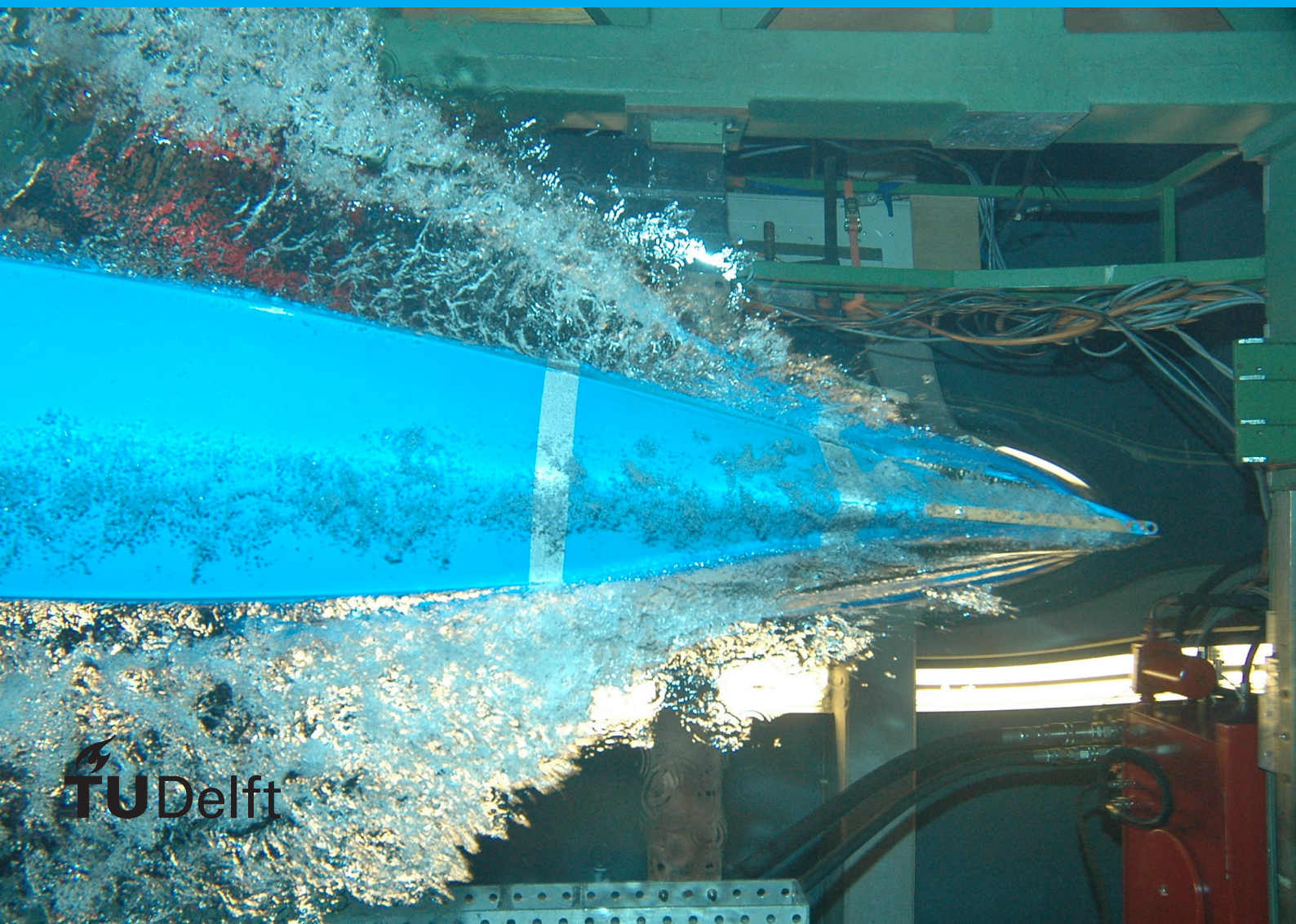
EuroToken
A Stable Digital Euro Based on TrustChain

R. W. Blokzijl

- Stablecoin
- Blockchain
- Cryptocurrencies
- TrustChain
- CBDC

R.W.Blokzijl@student.tudelft.nl



TU Delft

EuroToken

A Stable Digital Euro Based on TrustChain

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.

TU Delft

Preface

TODO: Add preface

R. W. Blokzijl
Delft, TODO

Contents

1

Introduction

- 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

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.

2.1. 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.

2.2. 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.

