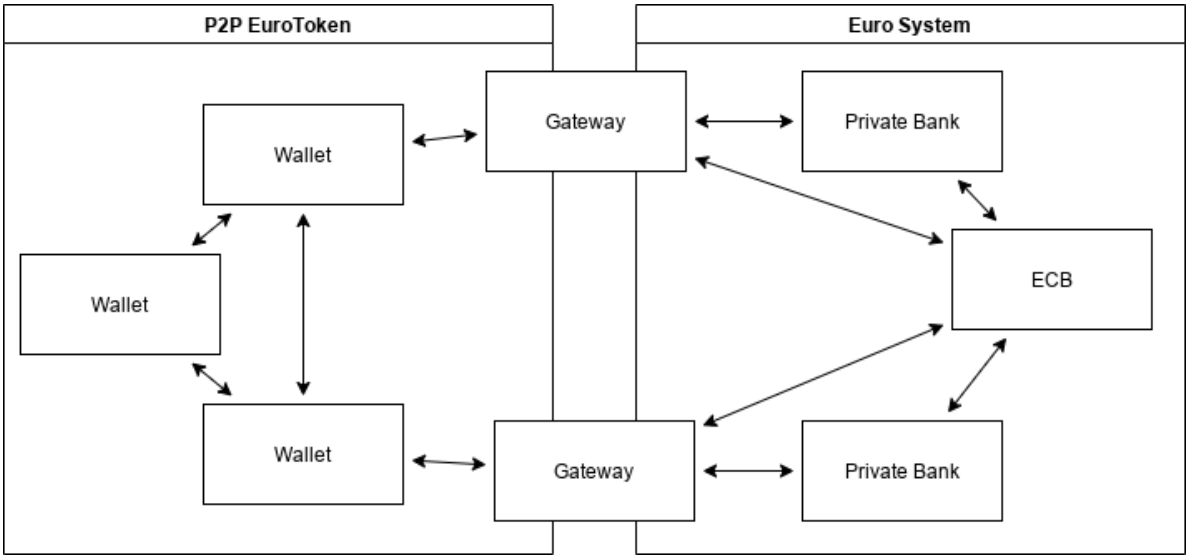


Figure 4.1: EuroToken architecture

In figure 4.1 we model the main communication channels. Within the P2P EuroToken system we have all wallet to wallet, and wallet to gateway communication. This communication happens directly between the communicating nodes using Peer-to-Peer technology.

In the Euro System, we make use of bank APIs for gateway to bank communication. The communication with the ECB symbolises monetary policy enacted by the ECB on bank EuroToken reserve

requirements or possible direct exchange of euro for EuroToken.

Rather than implementing both an exchange as well as a validator we chose to implement and test these as a single entity. However, since the gateways roles might be split in the future the technical implementation of the gateway keeps the validator roles separate from the exchange roles. This results in a single EuroToken exchange software product, that is able to perform either or both of the functions.

4.2. EuroToken transfer protocol

The method for accounting and transferring of EuroTokens lies at the heart of this project. Because of this the choices regarding the implementation of the networking stack and blockchain technology will have a direct effect on the feature set and scalability of the whole EuroToken network. We need a network stack that allows communication both off-line directly between devices, as well as online across the world. Finding and connecting to any wallet without relying on central servers is a main requirement. In addition, the off-line transfer ability of the system is best demonstrated by creating an android client. Another requirement is therefore that an implementation is available for android as well.

One option is to implement a full blockchain protocol and associated network stack from the ground up to adhere to our exact requirements. This would give us a lot of say in the exact feature set of the network. However, since the science of distributed networking algorithms has mostly settled, most peer to peer communication technologies have already been implemented somewhere.

The second option is then to build upon some existing peer to peer networking library, while implementing the blockchain protocol ourselves. This option has some benefits as the usage of a block-lattice is not yet very common, and thus is not implemented as a stand alone package anywhere. For the P2P library we have several options. We considered LibTorrent [18], Libp2p [17] and IPv8[PyIPv8:online]. LibTorrent has a number of interesting peer to peer features like peer discovery and data transfer but sadly fell short when it comes discovery of peers based on public keys. It can be classified more as a file location protocol than a peer location protocol. This would mean we would have to implement a peer location system ourselves. Libp2p is a modular peer to peer networking stack that provides a large suite of P2P tools. Libp2p uses a Distributed Hash Table (DHT) to allow peer discovery based on a peer-id [42]. There is an JVM/android implementation available, which also makes it possible to create an android client. Finally we looked at IPv8. IPv8 offers direct peer discovery based on public key and provides a framework for interaction called Overlay networks. Overlays provide a context for peers to interact within with particular message types. Crucially, IPv8 has an implementation in kotlin [36].

Rather than implementing the blockchain mechanism ourselves, there is a third option. IPv8 includes a module called TrustChain. TrustChain is in essence a block-lattice type distrusted ledger technology. The technology does not fully solve double spending they way we originally designed it, so some work is required to adapt TrustChain to the EuroToken system, but it would provide a good basis for our implementation.

We choose to build on IPv8/TrustChain for this project as it allows us to build on their kotlin implementation for the wallet as well as the python implementation for the gateway.

4.2.1. TrustChain structure

Every user runs a *Peer* which consists of a public/private key pair as well as a collection of their *transaction* history in the form of their *blockchain*. The Peer can be uniquely identified by their public key. Every statement made by the peer is signed using their private key, and the validity of any signature can be verified using the public key of the Peer.

Every peer has a list of their own history of transactions in the form of a collection of *blocks*. Every block is created and signed by a Peer, and includes the details of the transaction as well as a cryptographically secure hash of the previous block signed by the user. Importantly, the hash of a block uniquely identifies the block, as the trapdoor effect of cryptographically secure hashes ensures the infeasibility of finding another block with a given hash. The block thus uniquely references the previous transaction of the Peer. Since every transaction uniquely references the block before itself, the hash of any one block, recursively identifies *every* transaction made before by the Peer. This is as long as the Peer honestly references to their previous block. This referencing mechanism effectively links all blocks together in a gradually growing chain, thus making is a *blockchain*.

