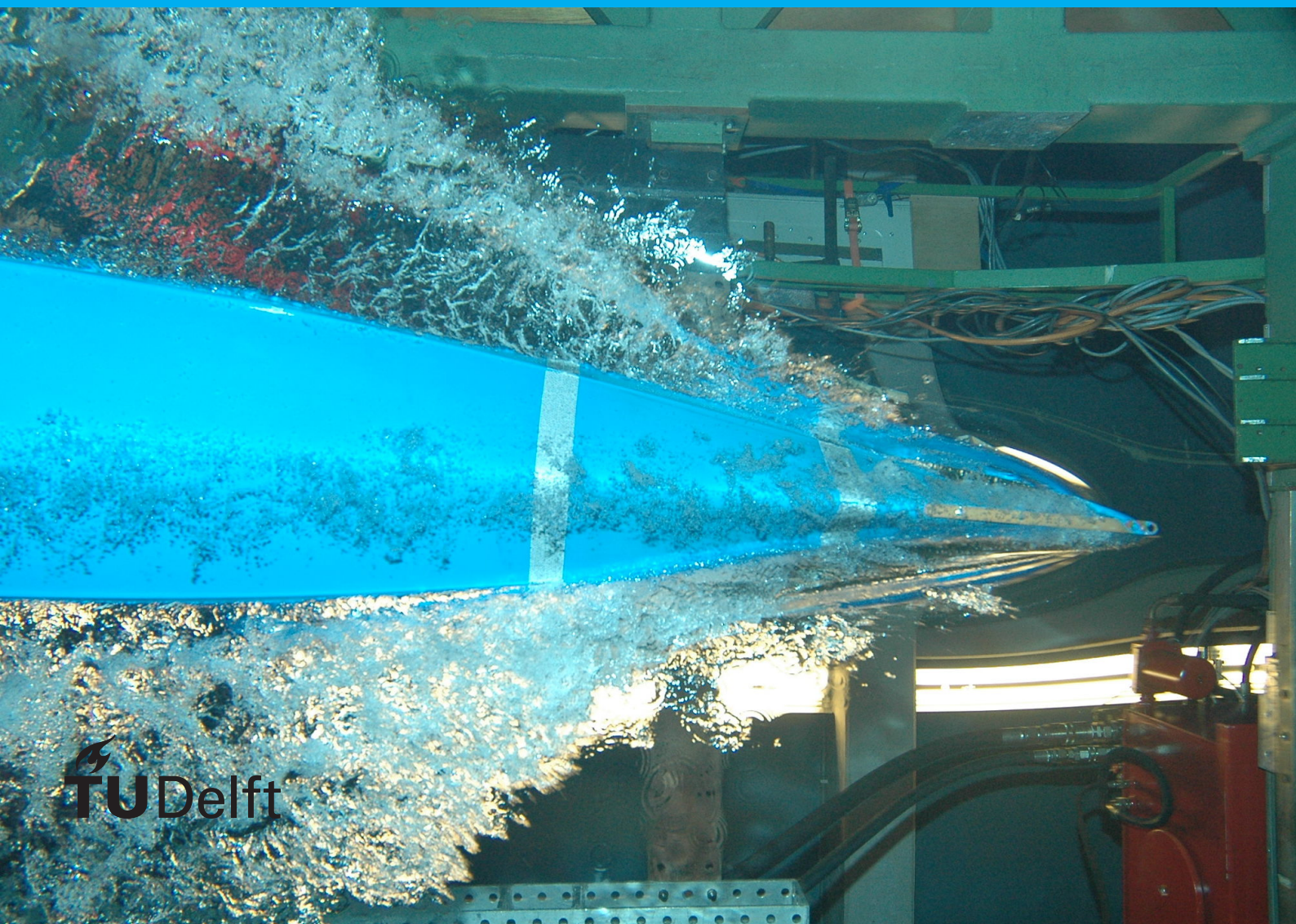
EuroToken

A digital extention of the Eurosystem with off-line transfer capapabilities

R. W. Blokzijl

- Stablecoin
- Blockchain
- Cryptocurrencies
- TrustChain
- CBDC

R.W.Blokzijl@student.tudelft.nl



EuroToken

A digital extention of the Eurosystem with off-line transfer capapabilities

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
	Member 2,	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 [TODO cite] 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 redistribute power towards the masses and away from large opaque organisations. 9 years later, Facebook announced a private currency controlled by a group of corporations [TODO cite]. 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) [TODO cite]. 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 determined by eccentric visionaries imagining a financial system that gave power back to the people and makes money open and programmable by anyone. Governments and large corporations are now competing to create the worlds main digital coin. The winner will be left controlling and overseeing all the worlds transactions, either for profit, or their national interests.

