POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Classifications, models and stability stress test for stablecoin

Tesi di Laurea Magistrale in
Mathematical Engineering - Ingegneria Matematica

Author: **Alessandro Toffalini**

Student ID: 953036
Advisor: Prof. Daniele Marazzina
Co-advisors: Dott.ssa Nadia Fabrizio
Academic Year: 2021-22

Abstract

Stablecoins are cryptographic tokens that live on a blockchain with the aim of maintaining a stable price with respect to an asset, that could be a fiat currency or a commodity i.e. gold. In this work we're going to investigate how it is possible to create a stablecoin and which mechanisms are available. In particular, we discuss some classification schemas for stablecoins. Among all possible classifications, we identify three main possibilities: tokenized fund, crypto collateralised stablecoin, and purely algorithmic stablecoin. Tokenized funds are mainly tokens emitted by a company that holds fiat currency as a reserve to maintain the value of the token. Crypto collateralised stablecoins are tokens managed by a smart contract that for every unit of stablecoin has some amount of cryptocurrency to maintain the value. Purely algorithmic stablecoins do not have a collateral to maintain the peg. They are managed by a smart contract that has the capabilities of creating the stablecoin and other tokens and buy or sell them directly on the decentralized exchanges (DEX) available as decentralized finance (DeFi) applications. In the last part, we provide the use case of Terra, an algorithmic central bank stablecoin. We describe how the Terra ecosystem works. Terra smart contract acts as a central bank and performs market operations on DEXes. Then a stress test is conducted after building a model where we simulate the market demand of Terra with a geometric brownian motion with a drift that changes between two states following a Markov chain. Another independent geometric brownian motion is used for modeling the price of Luna. The test uses two stress factors and monitors two risk parameters: increase of Luna supply and decrease in miner rewards. We conclude that the risk of de-peg comes from huge market volatility in both risk factors in contemporary.

Keywords: Blockchain, DeFi, stablecoin, cryptocurrency, stress test

Abstract in lingua italiana

Le stablecoin sono token crittografici basati sulla blockchain con lo scopo di mantenere un prezzo stabile rispetto ad un asset, questo può essere una valuta fiat oppure una commodity, i.e. l'oro. In questa tesi indagheremo com'è possibile creare una stablecoin e quali meccanismi sono disponibili. In particolare, mostreremo diverse possibili classificazioni per le stablecoin. Tra tutte le classificazioni ne identifichiamo tre: fondi tokenizzati, stablecoin collateralizzate tramite cryptovalue e stablecoin puramente algoritmiche. I primi sono token emessi da società che tengono riserve di valuta fiat come collaterale, le stablecoin collateralizzate tramite cryptovalue sono token controllati da smart contract, i quali per ogni token conservano un certo quantitativo di cryptovaluta che garantisce il valore. Nell'ultimo caso delle stablecoin puramente algoritmiche non abbiamo nessun collaterale per mantenere il valore. Sono controllate da smart contract con la capacità di creare stablecoin e altri token e tramite algoritmi decidere se vendere o comprare quest'ultimi direttamente su degli exchange decentralizzati (DEX) disponibili grazie alle applicazioni della finanza decentralizzata (DeFi). Nell'ultima parte, presentiamo Terra, una stablecoin algoritmica basata sul modello di una banca centrale. Descriviamo come funziona l'ecosistema di Terra nel dettaglio. Gli smart contract che gestiscono Terra funzionano come una banca centrale ed eseguono operazioni di mercato sui DEX. Dopo l'introduzione di un modello per descrivere l'ecosistema Terra verrà proposto uno stress test per valutarne la stabilità. La domanda di Terra viene simulata attraverso un moto browniano geometrico con un drift che cambia tra due stati seguendo una catena di Markov. Un altro moto browniano geometrico indipendente dal primo viene usato per modellare il prezzo di Luna. Questo test usa due fattori di stress e monitora due parametri di rischio: l'incremento nel numero di Luna in circolazione e la decrescita dei premi per i miner. Si conclude che il rischio di perdere la stabilità arriva da una grossa volatilità di mercato nei due fattori di rischio in contemporanea.

Parole chiave: Blockchain, DeFi, stablecoin, cryptovaluta, stress test

Contents

Introduction

This thesis is the result of a research work in the blockchain sector carried out during the internship at Cefriel, a company based in Milan which operates in the field of innovation and research. The objective of this internship and of this thesis is to study stablecoins, which is one of the emerging topics in the blockchain technology domain.
Stablecoins are exploding in popularity for several reasons: they allow to transfer value on a blockchain network without the risks connected with the high volatility of the traditional cryptocurrencies, but with the advantages of the blockchain technology, for example: immutability of the transactions, no central authority and transactions publicly verifiable. Stablecoin can also provide access to the decentralized financial system for people that don't have access to the traditional banking products, for example, there is the possibility to deposit stablecoin in DeFi (Decentralized Finance) application that can provide convenient interest rates.

We'll start with a background on the technologies that made the blockchain possible, and also some remarks on the different consensus algorithms. In particular, we will deal with some cryptographic concepts, consensus algorithms, smart contracts and token.
Then we'll get to the core providing a definition of what a stablecoin is. After that we'll go through why it is important to have the possibility to use stablecoin, and in particular we'll provide an example of Decentralized Finance.
We'll introduce several examples of stablecoins (Tether, USDC, Dai, Terra) and two different classifications, one provided by the European Central Bank (ECB) and the other one provided by a research team of the MIT. Each classification method sees the stablecoin from a different point of view, and provide a different vision over the stablecoin domain.
Then we will focus on a particular kind of stablecoin, the algorithmic one, and we'll go into detail in particular with three categories: full reserved crypto stablecoin, algorithmic central banks and seigniorage shares stablecoins. And for each category, we will provide an example of stablecoin with the corresponding stability mechanism.
Then we'll move to the more technical part: we'll describe in details an algorithmic stablecoin called Terra.

In particular, we'll describe the stability mechanism algorithm behind Terra and how the system auto-regulates the transaction fees and the burning rate, which are the two main levers to maintain the price stable.
The last chapter will be a stress test for the Terra ecosystem, we'll describe the model implemented in the original whitepaper of Terra. Then we'll start with developing our own model for all the variables involved in Terra stability (Luna supply, fees, etc.). Next we'll identify in which condition the Terra stablecoin is a risk of losing the peg with the currency of reference. After that we'll choose the parameters of the model to stress test. Afterwards we'll define a baseline scenario where we'll find some threshold values. Therefore, we run our simulations with several combinations of parameters and compute the probabilities of de-peg dividing the number of times the thresholds are broken by the total number of simulations. And in conclusion we're going to analyze the results of the stress test and compare it with the original one.

1 | Brief History of Cryptocurrencies

1.1. Birth and evolution: Blockchain, PoW, PoS

The aim of this chapter is to introduce the technological background of Blockchain, Smart Contracts and Cryptocurrencies related to the context of Stablecoins. In particular in the first part we will deal with the history of some technological concepts that will lay the foundations for the development of the Blockchain and its first implementation with Bitcoin. In the second, we are going to explore how two of the main used consensus algorithms work: the Proof of Work and the Proof of Stake. After that we will explain the concepts of Smart Contracts and tokens.

1.1.1. 1990-2008

This chapter will start with an historical overview of the technologies that made the Blockchain possible. The first time that the main blockchain concepts appeared were in 1992 when Stornetta inserted the Merkle root algorithm into his works [10]. This is the first time when the blockchain was used for theoretical researches.