Every Peer in within TrustChain has their own chain, yet most transactions are *between* users. For

this reason all transactions are made to happen in the chains of both users involved. In TrustChain, this is achieved by having one of the two parties create a proposal block. In addition to the public key of the Peer, their previous hash, and the contents of the statement, the proposal also includes the public key of the counterparty. When the counterparty receives the proposal and agrees to the terms in the statement, they create an acceptance block. This acceptance block functions includes the public key of the counterparty, as well as the hash of their previous block, thus placing it in their blockchain. In addition the acceptance includes a reference to the proposal, thus linking them together. Both the proposal and acceptance blocks are then stored by both users, so they can both prove the transaction fully happened.

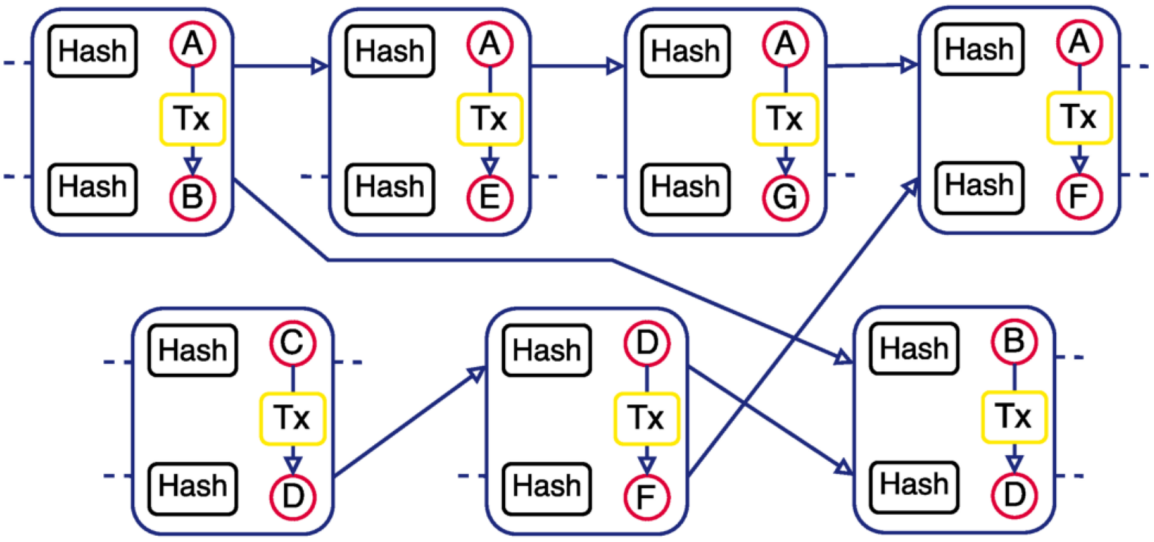


Figure 4.2: TrustChain block-lattice, interconnected personal blockchains[34] .

4.2.2. EuroToken extension

The structure inherited from TrustChain serves us quite well as it conforms quite well to the block-lattice design we require. However, neither the python, nor the kotlin implementation includes any logic for running a currency. Before TrustChain can be used for EuroToken, it needs to be expanded to allow for value tracking and transfer.

TrustChain is quite open for expansion. It allows users to define their own block-types as well as validation logic for these blocks. TrustChain will make sure blocks are valid as a chain, by enforcing typical block invariants like hash correctness and signature validity. TrustChain makes use of IPv8 for its communication and exposes an API to create and sign blocks to other peers. TrustChain will then handle the process of sending the blocks over the IPv8 network.

In order to create the EuroToken logic we defined a number of TrustChain block types to achieve our goals. In order to conform to the scalability requirements all EuroToken proposal blocks by a user will include the balance of that user. This is part of the rolling-checkpoint mechanic that allows us to scale each users personal blockchain indefinitely without sacrificing scalability. The EuroToken block subtypes are as follows:

Transfer block

The transfer block is the core of how users interact. The proposal is created by the sender of a transaction and the acceptance by the receiving party. The block includes the amount to be sent as well as the balance of the sender at that point. The receiver will verify that the balances of the sender are valid before creating the acceptance. The receiver will then calculate the spendable balance all the way back to the last “full” checkpoint block in order to validate whether the balance of the sender is even spendable.

Checkpoint block

In order to be able to spend the balance a user has received they need to proof that a validator has taken notice of the blocks of the senders. The checkpoint block serves as this proof of validator. The proposal is created by the user and the acceptance is created by the validator. A checkpoint block is only considered “full” if the both the proposal and acceptance exist. If the acceptance does not exist, the block is meaningless and any validation will keep recursing the chain until a full checkpoint is found.

Creation block

The creation is a special type of transfer that is done by a trusted exchange. This block is the only way in which new EuroToken are allowed to enter the system and will only be considered valid if it is made by a trusted party. The proposal is made by the exchange and the acceptance is made by a user.

Destruction block

The destruction is the opposite of a creation. The proposal is made by a user and acts as a transfer to an exchange. The exchange then also creates an acceptance block. The creation and destruction blocks are used to convert between Euro and EuroTokens.

4.3. Wallet

The core of the EuroToken network is the wallet. The wallet allows users to transfer funds to any other wallet anywhere on earth over the internet, or directly from device to device over Bluetooth. The wallet also has the capacity to exchange Euro for EuroToken and vice versa.

Instead of building a wallet from scratch we build on top of the TrustChain superapp [14] [48]. This app was developed to showcase the capabilities of the kotlin implementation of IPv8 [36]. The superapp in implemented as a collection of different subapps that use the same underlying IPv8 implementation. The app includes multiple other projects which we can integrate with the EuroToken system.

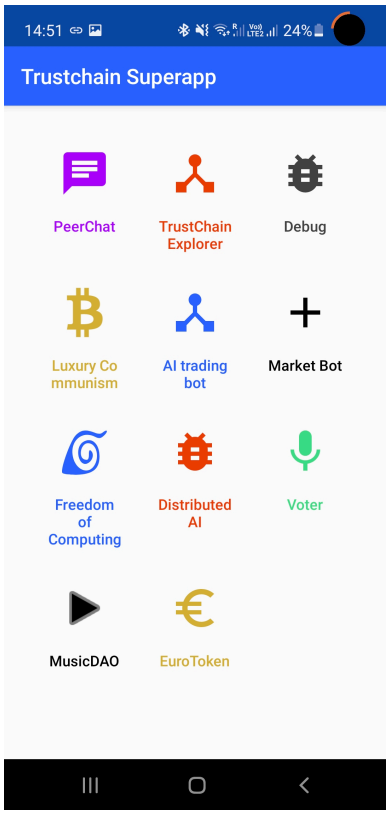


Figure 4.3: TrustChain Superapp [14]

4.3.1. Peer-to-Peer transfer

The main feature to showcase is the ability to transfer the EuroTokens. Before a user can send money to another user, they first need to know their public key. While sending money directly to a user is possible as part of the app, the share and transfer of public keys is not very practical. For this reason we implemented 2 ways of handling this. The first way is by generating a money request with QR code. This works best when the users are in the same room, or can share the QR code through some other means.