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, 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, the Eurozone is still deliberating while they might have a vital role in including the democratic process in the running of the future financial markets.

This thesis aims to provide a design for a digital euro that utilizes mechanisms used by todays stablecoin in order to create a digital euro analog called EuroToken. EuroToken is a design for a scalable system that allows transfer between parties in a scalable and peer to peer way, while maintaining price stability through maintaining collateral euros in a bank account.

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 all the benefits of programmable money with the price stability of the euro.

2

Problem description

2.1. The state of the financial system

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 [TODO cite]. While digital card payments are often more practical and safer than cash payments, the societal move towards digital money has some unintended side effects.

Digital card payments require an internet connection to succeed. While this is often available, it is not uncommon that a bank IT outage leads to payment capacities being unavailable, for a period of time.

When transfering money to an account with a different bank, especially across borders, the structure of the current system dictates a transfer time of days.

Money stored in a bank account is guaranteed by the bank. If the bank is to fail, the money in the account is usually insured up to 100.000 euro by various local government funds. When banks fail, the uninsured money is gone.

The move towards a bank dependent society will have the greatest impact on the worlds unbanked. In 2017, 3.6 percent of Europe's household had no registered bank account [TODO cite]. As more and more businesses become pin only. These people see their means of payment decrease.

Central bank money, or Cash, has already solved a lot of these issues. It's physical nature makes off-line payment trivial. Transfer is instant, and the value of the money is guaranteed by the Central Bank itself, with the note or coin itself being proof of its validity.

However, due to its appeal to robbers and thieves, and its physical limitations, the risks associated with digital money are accepted by many as the cost of doing business. As a result we have two half-solutions to the problem of wealth storage and transfer.

2.2. European Central Bank plans for a Digital Euro

For the reasons stated above combined with the erosion of its control over its currency, the ECB is looking into the creation of a digital euro [TODO cite]. While the EU is still in its exploration phase a number of scenarios and requirements have been worked out. Research is in progress as well, with a decision by the Eurosystem on whether to move forward a digital Euro planned mid-2021 [TODO cite].

The ECB sees a number of scenarios where a digital euro is desirable, and sometimes necessary, and from this need, a number of requirements are specified. The ECB

does not necessarily see only one digital euro, but is also open to a few different systems that work together to meet the needs and goals of the Eurozone.

First, the ever digitalising European economy may at some point benefit from the features of a digital currency with advanced features. As such a new euro would have to provide a feature set that would provide features that are at the technological frontier and form the basis for the economies of the future.

Second, a reduction of use of cash which threatens to leave the unbanked and vulnerable of Europe as well as those with a distrust in authority on the outside of the economy and therefore society. This opens the requirement of a currency with cash-like guarantees and features like off-line, private and permissionless transfer.

Third, a foreign or unregulated currency might become available that provides features such that the euro in its current incarnations becomes obsolete. This might be crypto-assets like Bitcoin, or a digital currency from other nation states who's interests are not necessarily aligned with that of the EU. As such a solution might be needed

Fourth, there may come a time where a digital euro and its features becomes desirable or necessary from a monetary policy perspective. In this case, a fully digital and standardized currency can provide a direct transmission channel for monetary policy like a global remuneration rate.

Fifth, if the current options for payment become unavailable or fall out of use because of large scale cyber incidents, natural disasters or a pandemic. A method of payment that operated over different channels than other options can provide a backup system for transactions.

Sixth, the euro takes a role in international markets. This would require a payment system that is convenient to residents outside of the euro area.

Seventh, the ecological footprint of the euro has to improve. A new currency could be organised differently and more efficiently. Simply by reducing the overhead of transactions, technically or organisationally would have a positive impact on its ecological footprint as well as reduce operational costs.