Merkle tree We will start with the Merkle tree, which is a technology fundamental for creating the "blocks" of a Blockchain. Every block contains some data, and the Merkle tree provides a digital signature (i.e. a unique identifier for a group of data) that guarantee the validity of that data. Merkle tree is essentially an approach to digital signatures. The idea comes from a PhD thesis by Ralph Merkle at Stanford University in 1990 [14]. Merkle tree provides a data structure for verifying individual records. Consider a Hash function, which is a function that given an input of any size will produce an output of fixed size, non-invertible almost surely:

$$f \text{ hash function }, a \text{ input of any size }, b \text{ output of fixed size}$$
$$f(a) = b, \text{ given b is not feasible to find a s.t. } f(a) = b$$

So, every input will have an almost surely unique output. But why inverting f is so difficult? Because f is a non-invertible so the only way to find which input will produce a pre-determined output is to try with all possible input, and given the infinity possible input this operation of brute force will require an incredible amount of hash computations.

Going back to the Merkle tree technology, consider many data L1, L2, ... Ln. Merkle proposed an approach to get a single signature for all of them exploiting the hash function. First, each data is hashed, so we get Hash(L1), Hash(L2), ... , Hash(Ln). The we add all the couples of hash: Hash(L1)+Hash(L2) , and all the others. Then we hash all the sums: Hash(Hash(L1)+Hash(L2)) and so on. This process will iterate again and again until we remain with one hash called *Top Hash* or *Merkle root*. A schema of this process is represented in the figure 1.1.

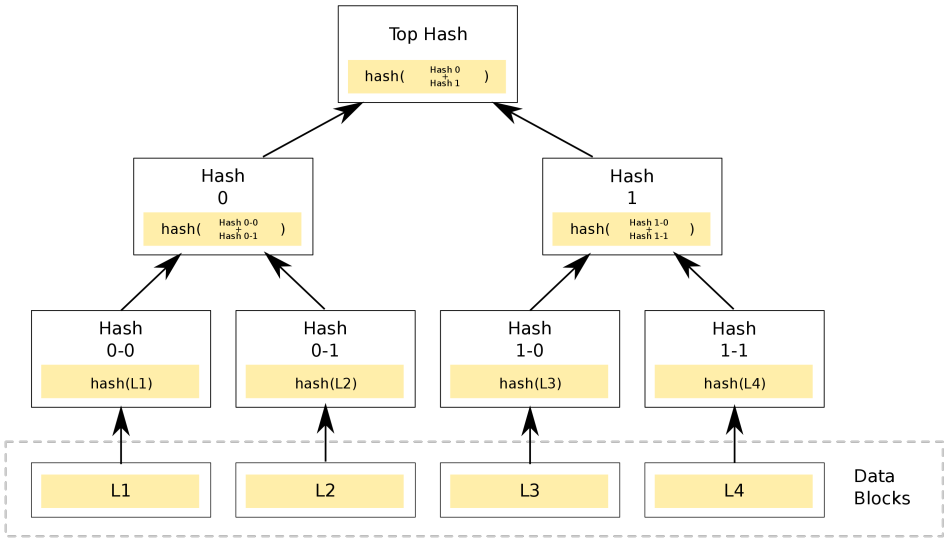


Figure 1.1: Merkle tree schema

Given the set of data L1, L2,... Ln and its Merkle root (with the hypothesis that the Merkle root is correct) we can easily check, for example, that the data L1, L2, ... , Ln have been correctly reported and were not manipulated. Because if a single bit of data has been changed, then thanks to the almost surely uniqueness of the outputs, the Merkle root of the compromised data will be completely different from the correct one. To give an insight about how the output drastically changes even with smaller differences, let's use the SHA-256 function and compute the hash of the words "apple" and "Apple":

SHA256(apple)=

$$3a7bd3e2360a3d29eea436fcfb7e44c735d117c42d1c1835420b6b9942dd4f1b$$

SHA256(Apple)=

$$f223faa96f22916294922b171a2696d868fd1f9129302eb41a45b2a2ea2ebbfd$$

If we consider only the output 3a7bd... , the only way to find the input ("apple") is to try all the possible combinations of letters numbers and symbols of any length. The basic concept of the Merkle tree is that we can have an almost surely unique digital signature for a set of data, and any variation of the elements of our set of data will cause an invalid signature.

Ancient Blockchain In the 1991 Stuart Haber and W. Scott Stornetta created something very similar to the modern Blockchain technology. The focus of their researches was finding a way to prove that a document was existing at a precise time and date in the past, since with digital documents is quite easy to modify and simply backdate them. Haber and Stornetta aim was to introduce a computationally practical solution for time-stamping digital documents, in order to avoid tampering and backdating. To achieve this result, they used cryptographic techniques. In their paper [10] they created some mechanisms to produce a digital timestamp and order registered file in a unique and safe way.

Smart Contracts In 1998 Nick Szabo (computer scientist and law scholar) designed Bit gold. This was only a theoretical work that has never become available since it did not prevent double spending in an effective way, anyway the core ideas behind the project were:

- Decentralized structure
- Privacy focused approach
- Use of cryptography
- Some proof-of-work

All these concepts have inspired the bitcoin protocol, which will come ten years later.

Proof of Work In 2002 Adam Back invented the modern concept of Proof-of-Work, that will be adopted by Bitcoin. The name of the project was Hashcash [3]. The aim of this system was to reduce spam emails and the DDoS attacks. The main idea was the creation of a token that was a proof that the user was not a spammer, in particular that he invested some resources of his computer (CPU time) in order to be able to use internet resources (for example email). So before sending an email the user will have to find a random number that satisfies a hash condition (e.g. the first five digits of the SHA1 160 bit hash equal to zero), for a normal user this task requires less than a second, but if you are a spammer that send millions of emails, it isn't feasible.

1.1.2. Bitcoin

On the 11^{th} of February 2009 a new thread appeared on the forum p2pfoundation.ning.com, its title was: "Bitcoin open-source implementation of P2P currency". The author (Which was under the pseudonym of Satoshi Nakamoto) presented a technical whitepaper [15] where he discussed the possibility of a form of currency without the need of a central server or trusted parties. The system was completely based on a decentralized cryptographic algorithm, this was possible thanks to the Proof-of-Work Blockchain. Now we'll recall some technicalities, in order to make sure the rest of this work is more understandable. Firstly, we remark that the Blockchain is a decentralized distributed ledger, where information is written on "blocks", and they're linked and ordered together exploiting an hash function. A description of those blocks is done in figure 1.2.

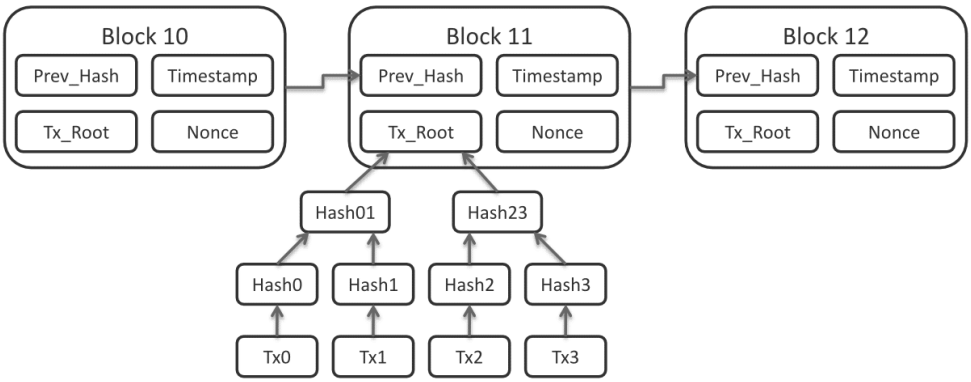


Figure 1.2: Block structure

If we consider a normal block inside the Blockchain we'll find several components, among them there are:

- List of all the transaction of the block (Tx0, Tx1, Tx2, ...)
  and the relative Merkle root (Tx_Root)

- Hash of the previous block (Prev_Hash)
- Nonce
- Timestamp

Blocks are generated by some nodes of a peer-to-peer network called “miners”. These miners collect all the transaction and verify them using digital signatures and avoiding double-spending. After that, they compute the Merkle root of the transaction, so we have a unique digital signature for all the transactions. Then miners start what in general is called "mining" process, which is the research for a number called *nonce*, such that the SHA-256 hash of the block start at least with a fixed amount of zeros. If we consider the SHA-256 output like a hexadecimal base number this is equivalent of try all possible nonce until we find a number less than a threshold. The fixed number of zeros is decided by the algorithm in order to set the difficulty of the mining process. The difficulty is proportional to the time spent solving the problem, and the protocol is coded such that the mining time for producing one block should me more or less 10 minutes.