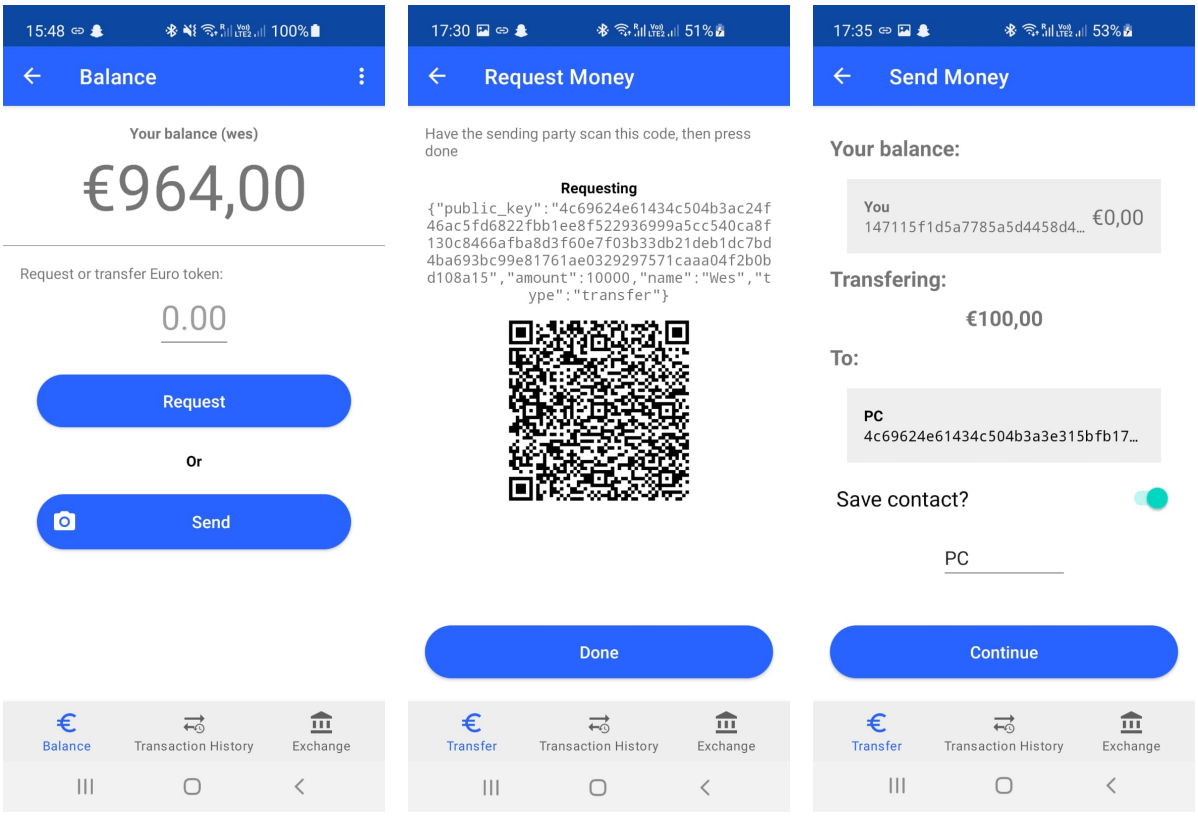


Figure 4.4: Wallet transfer by QR

A second and more user-friendly away to send money is through the already existing chat app PeerChat. PeerChat allows users to add each other as contacts and then uses IPv8 to send public key addressed messages. Instead of reinventing the wheel we added EuroToken payments to PeerChat. This mirrors payment apps like the Chinese WeChat Pay and the Norwegian Vipps. We believe this method of payment in the most natural for users.

Regardless of how users send money, their entire transaction history is available within the EuroToken sub app. Here all the different transaction types can be seen. The transaction screen also shows debug information like all checkpoints that have been performed with a validator. It also shows information about whether an acceptance block has been received from the counterparty. On this screen money can be payed back as well, and blocks van be resent in case of network failure.

4.4. Exchange

For the EuroToken to be part of the Euro system a mechanism of exchange is required. The exchange forms the bridge between the digital EuroToken and the rest of the Euro systems. The exchange mechanism must support the two main flows of value.

The first we call the creation flow. This flow handles the exchange of Euro for EuroToken, thus creating EuroToken. This involves the handling of payment into a bank account, verifying this, and paying out and equivalent amount of EuroToken to the users wallet. The second flow is the destruction flow. This flow does handles the opposite conversion. It handles a payment of EuroToken and pays out the equivalent amount to a IBAN bank account.

