

EuroToken

A Stable Digital Euro Based on TrustChain

R. W. Blokzijl

- Stablecoin
- Blockchain
- Cryptocurrencies
- TrustChain
- CBDC

R.W.Blokzijl@student.tudelft.nl

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 December 31, 2013.

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

Preface

TODO: Add preface

R. W. Blokzijl
Delft, TODO

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 2 | Problem description | 3 |
| 2.1 | Background | 3 |
| 2.2 | Stablecoin primer | 3 |
| 2.3 | Goals | 3 |
| 2.4 | Terms used | 3 |
| 3 | Design | 5 |
| 3.1 | Design requirements | 5 |
| 3.2 | System architecture | 5 |
| 3.3 | How does this solve the requirements. | 5 |
| 3.4 | Multiple perspectives | 5 |
| 3.5 | Theoretical expansion of the concepts | 5 |
| 3.6 | TrustChain as an accounting platform for financial transactions | 5 |
| 3.6.1 | Day to day money transfer in the 21 century | 5 |
| 3.7 | System considerations | 5 |
| 3.7.1 | Security | 5 |
| 3.7.2 | Scalability | 5 |
| 3.7.3 | Usability | 5 |
| 3.7.4 | Audibility | 5 |
| 4 | Implementation | 7 |
| 4.1 | Gateway (Central Bank API) | 7 |
| 4.1.1 | EuroToken Creation | 7 |
| 4.1.2 | EuroToken Destruction | 7 |
| 4.1.3 | Frontend | 8 |
| 4.1.4 | Implementation considerations | 8 |
| 4.2 | Android Wallet | 8 |
| 4.2.1 | Trustchain. | 8 |
| 5 | Field trial | 9 |
| 6 | Discussion | 11 |
| 6.1 | System dangers | 11 |
| 6.1.1 | Under-collateralization | 11 |
| 6.2 | System future | 11 |
| 7 | Conclusion | 13 |
| | Related Work | 15 |
| | Bibliography | 17 |

1

Introduction

(Tribler, n.d.)
Libra bad, CBDC better.

2

Problem description

2.1. Background

digital currencies

- Money, its requirements and benefits
- problems with money and how digital money solves them
- Problems with digital money and how bitcoin solved them
- Problems with bitcoin and how trustchain solved them
- Leftover discrepancies between traditional and digital money -> segue to next chapter

2.2. Stablecoin primer

- What makes a digital currency unstable, real question: what makes a normal currency stable
- What is a stablecoin
- how to peg a currency
- Other stablecoins in the wild
- Vision of the future of the eurozone

2.3. Goals

- Imagining a new accounting layer

2.4. Terms used

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

3

Design

3.1. Design requirements

3.2. System architecture

3.3. How does this solve the requirements

3.4. Multiple perspectives

- Stablecoin or tokenised euro?
- tokenised euro or standardised, decentralised bank ledger accounting?

3.5. Theoretical expansion of the concepts

- Multi bank design
- Identity integration
-

3.6. TrustChain as an accounting platform for financial transactions

3.6.1. Day to day money transfer in the 21 century

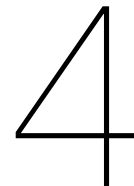
3.7. System considerations

3.7.1. Security

3.7.2. Scalability

3.7.3. Usability

3.7.4. Audibility



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 transferred 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/recieve 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 as 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 of the TrustChain software (Tribler, n.d.) is built 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

4.2.1. Trustchain

5

Field trial

6

Discussion

6.1. System dangers

6.1.1. 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

6.2. System future

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

7

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>.