1.1.3. After Bitcoin

After the creation of Bitcoin in 2009 several other cryptoassets have been created over the same protocol.

Namecoin The first one comes in April 2011 and it’s called Namecoin. It is a fork of the Bitcoin software and the main innovation is the possibility to register internet domain on the Namecoin Blockchain. With Namecoin anyone can register a website name on a blockchain and move it like a cryptocurrency.

Litecoin In 2011 the Bitcoin miners stopped from using CPUs for mining since the difficulty of creating a new block has increased, so they started using GPUs. Part of the community did not like the high barrier to entry in the mining world, so they forked another version of Bitcoin called Litecoin. Litecoin instead of using the SHA-256 algorithm used "scrypt" which was "GPU-resistant", so miners are limited to the CPU power. Litecoin does not add any extra functionality over Bitcoin.
Many other cryptocurrencies have been created, some of them with their own blockchains, other are built on pre-existing blockchain thanks to the existence of smart contracts.

1.1.4. Proof-of-Work (PoW)

How to add a new block to the Blockchain? And how can we "elect" a miner to produce it?
Users send the signed transactions to the network, and miners collect them. After that miners check that every transaction has a correct cryptographic signature, and that the bitcoin are not already spent. Then the miner computes the Merkle root in order to have a unique digital signature for all the transactions. Miners apply timestamp and add the Hash of the previous block. After that the hard part of the mining process begin, now the miners have to find the number called nonce, which is a 32 bit integer number such that the hash of the whole block start with at least a fixed number of zeros. The only way for the miner to find that number is to try all the possible numbers, like a brute force attack.

Let's see an example: we consider a block with three information:

- Number of the block
- Nonce
- A string

The block is stored simply concatenating the elements, so block number 1, with nonce 1, and string= "information" will be stored as: "11information", and it's SHA256 hash is:

$$7b7da34af0759a9251f9238500de6de94c4abb037d7159597aead37e838df976$$

Suppose that our PoW requires a hash of the block that start with at least four zeros, we can try with Nonce=2, and check if it's right. If not, we move to next number which is Nonce=3 and so on. When we try Nonce=16459 (we recall that in this case the block is "116459information"), the SHA256 hash will be:

$$00004f9464c98f66c16fa102101d4c944b06efffe39fe3fdd03d17958671288ef$$

This nonce satisfies our condition. So this is a valid block and if we are the first miner that discover it, we can broadcast our new block in order to let everyone verify it and add it to the Blockchain. The mining process is encouraged with a reward for everyone who first find a valid block and attach it to the Blockchain.
In the case of two miners that create a valid block at the same time a fork of the Blockchain could happen, in this case the system will automatically select the branch which is longer.

The main drawback of the PoW is the huge amount of computational power needed in order to mine a block, since the system is designed to increment the difficulty of the mathematical problem and so the amount of computation needed. In the graph below we can see the amount of computational power, the hash rate, which is how many hash are computed in one second by the miners in order to find the right Nonce for the block.

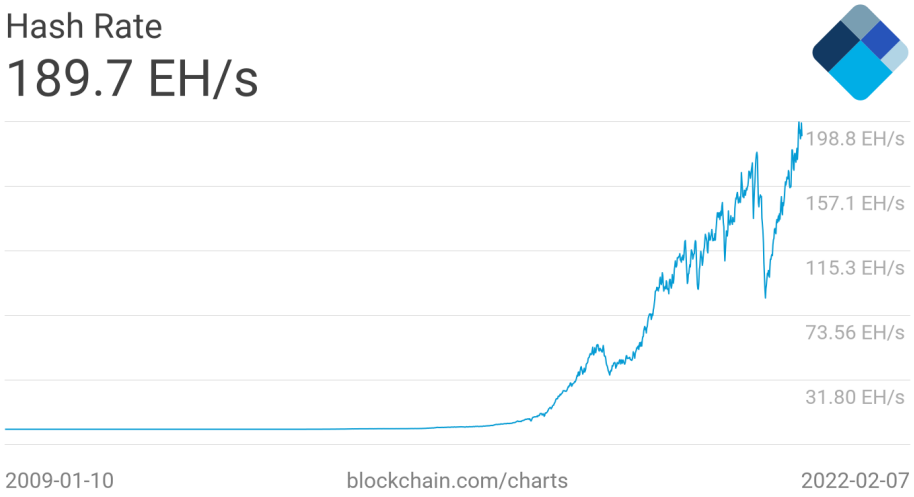


Figure 1.3: Hash rate of the Bitcoin netowork (from blockchain.com)

Another drawback is that the PoW tends to centralize the network. For miners it is more efficient to compute the solution together, rather than alone. Miners tend to group into something called "pool", so every miner is working on the solution and if someone finds it, they divide the reward. Actually the majority of the mined blocks from the beginning of the Bitcoin protocol were generated inside those pools (figure 1.4).

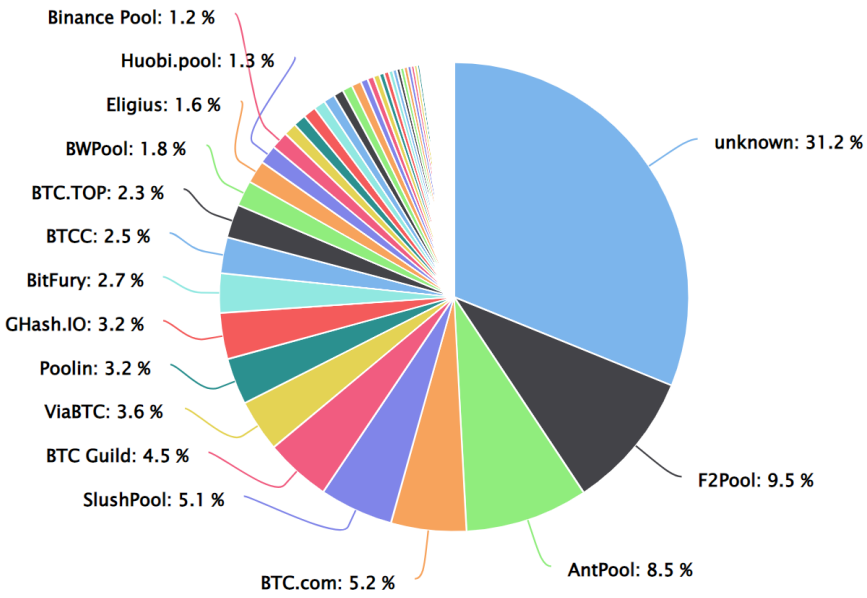


Figure 1.4: Pool Distribution (from btc.com)

PoW tends also to limit the network throughput in terms of transactions, therefore, has an impact on scalability. To give an idea of the magnitude of the scalability problem of Bitcoin, we should consider that the Bitcoin system can process more or less 7 transaction per second, this is a small number of transactions compared with the average of 1'700 transactions of the Visa network.
Proof of Work is not energy efficient and presents a serious problem of scalability. One possible solution for these problems is to change the consensus mechanism. One of the most used alternative is the Proof of Stake, that will be described in the next section.