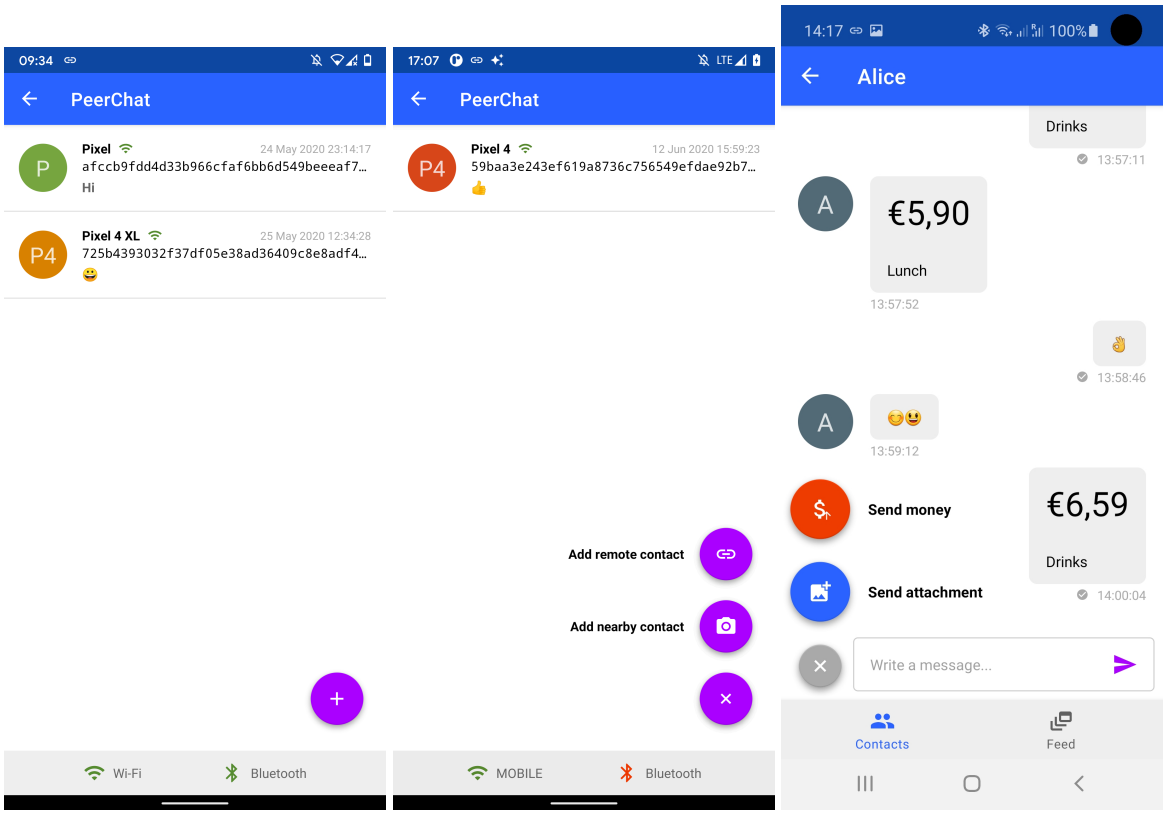


Figure 4.5: PeerChat, contacts [48] and pay via PeerChat

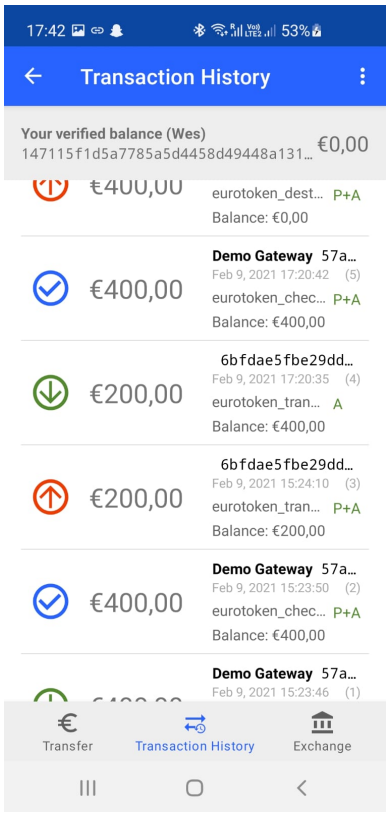


Figure 4.6: Wallet transactions

To handle this flow we build the exchange node. This node exposes a web frontend that allows the user to exchange their money in either direction. The exchange is implemented in python, and is based on the python implementation of IPv8 [11].

4.4.1. Buy and sell instantly

The frontend of the exchange is kept as simple as possible for demonstration purposes. Users do not need to login, and can buy or sell their EuroTokens directly on the front page.

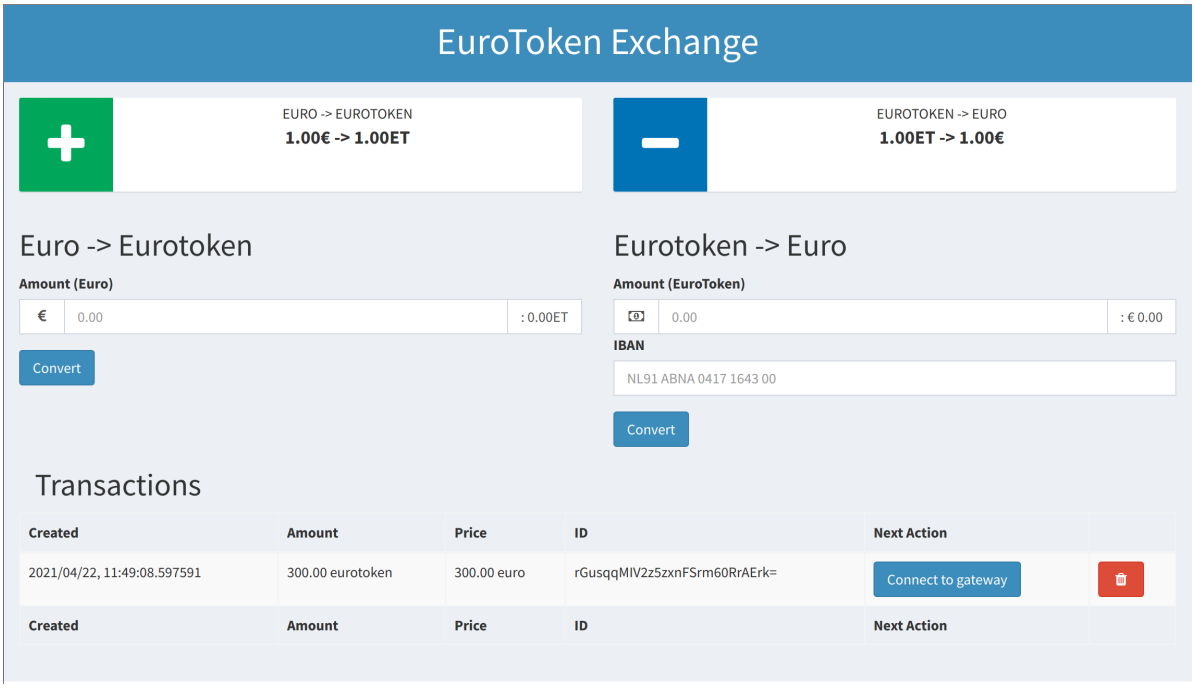


Figure 4.7: EuroToken Exchange Frontend

4.4.2. Exchange flow

The flow of exchange is different in each direction and require different steps from the user.