Other than these scenarios that would warrant a digital currency with additional features, the ECB has some general requirements that any digital euro have to adhear to.

First, the ECB requires the ability to control the total supply of digital euros in circulation. This is important to prevent the use of the currency as an investment vehicle, which would draw the money away from the banks and

- be careful with developing
  -
- report specifies general requirements and requirements for a nubmer of scenarios
  - `I should pick a scenarios and its requiremnts, and use it as the base motivation and problem for the thesis.`
- ECB requires a broad system that allows multiple digieuros to coexist
  - i do this by providing "general" e-euro management and tracking.
  - Verification could:
    - either be a service foy/by the ECB (CBDC)
    - or tracking of interbank IOUs to settle later among banks "regulated" by the ECB

With the looming threat of the Chinese CBDC, the

- lack of a european way of addressing current market concerns opens up Europe to foreign currencies to displace current local payment options -> strategic disadvantage
- What is a CBDC
  - a currency that has public money guarantees and stability, while having features and efficiency of digital money
  - Also needs to be available everywhere any time like physical money
  - privacy maintaining like physical money
- What does Europe want
  - `satisfy the emerging payment needs of a modern economy by offering, alongside cash, a safe digital asset with advanced functionalities.`
  - Strategic autonomy
  - Protection of citicens (the unbanked and vulnerable) when use of cash declines
    - Requires open access
- What are Europe's requirements?
  - robustness
  - safety
  - efficiency
  - protection of privacy
  - complying with relevant legislation
    - on money laundering
    - on financing of terrorism.

2.3. The state of Distributed Ledger Technologies

2.4. Fitting stablecoin technologies to the Euro

2.5. random text:

Once crypto currencies had been adopted by a significant amount of people and started to be used for real transfers of value, the speculators came.

Cryptocurrencies like Bitcoin were rapidly growing in popularity and therefore price. The interest in a functional distributed currency, combined with the prospect of riding the rise of this new store of value, pulled in enough people to reach a market capitalization of 328 billion USD in late 2017. With the crypto bubble of December 2017 and subsequent crash in early 2018, the significance of these concepts had been demonstrated to the entire world.

The popularity of decentralized crypto currencies is undeniable, yet they have failed to be gain widespread adoption in everyday use. Many crypto currencies are plagued by a lack of scalability, a lack of price stability as well as other issues. While these problems may well be solve in the future, government organisations are looking into CBDCs now.

When the facebook connected Libra was introduced in 2017. Many big payment vendors like Visa and MasterCard were members of the Libra Association.

The Central
This sparked a debate on the shape of the future of money. In order
Completely decentralized crypto currencies like Bitcoin have always been plagued by the same issue.
or at least failed to die,
the future impact of would
impact that this new form of money would h

- Physical money is slow - restricted by the speed of travel
- Digital money is less slow - restricted by the speed of human communication
- New digital money - restricted by the speed of light (and the runtime of cryptographic algorithms)

2.6. problem description

The euro zone is missing an option for a digital currency that mirrors all features of the euro, while providing the benefits of distributed accounting and programmable money.

In the historic paper “On the Origin of Money” (**???**) Karl Menger describes how people settle on a currency as a method of exchange. He describes that the willingness of people to exchange their goods for a commodity depends upon:

1. The ability to trade the currency for goods and services
2. The scarcity of the commodity
3. The uniformity, divisibility, durability and practicality of the commodity.
4. The development of the market, and how others speculate.
5. The limitations imposed politically and socially upon exchange, consumption and transfer from one period of time to another

All these aspects must be managed in any successful currency. In this chapter we will work out these requirements into a concrete set of requirements for the EuroToken system.

2.7. The general requirements for money

Points 1 and 4, the future usefulness of the currency and it’s market demand, are where digital currencies still fall short of traditional currencies. Because of the price volatility there is no way to know whether the coin you have today can still be used to buy the same amount tomorrow.

Anything that aims to replace the euro needs to be as price stable as the euro. Adding guarantees about the price, will make merchants more willing to accept the currency, thus providing the ability to trade the currency for goods.

The EuroToken system needs to satisfy all 5 requirements in order to be a viable currency. However, points 2 and 5 are dependent on the real world implementation, legal guarantees and political backing. And are thus out of scope for this project.