1.1.5. Proof of Stake (PoS)

The first implementation of the Proof of Stake is a cryptocurrency called Peercoin. Peercoin was created on 12 August 2012 and after that several other coins adopted the PoS. Even Ethereum is trying to transact from PoW to PoS.

In PoS we don't have miners anymore (in a strict sense) since there is no Nonce to calculate. Instead the name of the node who writes on the Blockchain is now "validator". A validator is a node of the network who has blocked some funds in form of cryptocurrency (this operation is called "staking") in order to have the possibility to write the new block and collect part of the transaction fees of the block. The system selects the new validator in a pseudo-random way, keeping in account the amount of stake (more fund in staking, more probability to be the next block writer). The native coin now is not a proof

of computational power spent in the system, but represent the potential future mining power. The validators are encouraged to behave in a good way, since if they approve some fraudulent transaction, then they will lose their fund in staking. Also the miner should behave honestly considering that he will collect part of all the fees of the blocks that he produces, and this will increase the possibility to be elected again as block creator. So other than the fear of losing the stake there is an economic stimulus.
For further insights of the Proof-of-Stake, please refer to [19].

These are the main ideas behind the PoS. Every Blockchain that uses PoS has a different way to implement it and how to choose the next validator.

The consensus mechanism is fundamental in order to determine some aspects of the Blockchain, for example the incentive model, the governance and the security.

Also the PoS usually leads to small transaction costs, this is almost always desirable, and fundamental if we want to create a coin that replicate the value of a fiat currency. For example a cryptocurrency that has a value of 1 US dollar will not be considered for small transactions if the fees are about 20 US dollars.

1.2. Smart Contracts

The idea of smart contracts comes from Nick Szabo in 1994, he proposed to use distributed ledger to store contract [21].
From his work we report the definition of smart contract:

A smart contract is a computerized transaction protocol that executes the terms of a contract. The general objectives of smart contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitration and enforcement costs, and other transaction costs

The first Blockchain with capabilities to run smart contracts is Ethereum, created by Vitalik Buterin. Ethereum aims to be "A next generation smart contract & decentralized application platform" [6].
With the programming language Solidity (similar to javascript) anyone can write code that can be deployed on the Ethereum network, and then interact with it.
To provide an example, let's consider Kickstarter[1], which is a website that allows entrepreneur to propose a new product to the public, and the if the project gains a minimum threshold of money then the realization phase can start. In this frame users and entrepreneurs must trust Kickstarter and also pay fees for it. With a smart contract we can create a smart contract on the Ethereum network that allows entrepreneurs to create a new project and set an amount of Ether that can help the production of a new product. Users can send Ether to the smart contract and if the project reaches the goal, then the Ether will be transferred to the entrepreneur address. Otherwise, all Ether will be given back to the users. All this happens in a decentralized and autonomous way.
Smart contracts allow also the creation of tokens over a Blockchain, like a bank, we can emit some tokens, give them to some users and then they can transfer those tokens exploiting the smart contract functionalities. This will be one of the pillars that will allow the creation of what we will later call stablecoin.

It is important to mention the concept of **dApp** (Decentralized Application). Briefly, a dApp is a combination of a smart contract with a front end interface that allows an easy interaction with users and Blockchain. So any user with a basic knowledge of how a

[1] kickstarter.com

Blockchain wallet work can interact with smart contracts.

1.3. Token

Tokens are a crucial point in Blockchain technology since they allow the passage from a pure currency point of view to a broader one. Tokens were born with smart contract and they unlock almost infinite possibilities of a Blockchain network. They are central in this thesis since almost all the stablecoins are de facto tokens.
But what are tokens? We use tokens every day: for example a train ticket is a token that gives the possibility to travel in a specific train at a specific date and time. Another example, could be the fidelity card of a supermarket, we can accumulate fidelity points for the money that we spend in that supermarket and then if we have enough points we can get some rewards. The idea of token, it's quite abstract since it's something that can represent almost anything. It can represent money, objects, rights or obligations. In this subsection we will define a particular kind of token focused on the Blockchain technology which is: cryptographic token.

Definition. *Cryptographic tokens represent digital assets or authorizations issued on a Blockchain to exchange value, exchange verifiable data and achieve coordination between users. They are managed by the Blockchain protocol or by smart contracts [18].*

Now we're going through the classification of tokens, this is a very important topic for several reasons:

- It helps in the pre-launch phase. Since it is fundamental to design a token that fulfills at best what the network is meant for. With a detailed classification we can make the best choices. And also we can develop new concepts outside the current classification schema.
- Classification helps in finding investors, since we can better communicate the purposes and functionalities of the token.
- Increases the speed of development of a comprehensive regulatory framework. It is easier to develop rules for an object that is well known in all of his facets.
- Increases the speed of token deployment. Since it helps to show the characteristics of the token to a non-technical audience and this is also helping to reach mass adoption.

There are several possible ways to classify tokens, in order to remain inside the topic

of stablecoin we distinguish token by the rights that they could represent. The possible rights that a token can provide are several, we would like to mention:

1. Right to an asset. In this case the token represents an asset, so the owner of the token is the owner of the asset. The token acquires the value and have all the rights connected to that asset. If the token is pegged to an off-chain asset, then is called asset-backed. For example, consider a token that is backed by a share, in this case the owner of the token is also the owner of the share and so he is the owner of that part of the company, and he has all the rights connected like the right to get dividends or the right to vote.
2. Right to do a work. This kind of token is usually native, and it gives the holder the right to do the work and receive, in return, a portion of the fees associated with the work done. This term is also used for tokens which are just rewards for users who complete a certain action or show a certain behaviour.
3. Right to use a good or service that others possess. The owner of this token has the legal right to use something in someone else ownership. Usually they're called usage tokens.
4. Voting right. This token gives the owner the right to vote for something. This vote may regard the Blockchain related matters, or something external. Usually they're called voting tokens.
5. Multiple rights at once. The above rights are not mutually exclusive so a token can grant multiple rights in a single token.

Another kind of classification is the one proposed in [17]. The parameters considered to represent the four main dimensions to classify a token are:

- Purpose parameters: this is the aim of the token in the network. This purpose is influenced by its class, its function and its role.
- Governance parameters: the characteristics that are related to the way the platform is governed and managed.
- Functional parameters: the token characteristics that may alter its ownership or its existence.
- Technical parameters: technical attributes of the token.

We can see the classification table in the figure 1.5

Please refer to [18] for further information about tokens.

Purpose Parameters	**Class**	Coin / Cryptocurrency		Utility Token		Tokenised Security	
	Function	Asset-Based Token		Usage Token		Work Token	
	Role	Right	Value Exchange	Toll	Reward	Currency	Earnings
Governance Parameters	**Representation**	Digital		Physical		Legal	
	Supply	Schedule-based	Pre-mined, scheduled distribution		Pre-mined, one-off distribution		Discretionary
	Incentive System	Enter Platform	Use Platform		Stay Long-Term		Leave Platform
Functional Parameters	**Spendability**	Spendable			Non-Spendable		
	Tradability	Tradable			Non-Tradable		
	Burnability	Burnable			Non-Burnable		
	Expirability	Expirable			Non-Expirable		
	Fungibility	Fungible			Non-Fungible		
Technical Parameters	**Layer**	Blockchain (Native)		Protocol (Non-Native)		Application (dApp)	
	Chain	New Chain new Code	New Chain, forked Code		Forked Chain, forked Code		Issued on top of a protocol

Figure 1.5: Extended tokens classification [17]

2 | Stablecoin: introduction and classification

In this chapter, we will introduce the concept of stablecoin, in term of definitions, features and how it can be classified. In particular, we're going to explore two alternative classification methods: one provided by the European Central Bank (ECB) [9] and the other provided by a research team of the Massachusetts Institute of Technology (MIT) [12].