Creation

In our prototype we user the Tikkie API to enable users to pay us Euros. The creation flow can be seen in figure 4.8. The creation step is the most complex. This is because the sending of money to a user requires the exchange to know the public key of the user. In order to obtain EuroTokens the user accesses the web interface of the exchange, which will lead it through the following steps:

1. The user specifies the amount of EuroToken to buy. This creates a new transaction. The user then scans a QR code generated by the exchange using the wallet. The QR code contains the public key of the exchange, as well as a payment id. The wallet will then send a special connect message to the exchange over IPv8 with the payment id. When the exchange receives the message, the public key of the sender of the message is stored in association with the payment. This will be the public key to which the EuroToken will be transfered once the transaction is complete.
2. The exchange creates a new Tikkie payment request for the specified amount, which the user is then redirected to.
3. The exchange is alerted by Tikkie once the payment is complete.
4. The exchange will now send the money over IPv8.

Destruction

The destruction flow is a simpler process. If the user knows the public key of the exchange it can be performed completely in the wallet app. The user would simply send a destruction transaction to the

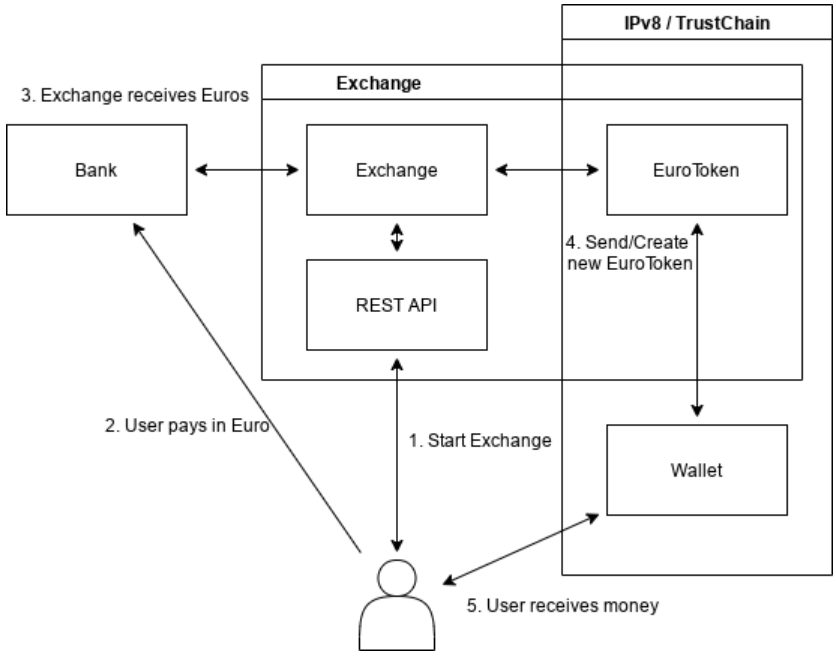


Figure 4.8: EuroToken Creation Flow

exchange which includes the IBAN the user would like the money to be payed out to as part of the block.

However if a user does not know the public key of the exchange, the interaction has to happen through the UI. This would involve the following flow:

1. The user specifies the amount to exchange along with their IBAN.
2. The exchange generates a QR code which includes their public key, as well as the amount.
3. The user scans the QR code and confirms the transaction in the app.

Within the app the exchange flows are handled using the pages shown in figure 4.9.

4.5. Validator

Together with the exchange, the validator is one of the special nodes that allow the EuroToken system to function. As can be seen in 4.6, a validation checkpoint is automatically requested after a transaction has been received by a user. The checkpoint makes the entire balance if the user “spendable”. The main task of the validator is to maintain the last blocks of all users in the network. This makes it impossible to double spend a transaction since any conflicting block has already been accepted by the validator. This makes the first block to arrive to the validator the one and only block at that position in a users chain.

Since the output of a transaction is only spendable when a full checkpoint comes after it, the wallet automatically performs a checkpoint after every transaction. This keeps the amount of blocks that have to be validated during every transaction as low as possible. This leads to every transaction involving only 4 half-blocks from the perspective of the sender. A sender only needs to share the transaction proposal itself, the block before (which is a checkpoint proposal), and the associated checkpoint acceptance. The receiver then only needs to verify the correctness of these 3 blocks and send back the acceptance to the sender. This preserves the transaction privacy of the both the sender and the receiver, revealing only the relevant transaction.