Point 3 is where crypto-currencies add their value, through their digital and distributed natures.

2.8. Requirements for digital assets

The usefulness and viability of a currency is still dependent on its functional aspects. With traditional currencies the issues of uniformity, divisibility, durability and practicality have long been solved. However in digital currencies these aspects bring with them many sub-requirements. In “On the Origin of Money” (**???**) Menger uses the concepts of spacial and “time” limits.

Space limits - The space limits of a currency is how costly a currency is to store, transport, and manage across multiple ‘market places’. Because of the digital nature of crypto currencies, the price of storing any amount of money is the price of storing some data. However, the transport and transfer (practicality) of the currency is dependent on having access to the data and equipment needed to do the transfer. For many digital currencies an internet connection is also required to facilitate a transfer. Additionally any network required for verification needs to have the capacity to transfer the currency for a sufficiently low cost.

Time limits - For digital currencies the time limits of a currency brings with it more complexity than physical money. While A users guarantee that their money is durable and will not be lost becomes a problem of IT systems, backups and cyber-security. As such any protocol and edge implementations need to be secure / securable in order for the currency to be properly durable.

Finally there is one more requirement of money that needs to be maintained in a digital system. That is the uniformity of money, also known as the fungibility of money. This is the concept that any 2 individual units of the currency have to be essentially interchangeable. This means that there cannot be any difference in value or risk based on the source of money. When building on a trust based, by default, the risk attached to any money received is dependent on the trust you have in the sender of the money. As such there needs to be a mechanism that reduces the transaction risk to a negligible level.

2.9. Political requirements for a future of money

The adoption of any new currency system across the euro zone will be dependent on many more factors than just the technical design. Even if the EuroToken system meets all the requirements specified in the last chapters, trust in the system will depend on “The limitations imposed politically and socially upon exchange”.

It is impossible, however tempting, to make any statements about how the system should handle a number of political issues. Instead a some trade-offs will be highlighted and discussed. Any value judgements will be reserved and left out of scope. The political considerations for any real world implementation of the EuroToken system, might include, but are not limited to:

1. the openness of its access vs the prevention of malicious activity
2. the privacy of its users vs the ability of the state to track malicious behaviour
3. the economic tools provided to the central bank vs the natural price development of the market

In the following sections each of these trade-offs will be discussed while specifying some requirements for different positions on the trade-off spectrum.

2.9.1. Openness vs Control

- The openness of its access vs the prevention of malicious activity

Case for openness
Case for control
Requirements:
Openness - No central servers for the core functioning of the network. (transfer without saction)
Control - Central rules that determine the validity of funds based on the source of the transaction. (identity should be verifiable)

2.9.2. Privacy vs Policability

- The privacy of the users vs the ability of the state to track malicious behaviour

Case for privacy
Case for security
Requirements:
Privacy - Future support for encrypting transactions while allowing for ZK-proofs for verifying availability of funds.
Policability - the ability to see information about a transaction regardless of the encryption
Finding a balance in this is not a trivial task.

2.9.3. Central control vs self regulation

- The economic tools provided to the central bank vs the natural price development of the market

Case for central control based on current politics.
Case for free market, new democratic backbone.
Central control - Allow the minting of new coins by the central bank, akin to QE
Self regulation - Build in features that guarantee market invariants, like a set inflation based on trackers. Or minting as a joint decision of many parties, possibly DAO style. Possibly a democratic system in the system, based on identity, separate from nation politics. Possibly deferring voting to financial institutions.

2.10. Summary

Deriving from the fundamental requirements of money the following requirements for the EuroToken system have been determined:
The system must:

- Allow the user to transfer funds to other users
- Allow the user to store funds for a period of time
- Have a mechanism to maintain the value of 1 token at 1 euro
- Maintain the uniformity/fungibility of the token
- Be theoretically scalable to billions of users
- Be completely digital and automated
- Be secure against double-spend attacks

The system should be expandable to allow:

- An integrated identity mechanism that verifies users and makes them accountable
- Mechanisms to hide the details of any transaction
- Mechanisms to bind network properties and rules to a democratic mechanism

System invariants:

- The market equivalence invariant - The amount of euro + EuroToken on the market must never be changed because of a system transaction. (the EuroToken shouldn't change euro supply)

2.11. Research focus and stucture

Can we design a digital financial platform that is:

+ Stable / Dependable
+ Peer to Peer (minimally dependent on centralised parties)
+ Globally scapable
+ Transferable off-line
+ Accountable

3

Design

3.1. TrustChain: Distributed accounting and networking

The possibilities and limitations of any virtual currency are dependent on its system of accounting. In order to facilitate the off-line and transparency requirements a system of distributed accounting is chosen.