2.1. Definition and History

The idea of cryptoassets with "stable" price started to spread in the community some years ago, so several initiatives and projects started studying possible approaches in order to reach stability. So a part of the community was in search of a coin that did not change in price, namely a "stablecoin".
Stablecoins are one of the major innovations in the cryptocurrency domain, since most of the cryptocurrencies manifest huge volatility in price. Let's consider, for example the prices of the first four cryptocurrencies by market capitalization (price of one coin $\times$ number of coins available), in figure 2.1.

With that information in mind, we can deduce that a cryptocurrency in general is not the best tool to exchange value between people. The price fluctuations will make holding a cryptocurrency a very risky option. Therefore, if someone want to buy something with cryptocurrency you need to choose one with a stable value. The stability can be achieved in relation to several asset, mainly fiat currencies or commodities like gold. This has nothing to do with how the stablecoin mechanism works, but it is about "with respect to what is stable". We consider stable our local value, for example, if we live in the European Union the Euro appears stable to us. But if you live in the United States and your life is centered on the US Dollar you can observe that the Euro price in USD has some volatility. For this reason it is crucial to identify with respect to what a stablecoin is referred to. And we should also consider that a stablecoin could be referred to a combination of asset,

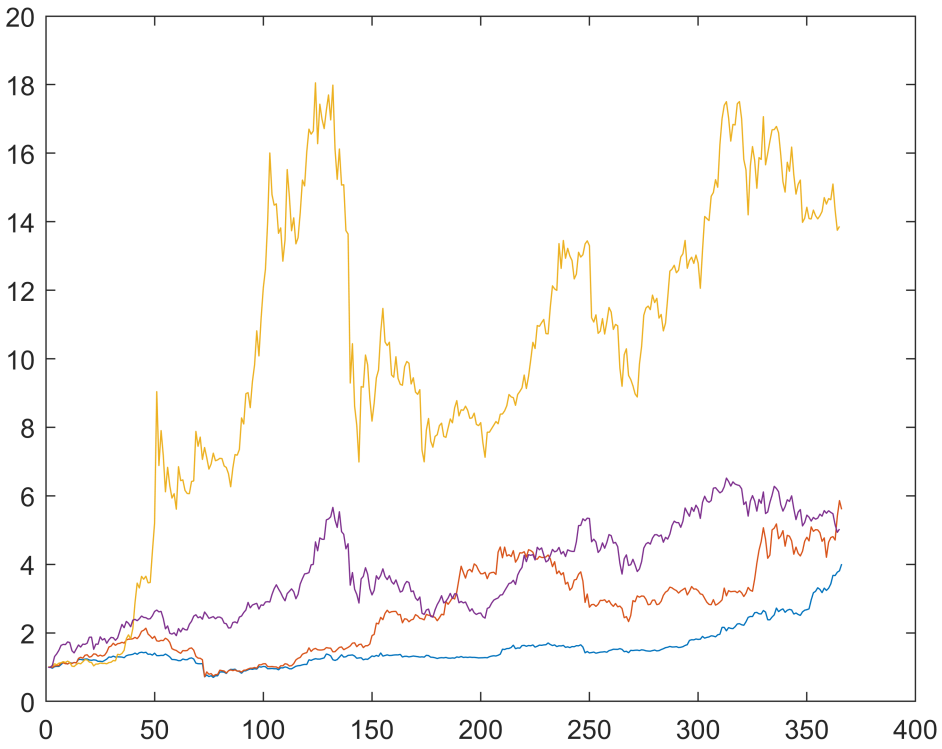


Figure 2.1: Normalized prices of BTC, ADA, ETH and BNB in 2021

for example 50% Euro and 50% Us Dollar.

We now formalize the definition of stablecoin:

"Digital units of value that are not a form of any specific currency (or basket thereof) but rely on a set of stabilization tools which are supposed to minimize fluctuations of their price in such currency(ies)"

This definition is the one provided by the ECB (European Central Bank)[9].
And as highlighted in [12], is a good definition for three main reasons:

1. It's technology neutral, so we can exclude already existing forms of currencies that simply use DLT, so we differentiate a new form of money from e.g. commercial bank money that uses a new technology.
2. There must be some form of stabilization mechanism.

3. Highlight that a stablecoin coin has its own price expressed in the target currency and is not always necessarily equal to one.

So as we said before, every stablecoin has an asset (or a combination of assets) of reference. The objective of the stablecoin is to replicate the value of this asset. It could be a commodity like gold, silver, oil, etc. or a fiat currency like USD, EUR, GBP.
Most of the stablecoins refer to US dollar as we can see from figure 2.2 taken from coingecko.com [1].

#	Coin		Price	24h Volume	Exchanges	Market Capitalization	30d	Last 30 Days Circulation
1	Tether	USDT	$1.00	$36,446,793,651	320	$78,566,806,273	0.1%	
2	USD Coin	USDC	$1.00	$2,394,183,644	249	$52,591,784,211	16.3%	
3	Binance USD	BUSD	$1.00	$2,637,480,995	89	$17,655,701,029	22.8%	
4	TerraUSD	UST	$1.00	$217,009,443	28	$11,564,197,204	8.7%	
5	Dai	DAI	$1.00	$178,657,446	172	$9,828,243,397	7.5%	
6	Magic Internet Money	MIM	$1.00	$68,174,454	25	$2,786,457,627	-39.8%	
7	Frax	FRAX	$1.01	$18,403,309	16	$2,669,126,556	26.4%	
8	TrueUSD	TUSD	$1.00	$107,544,774	69	$1,484,283,366	3.4%	
9	Pax Dollar	USDP	$1.00	$11,397,149	31	$1,078,537,996	2.2%	
10	Fei USD	FEI	$1.00	$12,186,706	7	$554,758,194	-26.4%	
11	Liquity USD	LUSD	$1.01	$7,954,735	7	$498,825,794	-36.8%	
12	Neutrino USD	USDN	$0.995878	$5,553,356	8	$443,140,054	-16.6%	
13	Tether Gold	XAUT	$1,862.05	$983,798	7	$421,470,488	2.1%	
14	Gemini Dollar	GUSD	$1.00	$1,074,875	22	$269,940,633	43.9%	

Figure 2.2: Major stablecoins by market capitalization, from coingecko.com - 14/02/2022

We now consider two of the first stablecoin and the ones with the largest market capitalization: Tether and USDC.

2.1.1. Tether

The initial name was Realcoin, it was rebranded Tether[2] after some months. The main idea was to create a cryptocurrency that for every coin in circulation had a reserve of