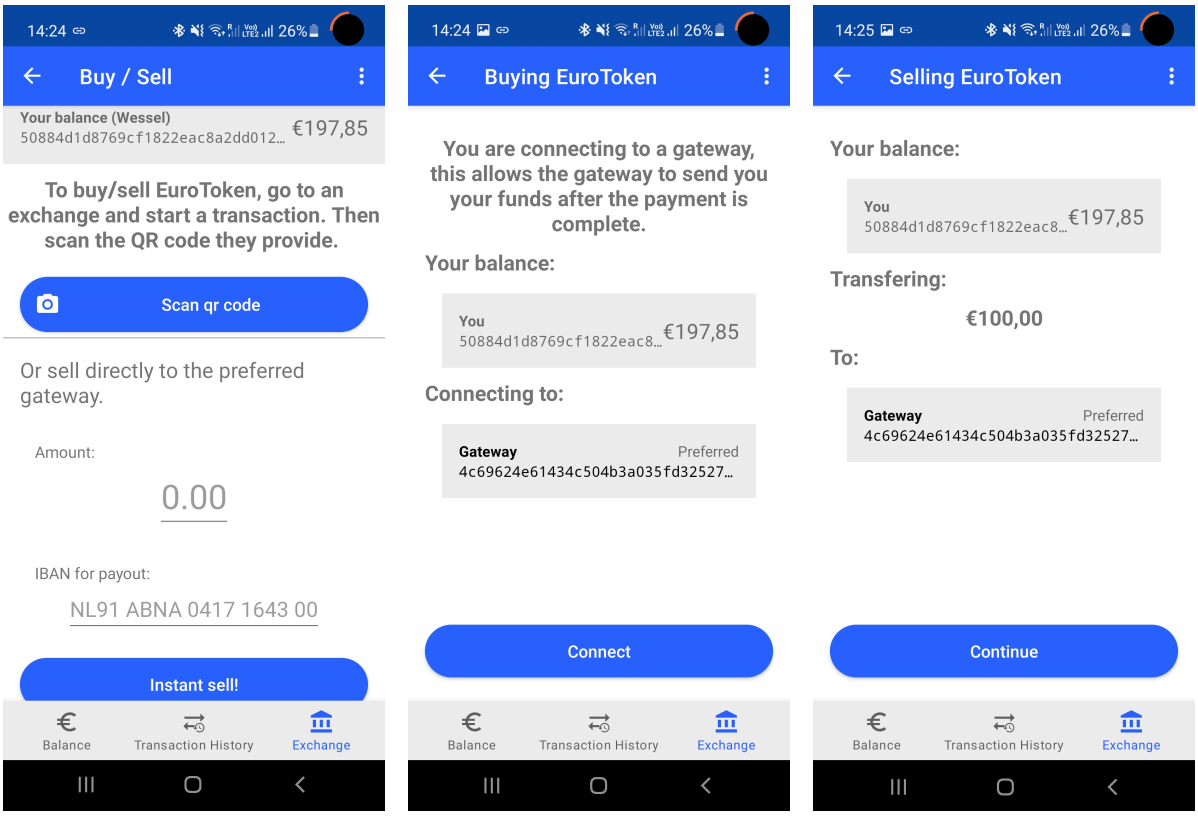


Figure 4.9: Wallet exchange

5

Evaluation

In the problem description we specified the following requirements as derived from the ECBs report on a digital euro [38].

1. **Enhanced digital efficiency**
2. **Cash-like features**
3. **Competitive features**
4. **Monetary policy option**
5. **Back-up system**
6. **International use**
7. Minimise ecological footprint (cost saving and environmentally friendly)
8. **Ability to control the amount of digital euro in circulation.**
9. Cooperation with market participants
10. Compliance with the regulatory framework
11. Safety and efficiency in the fulfilment of the Eurosystem's goals
12. Easy accessibility throughout the euro area
13. Conditional use by non-euro area residents

In this chapter we will evaluate our solution by these requirements. We emboldened the technical requirements as they will are guiding in our design, and we will go into more detail on how we met these requirements. The rest of the requirements will only be touched on lightly as they do not pertain to the topic of computer science and fall outside the area of expertise of the author.

We show how the EuroToken can be used to create: - a scalable CBDC - and provide all the benefits of programmable money - with the price stability of the euro.

5.1. Field trial

The purpose of the implementation in the super app was to showcase the usability of such a system. In order to test the implementation and the viability of the protocol in the real world, a field trail was conducted. We tested the EuroToken system during the morning hours, at café Doerak in Delft. As showcased in figure 5.1, the owner of the café generated a payment request for the amount of a single coffee and displayed it in the restaurant. Customers could then scan the code to transfer the money, and the owner who immediately see the money appear in their account.

This trail showcases the simplicity of taking digital payments without having to go through the process of registering with a traditional payment provider. Using the EuroToken system all the owner of Doerak needed was a smartphone in order to participate in the modern economy.

The ability of the EuroToken system to allow the easy participation in the economic system, without having to go through the gatekeepers of digital payments, positions it in a way to conform to the following requirements as set by the ECB:

- 1 **Enhanced digital efficiency**
- 3 **competitive features**

Figure 5.1: Field trial

5.2. Off-line trial

By building the EuroToken app on the TrustChain super app we could build on the bluetooth transfer features to implement the offline transfer of funds. In order to test this implementation and showcase the offline transfer capabilities of the EuroToken system, we conducted another trail away from civilisation. As showcased in Figure 5.5, in the mountains of norway, away from all network connectivity, we conducted a transfer of funds using the bluetooth connect feature of the superapp.

Figure 5.2: EuroToken off-line trial

There is room for improvement in the practicality of offline transfer of data between two devices. We found the process of creating a Peer-to-Peer bluetooth connection between two mobile devices somewhat cumbersome. And the system would greatly benefit in usability from proximity based data transfer via NFC.

Regardless of the possibilities for improvement, the trail successfully showed the viability of offline transfer. It shows the potential of the EuroToken system to act as a disaster proof payment system that remains functional at any distance from civilisation and during any disaster that would wipe out global communication infrastructure.

The user trades the risk of defering transaction validation until they connect to the network again for off-line transfers which allow for an instantatious transfer of funds, without requiring a connection to anyone in the rest of the network in order to perform the initial transfer.

Exploring the possibilities of reducing the transaction risk between initial transfer and transaction finalisation is an interesting topic for future research. Using reputation systems or digital identity solutions combinded with judicial accountability, the risk could potentially be reduced to near 0. - TODO work this out in more detail

In order to prevent double spending, the current implementation of the protocol disallows the re-spending of transactions that have not first been finalised. In order to provide a full disaster mode, re-spending funds without full transaction validation by the network is a must. This could be achieved by expanding the protocol to enable the settling of multi-hop transfers. Like the original off-line transfer system, this would require the 2nd receiver to accept the risk, that both of the peers that their transation depends on have double spent. When performing the initial off-line transaction, they would receive all the blocks neccesary to finalise all transactions before it with the gateways of the 2 peers before them. - TODO work this out in more detail

Going back to the requirements specified by the ECB, the offline transfer ability of the EuroToken system lays the groundwork for the following requirements:

- 2 **Cash-like features**
- 5 **back-up system**
- “Once over” spending
- Could be expanded to include “emergency mode” where trust is increased and reprocessing is performed later to find instances of double-spending

5.3. Controlled experiments

Besides real world tests we performed controlled experiments to explore whether the system has the properties we desire.