2.3. 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.

2.3.1. 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)

2.3.2. 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.

2.3.3. 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.

2.4. 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)

3

State of the art

Digital currencies

3.1. Money, its requirements and benefits

3.2. problems with money and how digital money solves them

3.3. Problems with digital money and how Bitcoin solved them

- mention ripple and why they aren't widely adopted yet

3.4. Problems with Bitcoin and how TrustChain is an alternative

3.5. Stablecoins

- Intro: Leftover discrepancies between traditional and digital money (stability)
- What is a stablecoin
- What makes a digital currency unstable, real question: what makes a normal currency stable
- What is a peg
- Other stablecoins in the wild
- Vision of the future of the euro zone

3.6. Terms used in this report

- Token
- Gateway
- Wallet
- CBDC - Central Bank Digital Currency

4

Design

TODO: Remove
Architecture Structure:

1. general ideas on keeping the currency stable
    a. Solve with Pegging ->
    b. Solve with Collatoral ->
    c. Solve with central bank ->
    d. Move to allowing central bank debt tracking
2. Global system architecture
    a. Money ingress and egress
        i. Trusted Gateway(s) to mint and burn coins.
            - Holding collateral
    b. Money transfer using trustchain
        i. Double spend protections
            - Checkpointing
3. On scalability
    a. Multiple gateways (Central Bank)
    b. A seperate set of trusted transaction validator nodes (Local banks)
4. Trustchain
    a. message types:
        i. Transfer
        ii. Creation
        iii. Destruction
        iv. Checkpoint
    b. User software
    c. Bank software
5. Scientific novelty
    a. Move the perspective from stablecoin to europe’s renewed infrastructure
    b. Other coins that aim to do this: ripple
    c. Our added value (to the current system and other solutions)
        - Users get
            * programmable money
            * Auditability by open source software
        - Banks get
            * One standard for banking ledgers
            * Added benefit of not losing central monetary policy handle.
            * Smooth long term migration to the new system with failsafe

TODO: Intro

4.1. Pegging to the euro (Design level 1)

As we have explored in chapter TODO, there are a few ways of stabilising a currency. Since the EuroToken system needs to function as a replacement of the current financial infrastructure and has to be stable with regard to the euro, we have chosen a direct peg to the euro. This requires a mechanism that allows any holder of the euro to exchange it for an equal amount of EuroTokens at all time.

Because of the system invariant to not change the total supply of euro + EuroToken, every addition of a EuroToken to the system should result in the removal of a euro, and vice versa. Since we cannot create and destroy euros at will, a collateral system will be used to ensure absolute availability of euros and EuroTokens at any time to facilitate both directions of exchange.

The collateral system will work as follows: A holder of euro will trade their euro for a EuroToken with a central bank certified gateway. The gateway will store the euro in a bank account, effectively removing it from the market. These gateways have the sole power of EuroToken minting and the only situation where EuroTokens will be minted is when a euro is provided as collateral.

Firstly, this maintains the market equivalence invariant as EuroTokens only enter the market when a euro is removed. And secondly the collateral equivalence invariant, that the amount of euro held in collateral is always equal to the amount of EuroToken on the market.

Because of this guarantee, the other direction of trade is always possible as well. When a EuroToken holder wants to exchange their token for a euro, it can always do so with a gateway. The gateway will remove the EuroToken from circulation while adding a euro back to the market. Thus again maintaining the market equivalence invariant as well as the collateral equivalence invariant.

4.2. System architecture

`TODO: intro 2 subsystems`

4.2.1. The blockchain

- Use of any distributed ledger is possible, but must adhere to
  - Must scale to planet scale with micro transactions
  - Must be immutable
    - To be accountable
  - Should have identity built in
  - Should allow for other features
  - Must include double spending protection

4.2.2. Considerations for TrustChain

- Transfer using any application
- Transfer scales infinitely
- Transfers Require verification by trusted node

4.2.3. Validators and judicial compatibility

- Verify balances and transactions to prevent fraud
- Expandable to verify other things (terrorism prevention iff identity)
- Scalability depends on network rules
  - When is a node trusted
  - What happens on validation failure (double spend)
    - Auditability through the blockchain
    - Accountability provided by Identity
    - Judicial system to settle disputes with signed proof of fraud

4.2.4. The gateway

- acts as a link between 2 assets
  - Could be expanded
- Is central bank certified

- Scalability might require backend communication
- Management of euros on backend is up to the central bank.
  - Individual banks might not have enough collateral for all trades.
  - What happens on bank failure
    - Exit scamming

4.3. Demo specific choices

4.3.1. What will this thesis implement

1. Single central bank
2. Central bank is also validator
3. Expansion on preexisting TrustChain super app to add EuroToken user capacities.