As its fundamental building block the EuroToken system uses a Hyper-Sharded personal blockchain that keeps track of every users transaction history on their own edge device. This opens up a possibility of off-line transaction between users, without any link to the outside world.

As illustrated in figure 3.1:, every users personal blockchain is structured as a chronological, one-dimentional string of “blocks”. Every block will include the complete has identifying what block preceded it. Because of the trapdoor effect of the hash, this has the effect that any block will uniquely identify all blocks that come before it. This allows any user to verify the entire history of another user, given the existence and validity of the last block in this history.

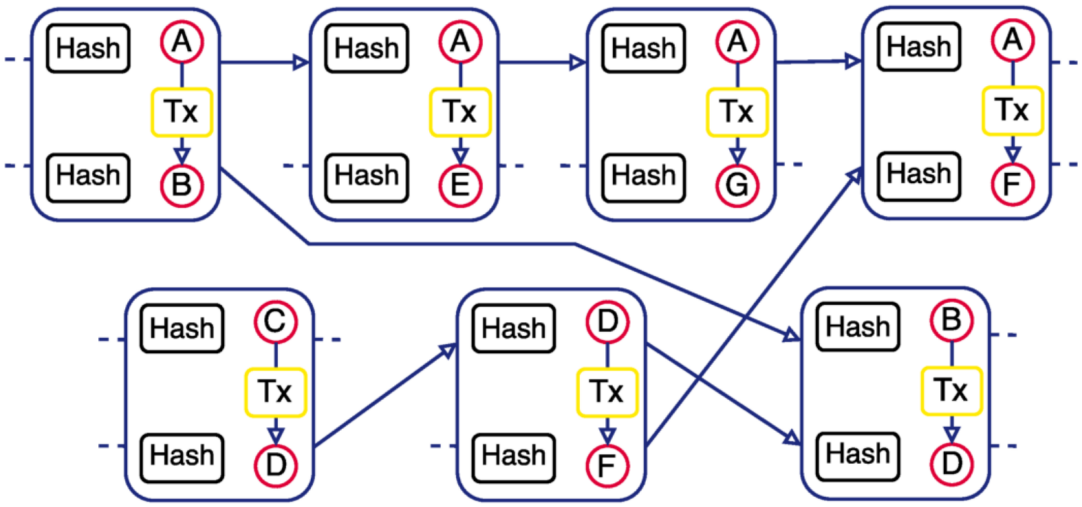


Figure 3.1: TrustChain (Vos and Pouwelse 2018)

Every block can contain 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 block with a declaration stating “I transfer 1 eurotoken to Bob”. In order to prevent fraudulent transactions every transaction is registered both Alice’s chain as well as Bob’s.

When bob receives the 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 chain 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, that he can use to prove the transaction as well as the chain of Alice at any point in the future.

3.2. Gateways: Euro to EuroToken exchange

The viability of any currency as a store of value over a time frame is dependent on its stability 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 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, and the euro as a digital, privately managed, unit of account. 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 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.

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 model, the gateways are designed to be run by public parties connected to the central bank. But we envision a possible future where multiple gateways are run by existing private money institutions who perform the heavy lifting of day to day transactions. In this system private banks act as an accounting system for the EuroToken without leveraging their EuroToken position. The Central bank would still have the option to “limit cash reserves” of these institutions to disconnect the impact of a failing euro on the EuroToken in the same way as the value of phisical public money is currently insulated from such failing.

This system allows for a smooth transition to a more EuroToken based euro system, while the established and regulated financial institutions are positioned properly in a place where financial services can be provided and a transition is smooth and beneficial to all parties.

3.3. 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.

3.3.1. 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 un-scalable. 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 distributed blockchains, 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

(**???**) 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 (Guerraoui et al. 2019) 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.

3.3.2. 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.