[1] coingecko.com
[2] More about Tether at tether.to

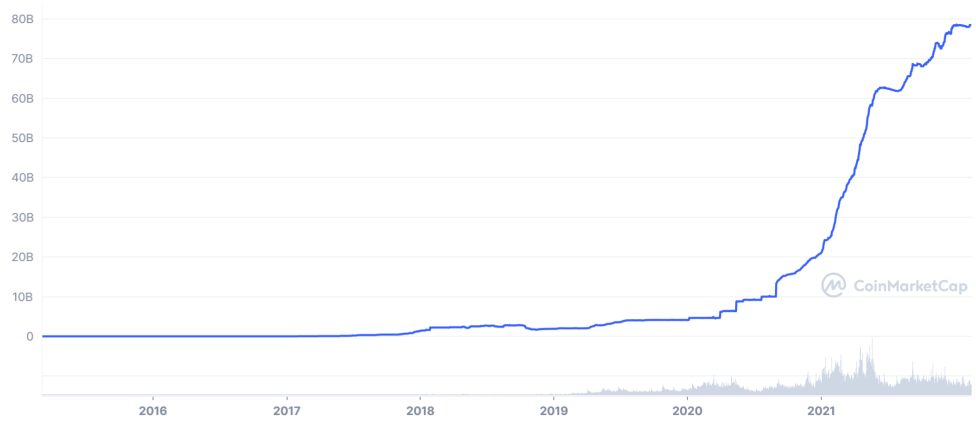


Figure 2.3: Market capitalization of tether

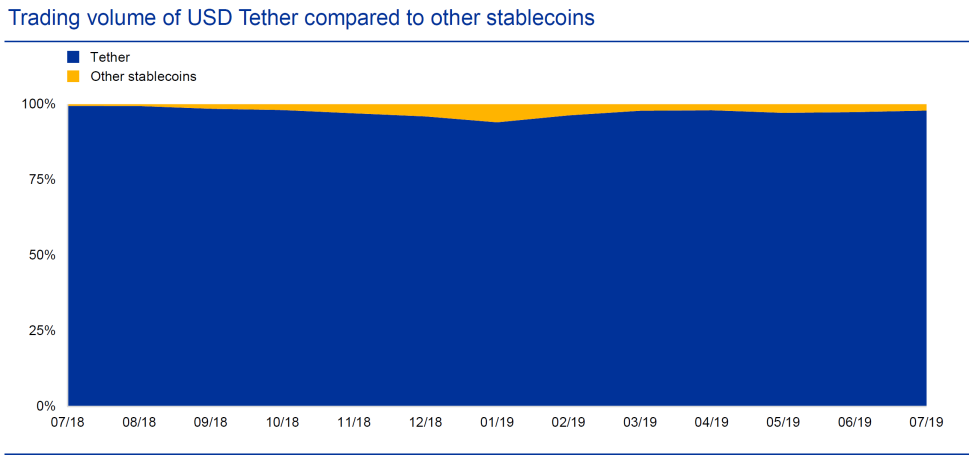


Figure 2.4: Tether trading volume dominance, ECB [5]

one US dollar. The Tether is created by Tether Limited (firm owned by Bitfinex, a cryptocurrencies exchange). The token was built originally over the Bitcoin protocol via omni-layer[3]. Tether has spread among the major blockchains: Ethereum, EOS, Tron, Algorand.

The price is maintained stable by the market itself since every USDT (Tether coin of US dollar) is redeemable for one US dollar. We can observe in figure 2.3 (data from coinmarketcap[4]) that Tether has grown a lot since its foundation in terms of market capitalization. And he also keeps a position of dominance in the market of stablecoin. The largest amount of trading volume of stablecoins is done with tether, as we can observe in figure 2.4.

[3]More about the omni layer protocol at omnilayer.org
[4]coninmarketcap.com

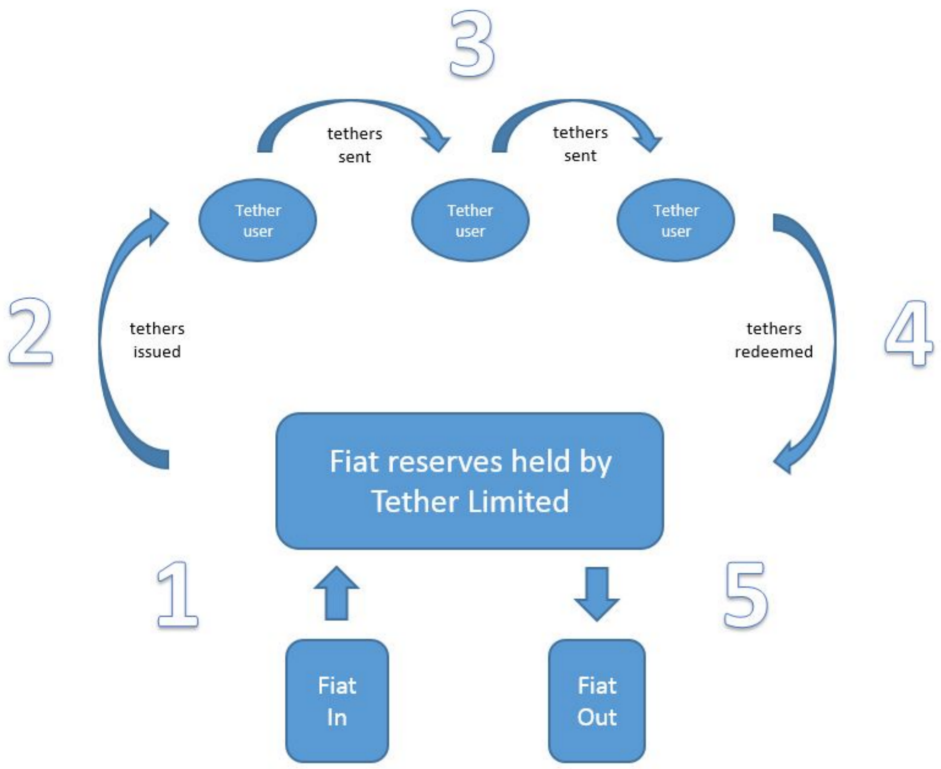


Figure 2.5: Lifecycle of a Tether [22]

Creation and distribution process of Tether are described in figure 2.5

1. Some user sends Fiat money to Tether Limited.
2. User now receive the corresponding amount of USDT.
3. Now he can send USDT to other users (e.g. as form of payment).
4. If a user want to go back to USD, he sends his USDT to the Tether Limited address.
5. Tether Limited sends USD to the user.

This is a simplification of the whole process, since we don't consider, for example, transaction costs and any fees that apply if we exit the Tether protocol. Almost all Tether transactions almost always happen between users in exchanges, since the minimum fiat withdrawal or deposit of Tether with Tether Limited is for sums above 100'000 USD, and we have a fee per fiat withdrawal, which is the greater between 1,000 $ and 0.1%, and the fee per fiat deposit is the 0.1% of the deposited amount.[5]

Tether Limited had some legal problem, and at some point of the legal procedure came out that every Tether was backed only by 0.74 $

[5]More on fees of Tether here: tether.to

As a final consideration on Tether I would like to reflect on the fact that USDT is fully controlled by a centralized entity (Tether limited) and that the decentralization enters the game only in the underlying blockchain protocol, that allows Tether to be transmitted from one user to another. And more important Tether does not have any legal obligation with the users. To be more clear I would like to cite Jacob Silverman [20]: "Tether,[...], isn't decentralized like Bitcoin or many other cryptocurrencies: One company owns, mints, and manages the Tether supply, which means it's also not transparent. And Tether isn't scarce; unlike currencies that are "mined," its production isn't bound by math and code that titrate the supply. Tether Limited, the company behind the eponymous coin, can mint as many coins as it wants. From there, it can use its own currency—and its relationship with Bitfinex, a cryptocurrency exchange also managed by Tether Limited's executives—to buy other cryptocurrencies, conduct unregulated trading, and even potentially launder money."

From this point of view Tether could be seen as an evil thing. Actually, as one of the first players in the stablecoin economy, Tether brought innovation in the field of crypto trading, and made also possible the creation of several decentralized application fundamental for the development of the DeFi world.

2.1.2. USD Coin

USD Coin (USDC) is another stablecoin (2nd after Tether by market capitalization, see figure 2.2) created by the Centre Consortium (Coinbase and Circle are the founder) in 2018. The concept is the same of the Tether, for every USDC there is a US dollar held by Centre Consortium in cash or cash equivalent reserve.
USDC tokens are available on several blockchain: Ethereum, Algorand, Solana, Stellar, Tron and many others. The main advantage in USDC is that this stablecoin is more transparent about the collateralization of its token. The amount of collateral is regularly attested by an external company monthly.