4.3.2. TrustChain message types

1. Transfer
2. Creation
3. Destruction
4. Checkpoint

4.4. Extra System considerations

4.4.1. Security

4.4.2. Scalability

4.4.3. Usability

4.4.4. Auditability

4.5. Scientific novelty

5

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.

5.1. 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.

5.1.1. 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.

5.1.2. 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.

5.1.3. 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]

5.1.4. 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.

5.2. 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.

5.2.1. 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.

The capacity to send transactions as shown in Figure 5.1 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.

5.2.2. 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.

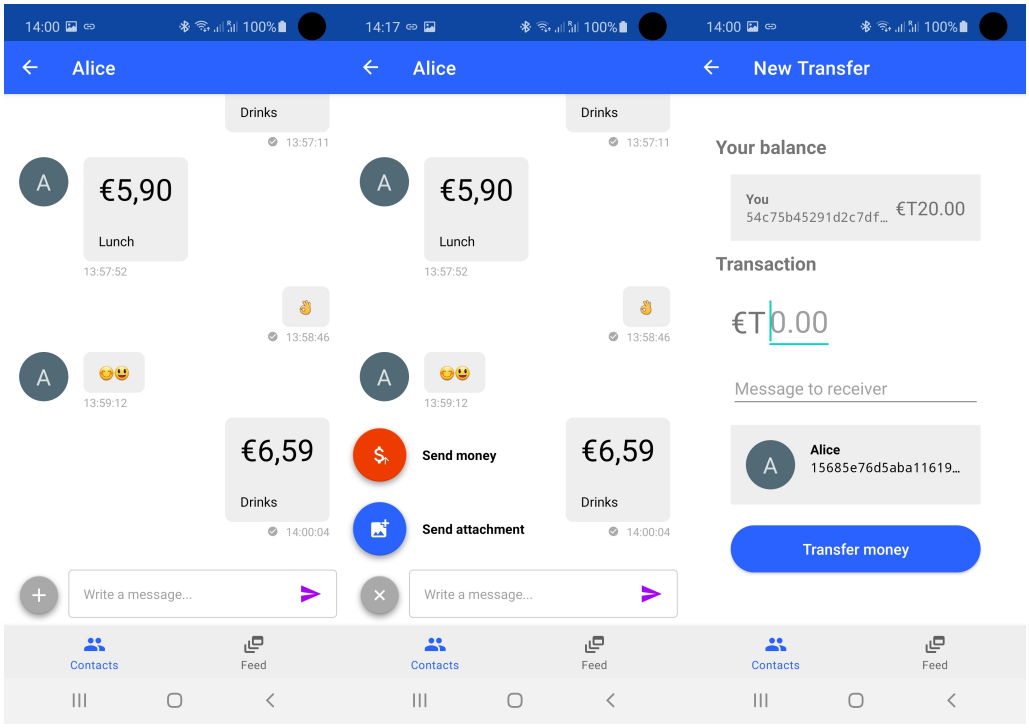


Figure 5.1: Attach money in PeerChat

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.

5.2.3. 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

5.2.4. 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

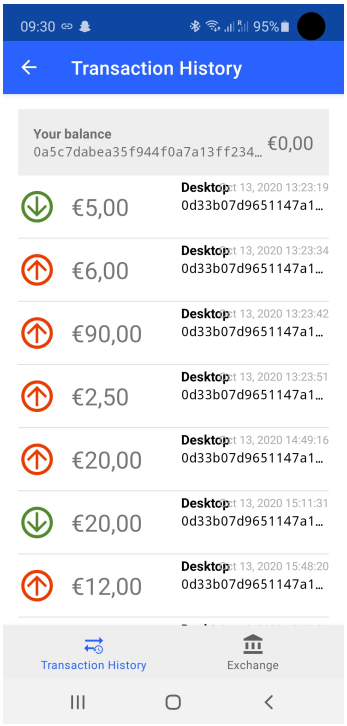


Figure 5.2: EuroToken Transaction History

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]

6

Field trial

7 Discussion

7.1. System dangers

7.1.1. 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

7.2. 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.

8

Conclusion

Related Work

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

Bibliography

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