3.3.3. 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.

Bob's validator will sign the finality proposal iff, all "receiving transactions" from wallets associated with this validator (including Bob's) are valid.

Since the validation is performed by Alice, yet Bob's chain has to be verified, during the transaction, Alice requests all information that is required to validate Bob's chain. Alice will deliver this information to the validator with the finality proposal.

Bob's validator will verify that there are no other transactions that conflict with the one from Alice. And 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 form from Alice becomes the only accepted fork, and the other one is rejected.

In the case that a different fork from Bob has arrived at the validator first, the fork where Alice receives money is rejected. In this case, since Alice has already accepted the transaction in her chain and may have built other transactions after it, she could be requested to roll-back this block before the validator will accept it. Since Alice is not permitted to spend the funds from Bob until it has been finalised this is the point where double spending is prevented.

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

Once the verifier has verified and finalised a number of transactions, their received amounts can be included in the spendable balance of Alice.

3.3.4. Verification

For a block to be considered valid:

1. All standard TrustChain 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

3.3.5. 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.

3.3.6. 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.

3.4. 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.

3.5. Deferred validation, conflict resolution and off-line transactions

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

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

3.5.1. On-line 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.

3.5.2. 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 the double spend. The validator would then sign a transaction with the receiver to invalidate the funds

from the sender, and transfer them from the validator. The validator will then pursue legal action against the sender for fraud.

In the meantime the validator could not allowing 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 candidate for future research.

3.6. Regulation of validators

4

Implementation

The implementation of the stablecoin system consists of 2 main elements: the wallet Android app, and the gateway REST API. A web front end for the rest API has also been created.

The wallet demonstrates the ability of TrustChain to handle the transfer of the EuroTokens peer to peer without a central entity.

The Gateway demonstrates how a bridge can be created between the traditional euro system and a blockchain based analog.

4.1. Gateway (Central Bank API)

The only way tokens are created is when a central bank creates them. In our implementation this only happens when a user has transfered an equal amount of euro into the central bank account.

The gateway is responsible for the exchange of euro for tokens and vice versa. This involves taking payments in both tokens and euros, and payments in both currencies. This means the gateway needs to interface with the bank to allow a user to make payments in euro when creating EuroTokens, as well as a mechanism for paying out euro to the user when they trade in EuroTokens. On the other side of the gate the system needs to be able to create/send, and destroy/receive tokens on TrustChain.

The gateway aims to automate and link all of this interaction, so EuroTokens can be bought and sold at any time by anyone.

4.1.1. EuroToken Creation

When a user wants to convert a euro to a EuroToken, a creation event is initiated with the gateway API. The user sends their TrustChain wallet address and amount to convert with the request.

The API will then create a payment request with the associated bank for the specified amount, and store the information in its database. The payment link is returned to the user.

When the user has paid the request, a transaction for the EuroTokens will be created using TrustChain. The gateway will create a proposal half-block which will be sent to the user, who will create an accepting half-block registering the transaction on both chains.

The user is now free to send the EuroTokens to anyone they like, requiring only a TrustChain transaction.

4.1.2. EuroToken Destruction

When a user wants to trade in a EuroToken for a euro the process happens in reverse. For the demo the user does a request to the API with the desired amount, their TrustChain address and an IBAN.

The system creates a TrustChain transaction for negative the amount. This transaction is sent for the user to accept.

When the user has then signed the accepting half-block. The system will pay out the amount to the specified IBAN.

4.1.3. Frontend

To aid everyday users in the purchase and sale of EuroTokens a web frontend is created where the user can interact with the API. It demonstrates the ease of use of the system.

[Screenshots]

4.1.4. Implementation considerations

The design specified a general architecture for the EuroToken system. However in order to make an implementation possible within the constraints of the project some implementation trade-offs have been made.

Bank support

The EuroToken is designed to work with any bank account for euro collateral. However in this implementation we only implemented the API for ABN AMRO. Adding other banks is a simple as implementing the `Bank` class.

Euro Payment Initiation

The design specifies a requirement of automatic euro payout on EuroToken destruction. In order to automate this, most banks (including ABN) requires registration and use of the PSD2 payment initiation API. This API requires a Payment Initiation Service Provider (PISP) licence, which in turn requires a banking licence. Since both of these licences require you to be a fully functioning bank, the payment initiation part of the ABN API has not been implemented and is done manually in the field trial.