Now we have briefly described the two greatest stablecoins by market capitalization. They were the first stablecoins proposed and they are also the ones with the simplest stabilization mechanism. After those, several other different stablecoins came out, also with different stability mechanisms. In order to make more clear the situation and also understand which mechanism works better, it is useful to introduce a classification. Several classifications are available and there is not a superior one. In this thesis, I will

present two different classifications in order to provide two different points of view on the stablecoins ecosystem.

2.2. Various classification: ECB and MIT

In this section we are going to analyze different classification schemas for stablecoin, since during the years, several methods to maintain the price stable came out, and some are still being discovered. So different institutions have different way to classify stablecoin.

2.2.1. ECB classification

We start from the ECB classification schema [5]. They first introduce the "crypto cube" (Figure 2.6) which is a three dimensional classification model with the following parameters:

- "Existence/absence of an issuer that is responsible for satisfying any attached claim"
- "Decentralisation/centralisation of responsibility over the stablecoin initiative"
- "What underpins the value of a stablecoin and its stability in the currency of reference"

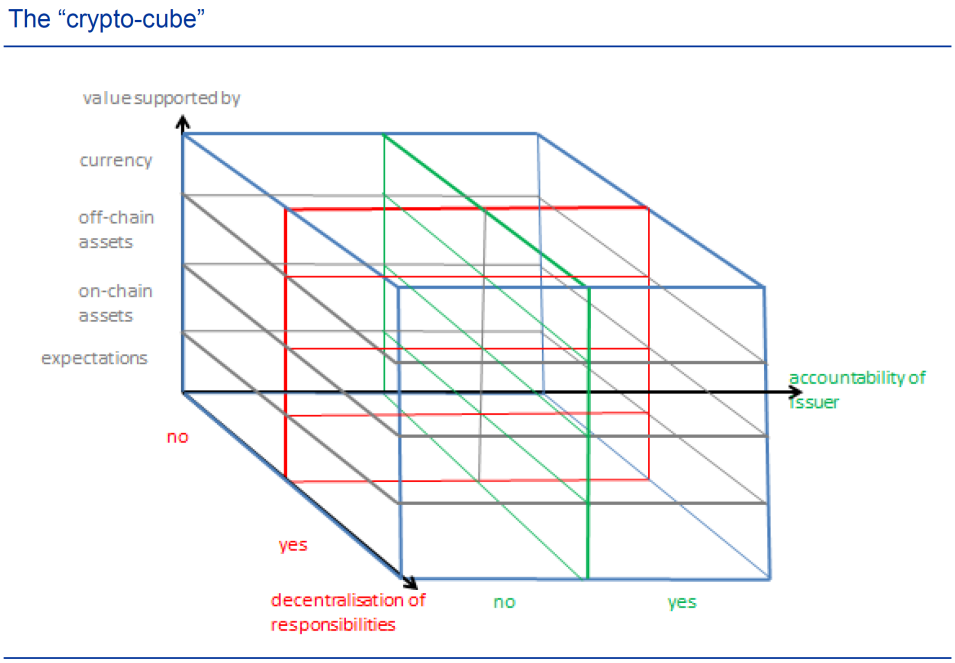


Figure 2.6: Crypto Cube [5]

Every stablecoin can be positioned in this cube. The ECB has found four main types

(based on what is backing their value) that collect the majority of available stablecoin:

- Tokenised funds
- Off-chain collateralised stablecoins
- On-chain collateralised stablecoins
- Algorithmic stablecoins

We can observe how this classification fit in the "Crypto Cube" in figure 2.7

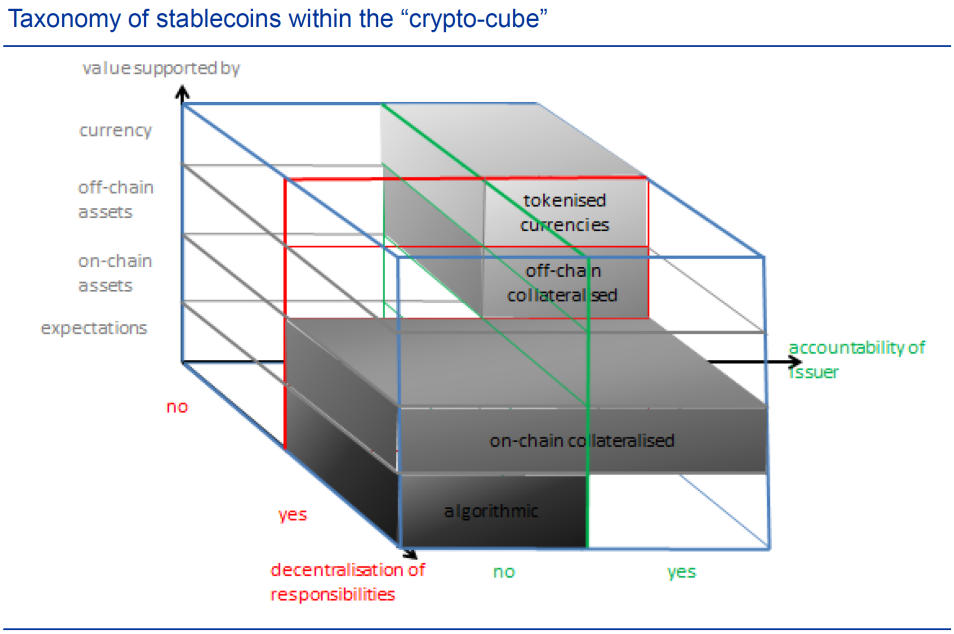


Figure 2.7: Fit of different stablecoins in the Crypto Cube [5]

Now we’re going to explore in the detail the four categories identified by the ECB.

Tokenised funds

This kind of stablecoin is often called "fiat-backed", indeed each unit of the token is backed by a corresponding unit of fiat currency kept in a reserve and granting its value. The lifecycle of a tokenised fund can be characterized by three important moments:

- Issuance
- Transfer
- Redemption

In order to **issue** new stablecoin a user need to send an amount of funds[6] to an custodian account opened by the issuer of the stablecoin that will keep them safe. After the confirmation, the issuer creates a new token (in jargon "mints") and transfer them to the user address via the stablecoin smart contract.
The **transfer** of these stablecoin follows the typical approach of all token over a DLT, this process is fully decentralized and network participants can all verify the transactions.
Once you want your funds back, you can **redeem** them, in a similar way to the issue process but in reverse. You send stablecoin to the address specified by the issuer, that will remove the stablecoin from the circulation (in the jargon "burn") in order to maintain the correct funds/token ratio. Once burnt, the custodian account will transfer the correct sum of funds to the user.
A schema for these transactions is proposed in figures 2.8, 2.9 and 2.10

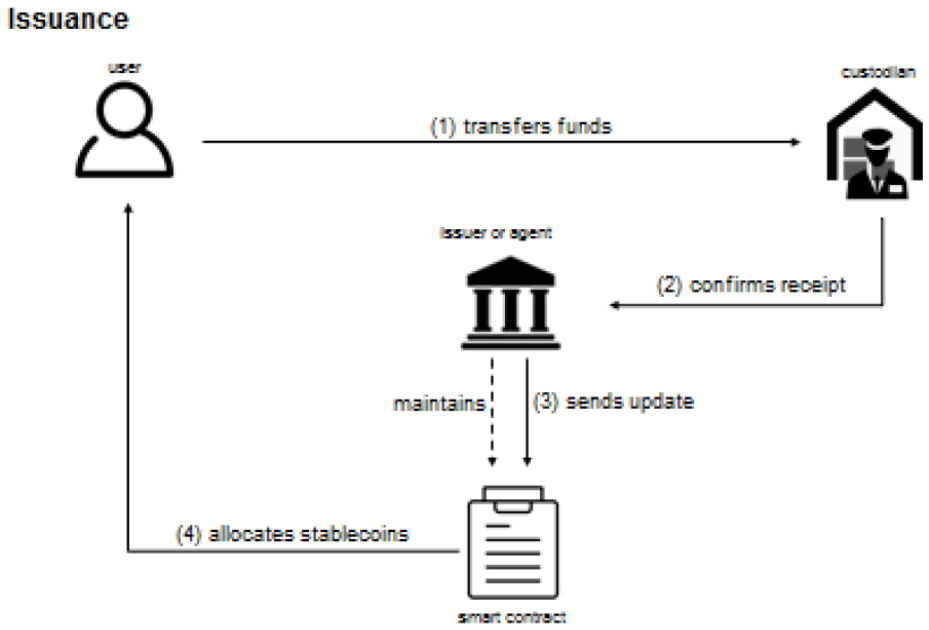


Figure 2.8: Issuance process for tokenised funds [5]