In these experiments we ran a number of wallets and had them transact randomly with eachother. We then logged all relevant data on the clients. We logged:

For the transactions we logged: - number of blocks validated by the client - validation time of the client - the users chain length at that point

For the checkpoints we logged: - number of blocks validated by the gateway - total validation time of the gateway - the users chain length at that point

We then varied: - The number of clients in the network - The number of gateways in the network - The frequency of checkpointing

5.4. Evaluating scalability

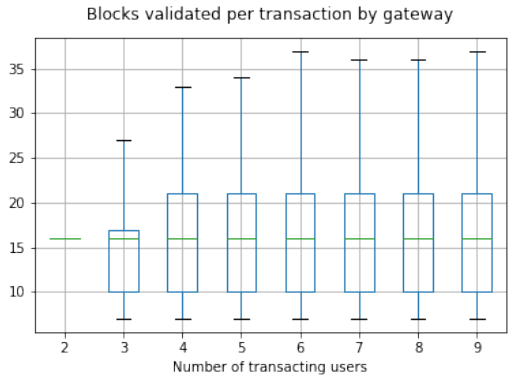


Figure 5.3: EuroToken off-line trial

5.5. Evaluating checkpointing

1 **Enhanced digital efficiency** 6 **international use**

In order to evaluate whether the EuroToken system can be deployed at a global scale while also

- Scalability is very important
- TPS of the gateway

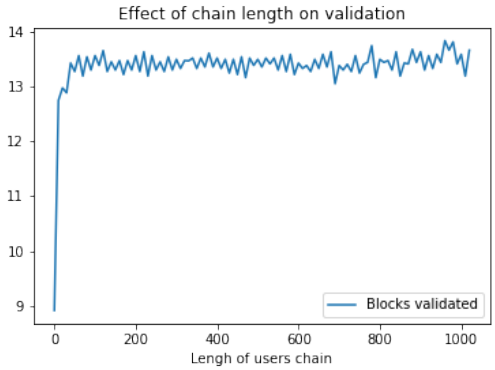


Figure 5.4: EuroToken off-line trial

- graphs and tables
- Scaling limits and how to potentially mitigate
- Description of the benefit of edge computing
- Instant international transfer
- Increased efficiency in regulation due to full standardization
- The innovation boost
- programmable money
- smart contracts
- new forms of money streaming

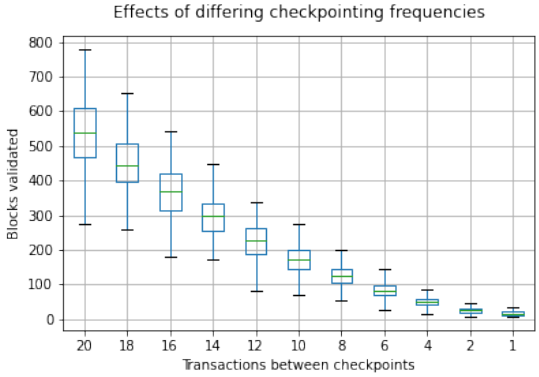


Figure 5.5: EuroToken off-line trial

2. Scalability Experiments In this chapter we evaluate to what degree the system conforms to the requirements 1 and 6 are met by the EuroToken system.

5.6. Evaluating security

Whether the system is actually secure against double spending is a difficult to prove in the real world. We showed how the system can be protected from exportation by the gateways. However this relies on the honesty of the gateways

5.7. ECB requirements

4 **monetary policy option** 8 **ability to control the amount of digital euro in circulation.**

- Central bank controlled supply
- Option for “global inflation rate”
- More granular and “smart contract based” policy enactment

5.8. Real world viability

Yes

5.9. Deployment consideration

Yes

6

Conclusion and future work

Yes

Bibliography

[1] Delft café premieres with eemcs blockchain euro. https://www.delta.tudelft.nl/article/delft-cafe-premieres-eemcs-blockchain-euro. (Accessed on 04/19/2021).

[2] Empsa - european mobile payment systems association. https://empsa.org/. (Accessed on 04/28/2021).

[3] Stablecoin on eos blockchain | eosdt. https://eosdt.com/en. (Accessed on 04/30/2021).

[4] E-commerce statistics for individuals. https://ec.europa.eu/eurostat/statistics-explained/pdfscache/46776.pdf. (Accessed on 04/12/2021).

[5] Ethereum 2.0 (eth2) vision | ethereum.org. https://ethereum.org/en/eth2/vision/. (Accessed on 04/28/2021).

[6] Statistics | eurostat. https://ec.europa.eu/eurostat/databrowser/view/ext_lt_maineu/default/table?lang=en, . (Accessed on 04/12/2021).

[7] International trade in goods. https://ec.europa.eu/eurostat/statistics-explained/index.php/International_trade_in_goods#Strong_increase_in_trade_in_goods_with_China_in_2010-2020, . (Accessed on 04/12/2021).

[8] Tangle_white_paper_v1.4.2.pdf. https://assets.ctfassets.net/r1dr6vzfxhev/4i3OM9JTleiE8M6Y04Ii28/d58bc5bb71cebe4adc18fadea1a79037/Tangle_White_Paper_v1.4.2.pdf. (Accessed on 04/12/2021).

[9] The maker protocol white paper | feb 2020. https://makerdao.com/en/whitepaper/. (Accessed on 04/30/2021).

[10] Nano_whitepaper_en.pdf. https://content.nano.org/whitepaper/Nano_Whitepaper_en.pdf. (Accessed on 04/12/2021).

[11] Tribler/py-ipv8: Python implementation of the ipv8 layer. https://github.com/Tribler/py-ipv8/. (Accessed on 03/23/2021).

[12] Online wallet for money transfers & online payments | skrill. https://www.skrill.com/en/. (Accessed on 04/28/2021).

[13] Stasis: Digital assets for intelligent investors. https://stasis.net/. (Accessed on 04/30/2021).

[14] Tribler/trustchain-superapp: Kotlin implementation of trustchain and ipv8 with rich networking: multihoming of local bluetooth+4g, decentral social networking, udp hole punching, etc. https://github.com/Tribler/trustchain-superapp. (Accessed on 03/23/2021).

[15] Introducing usd coin. https://www.circle.com/blog/introducing-usd-coin. (Accessed on 04/30/2021).

[16] Requiem for a bright idea. https://www.forbes.com/forbes/1999/1101/6411390a.html?sh=4e0608c6715f. (Accessed on 04/12/2021).

[17] libp2p. https://libp2p.io/, . (Accessed on 04/26/2021).

[18] libtorrent. https://www.libtorrent.org/, . (Accessed on 04/26/2021).

[19] *Introduction to PayPal*, pages 1–12. Apress, Berkeley, CA, 2007. ISBN 978-1-4302-0353-7. doi: 10.1007/978-1-4302-0353-7_1. URL https://doi.org/10.1007/978-1-4302-0353-7_1.