TrustChain

Since the main implementation if the TrustChain software (Tribler, n.d.) is build on python so is the gateway API. The server is provided as a single docker container that also provides the frontend.

4.2. Android Wallet

In order to use the EuroToken system on a daily basis, users need a way to send and receive the token. Because the added value of the system is its distributed nature, a way to send and receive the asset in a convenient and peer to peer way is needed. The TrustChain team has recently come out with an

Android super-app[TODO, cite] that showcases some of the IPv8 (Tribler, n.d.) and TrustChain[TODO CITE] capabilities. This app provides the perfect platform to showcase the EuroToken capabilities.

4.2.1. PeerChat Extension

The super-app already includes a number of applications, including PeerChat. A chat application that uses IPv8s peer to peer capabilities to communicate. In order to show

that the EuroToken can be used in a modern context, the PeerChat app has been expanded to include the capacity to send money attached to a message.

To send money, the user simply selects the option to send money, and is taken to a screen where a transaction can be created. The message is then sent to the receiver who within a few moments sees the transaction appear as a message in their shared chat. The transaction amount is also added to their balance.

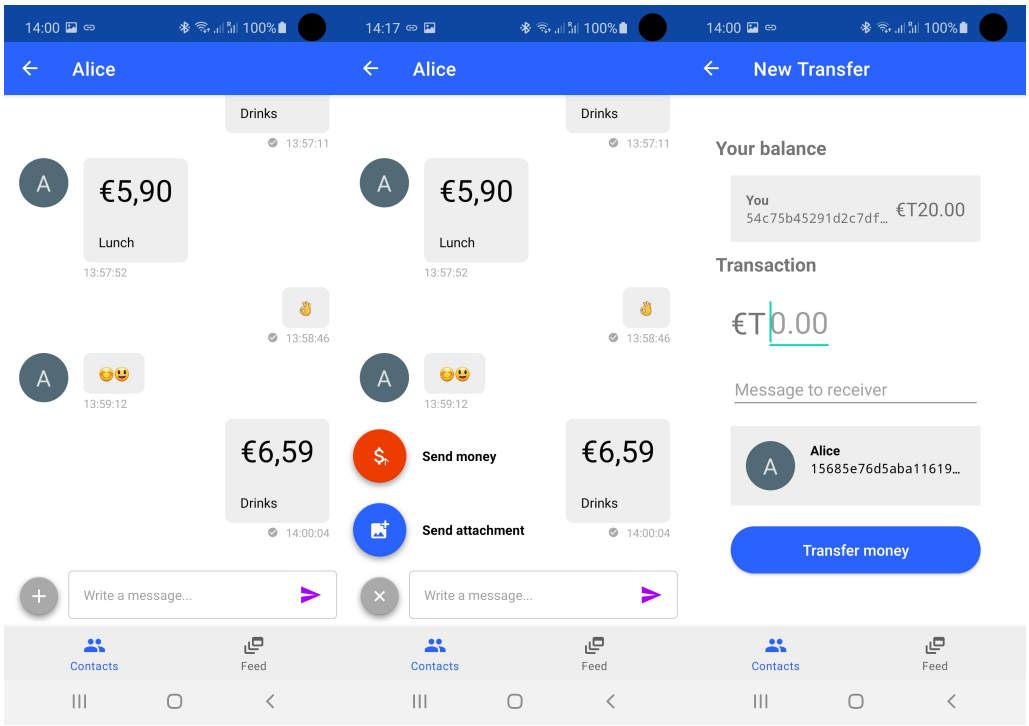


Figure 4.1: Attach money in PeerChat

The capacity to send transactions as shown in Figure 4.1 is not tied to PeerChat messaging. When money is sent, a transaction is created and transferred to the receiver using the TrustChain main community. The transaction hash is then sent as part of the PeerChat message. The receiver then fetches the transaction it received earlier via TrustChain.

- [TODO diagram of TrustChain and PeerChat interaction]

This implementation demonstrates the simple way in which EuroToken allows monetary transactions to be seamlessly and programmatically inserted into any application.