[20] Amelia Acker and Dhiraj Murthy. Venmo: Understanding mobile payments as social media. In *Proceedings of the 9th international conference on social media and society*, pages 5–12, 2018.

[21] ABN AMRO. Tikkie - bedrijven. https://www.tikkie.me/bedrijven. (Accessed on 04/27/2021).

[22] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J. Christopher Westland. Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3): 602–613, 2011. ISSN 0167-9236. doi: 10.1016/j.dss.2010.08.008. URL https://www.sciencedirect.com/science/article/pii/S0167923610001326.

[23] R.W. Blokzijl. rwblokzijl/stablecoin-exchange. https://github.com/rwblokzijl/stablecoin-exchange. (Accessed on 03/23/2021).

[24] Stefan A Brands. An efficient off-line electronic cash system based on the representation problem, 1993.

[25] Jetse Brouwer. Consensus-less security, 2020. URL http://resolver.tudelft.nl/uuid:d3d56dd8-60ee-47f7-b23a-cdc6c2650e14.

[26] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform. https://github.com/ethereum/wiki/wiki/White-Paper, 2013.

[27] D. Chaum. David chaum on electronic commerce how much do you trust big brother? *IEEE Internet Computing*, 1(6):8–16, 1997. doi: 10.1109/MIC.1997.643931.

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

[29] David Chaum, Christian Grothoff, and Thomas Moser. How to issue a central bank digital currency, 2021.

[30] CoinDesk. Bitcoin transactions are more expensive than ever - coindesk. https://www.coindesk.com/bitcoin-transaction-fees-more-expensive-than-ever. (Accessed on 04/28/2021).

[31] CoinDesk. Tether price | usdt price index and chart – coindesk 20. https://www.coindesk.com/price/tether, 03 2021. (Accessed on 03/19/2021).

[32] Richard N. Cooper, Rudiger Dornbusch, and Robert E. Hall. The gold standard: Historical facts and future prospects. *Brookings Papers on Economic Activity*, 1982(1):1–56, 1982. ISSN 00072303, 15334465. URL http://www.jstor.org/stable/2534316.

[33] Glyn Davies. *A history of money: from ancient times to the present day*. Cardiff: University of Wales Press, London, 2002.

[34] Martijn de Vos and Johan Pouwelse. Real-time money routing by trusting strangers with your funds. https://repository.tudelft.nl/islandora/object/uuid:c51ac99d-3013-44b3-8ddd-fbd951a2454a, 2018.

[35] Martijn de Vos, Can Umut Ileri, and Johan Pouwelse. Xchange: A blockchain-based mechanism for generic asset trading in resource-constrained environments, 2020.

[36] TU Delft. Tribler/kotlin-ipv8: P2p communication library for android. https://github.com/Tribler/kotlin-ipv8. (Accessed on 04/26/2021).

[37] Diem. White paper | diem association. https://www.diem.com/en-us/white-paper/, 04 2020. (Accessed on 03/19/2021).

[38] ECB European Central Bank. Report on a digital euro. https://www.ecb.europa.eu/pub/pdf/other/Report_on_a_digital_euro~4d7268b458.en.pdf, 10 2020. (Accessed on 03/19/2021).

[39] Forbes. Alibaba, tencent, five others to receive first chinese government cryptocurrency. https://www.forbes.com/sites/michaeldelcastillo/2019/08/27/alibaba-tencent-five-others-to-recieve-first-chinese-government-cryptocurrency/?sh=33d423fb1a51, 08 2019. (Accessed on 03/22/2021).

[40] Rachid Guerraoui, Petr Kuznetsov, Matteo Monti, Matej Pavlovic, Dragos-Adrian Seredinschi, and Yann Vonlanthen. Scalable byzantine reliable broadcast (extended version). 2019. doi: 10.4230/LIPIcs.DISC.2019.22.

[41] Charles M. Kahn and William Roberds. Credit and identity theft. *Journal of Monetary Economics*, 55(2):251–264, 2008. ISSN 0304-3932. doi: 10.1016/j.jmoneco.2007.08.001. URL https://www.sciencedirect.com/science/article/pii/S0304393207001250.

[42] LibP2P. Peer identity :: libp2p documentation. https://docs.libp2p.io/concepts/peer-id/. (Accessed on 04/26/2021).

[43] Tether International Limited. Tether whitepaper. https://tether.to/wp-content/uploads/2016/06/TetherWhitePaper.pdf, 06 2016. (Accessed on 03/19/2021).

[44] Karl Menger. On the Origin of Money. *The Economic Journal*, 2(6):239–255, 06 1892. ISSN 0013-0133. doi: 10.2307/2956146. URL https://doi.org/10.2307/2956146.

[45] Michael Ehrmann Miguel Ampudia. Financial inclusion: what's it worth? https://www.ecb.europa.eu/pub/pdf/scpwps/ecbwp1990.en.pdf, 01 2017. (Accessed on 03/23/2021).

[46] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2009. URL http://www.bitcoin.org/bitcoin.pdf.

[47] Reuters. China's $1.5 million digital currency giveaway impressed analysts. shoppers, not so much | reuters. https://www.reuters.com/article/china-currency-digital/chinas-1-5-mln-digital-currency-giveaway-impressed-analysts-shoppers-not-so-much-idUSL4N2H71NR?rpc=401&, 10 2020. (Accessed on 03/19/2021).

[48] Matouš Skála. Technology stack for decentralized mobile services | tu delft repositories. http://resolver.tudelft.nl/uuid:bd3a5fbd-430b-4af6-bc33-eab436f4f7db, 08 2020. (Accessed on 03/23/2021).

[49] The Maker Team. The maker protocol white paper | feb 2020. https://makerdao.com/en/whitepaper/, 02 2020. (Accessed on 03/19/2021).

[50] Fed The Federal Reserve. Preconditions for a general-purpose central bank digital currency. https://www.federalreserve.gov/econres/notes/feds-notes/preconditions-for-a-general-purpose-central-bank-digital-currency-20210224.htm, 02 2021. (Accessed on 03/19/2021).

[51] ECB Statistical Data Warehouse. Share of card payments in number of total payment transactions. https://sdw.ecb.europa.eu/quickview.do?SERIES_KEY=169.PSS.A.U2.F000.I1A.Z00Z.NP.X0.20.Z0Z.Z. (Accessed on 03/23/2021).