4.2.2. EuroToken app

The PeerChat app is one specific use case. In reality different applications would simultaneously use the EuroToken system. This would leave the user with a splintered record of their financial life.

In order to solve this, a EuroToken accounting app has been added to the super-app. The purpose of this is to show that the systems data can be reorganized in whatever way. The EuroToken app shows a history of all transactions, and provides another interface to the gateway.

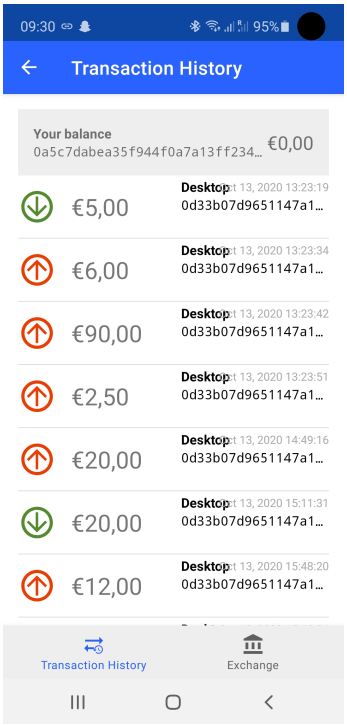


Figure 4.2: EuroToken Transaction History

4.2.3. EuroToken transactions in depth

- Validation (TODO)
  - Creation/destruction:
    - Trust Central Bank only
  - Transactions (prevent double spend)
    - Trusted users (based on identity later)
    - Validated by bank
    - Bami double-spend protection

4.2.4. EuroToken Settings

In addition to providing convenient services to the user, the EuroToken app has some configuration options to give the user control over their role in the EuroToken network.

Trusted Minters - Since the network has a central component that regulates the creation and destruction of the tokens, a demo requires a running server. Since there is no party to maintain such a server indefinitely right now, an option is added to allow the user to specify public keys of trusted “central banks”. If this option is enabled, the wallet in the super-app does only accepts blocks signed by the configured public keys.

- [TODO: image of minter config]

Trusted validators - Validation of transactions and prevention of double spending is unsolved in TrustChain but is an important part of any currency. Solving this problem in general is being worked on [TODO CITE bami] and is out of scope for this project. However the issue of transaction finality being important to a EuroToken system, a way to prevent double spending has been added. A transaction is not considered final until a trusted entity has signed a block in the senders chain that comes after the send block.

This means that the trusted entity is responsible for the validation of the block and its dependencies. These validators can be configured in the app in order to make de demo repeatable.

- [TODO: image of validator config]

5

Discussion

5.1. System dangers

5.1.1. Under-collateralization

Causes:

- By central bank printing without collateral
- Licenced gateway banks going bust, taking collateral with them

Effects:

Future bank runs could leave some token holders without their collateral, this makes token holders less confident in tokens. This would lower their value, but the direct exchange peg maintains the price. This hides the problem while undermining trust in the value of the tokens.

Solution:

- Don't print without collateral.
- Short term:
  - Keep collateral liquid at all times (also stops inflation)
- long term:
  - see system future

5.2. System future

- euros are deleted by banks on euro2token exchange, and created on token2euro exchange.
- Banks don't manange the collateral, only the CBDC exchange.
- Banks get a place in trust instead of investment.

6

Conclusion

Related Work

Guerraoui, Rachid, Petr Kuznetsov, Matteo Monti, Matej Pavlovic, Dragos-Adrian Seredinschi, and Yann Vonlanthen. 2019. “Scalable Byzantine Reliable Broadcast (Extended Version).” doi:10.4230/LIPIcs.DISC.2019.22.

Tribler. n.d. “Tribler/Py-Ipv8: Python Implementation of the Ipv8 Layer.” Accessed: June 13, 2020. https://github.com/Tribler/py-ipv8.

Vos, Martijn de, and Johan Pouwelse. 2018. “Real-Time Money Routing by Trusting Strangers with Your Funds.” https://repository.tudelft.nl/islandora/object/uuid:c51ac99d-3013-44b3-8ddd-fbd951a2454a.

Bibliography

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

[2] 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.

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

[4] 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.

[5] Tribler. Tribler/py-ipv8: Python implementation of the ipv8 layer. Accessed: June 13, 2020. URL https://github.com/Tribler/py-ipv8.