[6]Funds are: cash, electronic money, commercial bank money and reserve deposits kept at a central bank

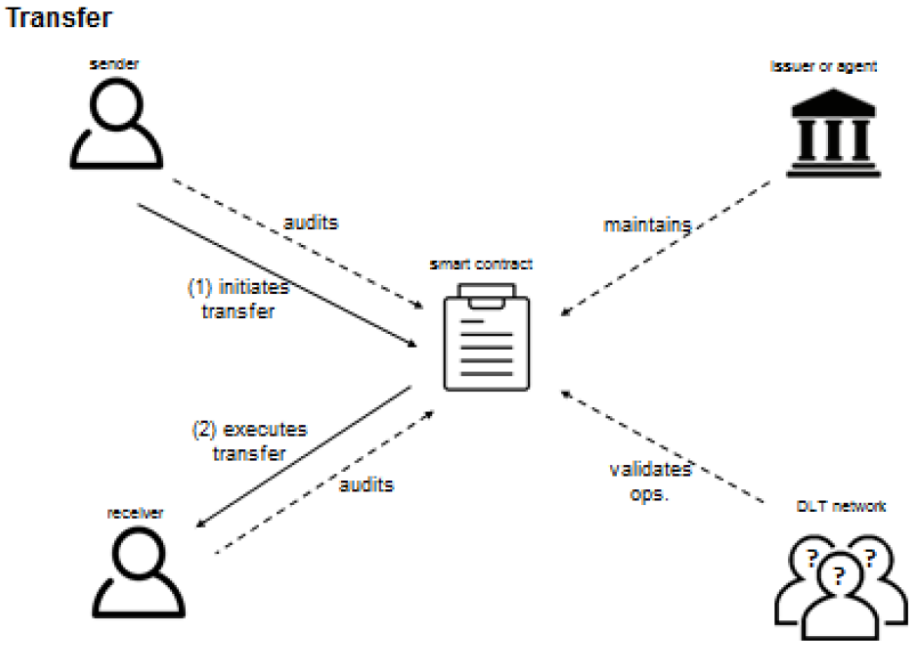


Figure 2.9: Transfer process for tokenised funds [5]

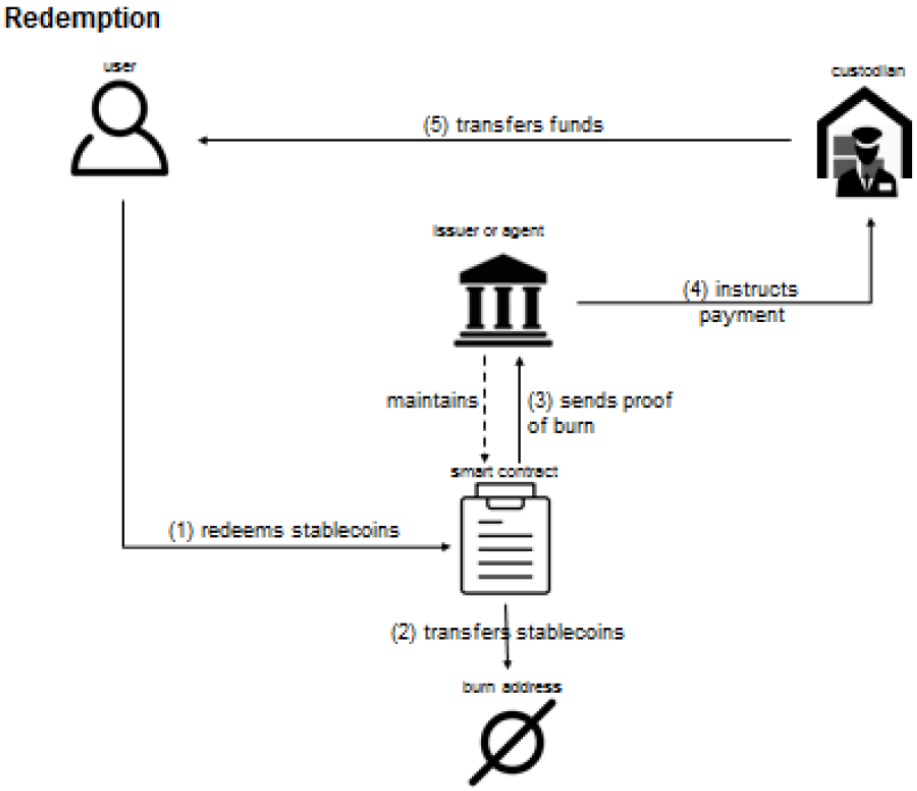


Figure 2.10: Redemption process for tokenised funds [5]

Tether and USDC are two examples of tokenised funds stablecoins.

This kind of stablecoin is the oldest and the simplest one. Being the oldest it has gone through several market conditions. We must mention that some stablecoins mentioned before, like Tether, had some momentary problems due to legal investigation. And in some moments the price of the USDT (Tether US dollar) was below or above one US dollar.

A final consideration on this type of stablecoin is about its decentralization. The tokens are transferred on Blockchain network so the whole value transfer happen in a decentralized way. But the money that is behind a single token is kept safe by a private company, and in some cases this company has no legal obligation to exchange one token for one unit of fiat currency. In this case we can send money without trust via blockchain technology but we must have trust in the entity that is emitting the stablecoin. So we are not in a completely trustless environment.

Now that we have explained the tokenised funds we move to the "Off-chain collateralized" stablecoin which is one of the less common type of stablecoin, but it's worth a mention due to its peculiarity.

Off-chain collateralised stablecoins

There are two main reasons for an off-chain collateral. Firstly, from a legal point of view we have a problem since most jurisdictions do not grant legal effect to the transfer of asset outside the control of any institution. Secondly, in case of physical commodity we cannot transform it in a digital form (e.g. gold). In this case the issuer of stablecoin has the responsibility to: keeping the commodity safe outside the database and deliver it when requested. Let's now analyze the two phases that are different from the tokenised funds: Issuance and Redemption (which can be voluntary or compulsory).

Issuance follows more or less the same principle as for tokenised funds, but we have the additional complexity of transferring the commodity, in some cases the issuer allow user to send non-eligible collateral that will be converted in the original commodity from the issuer. Once arrived the new stablecoin will be minted and sent to the user address. For example, consider a stablecoin following the price of gold. It is no strictly necessary to send physically a gold bar to the issuer. We can simply send an amount of fiat currency or cryptocurrency to the issuer. Then the issuer will transform the received fiat currency or cryptocurrency into gold and will mint our gold collateralised stablecoin.
Redeeming can be voluntary or compulsory. In the voluntary case the process is the same as the issuance one but in reverse. Compulsory redemption may happen when the value of the collateral drops below the over-collateralization ratio[7], this is pre-determined by the stablecoin protocol.
A schema for these transactions is proposed in figures 2.11, 2.12 and 2.13.

This kind of stablecoin is mainly useful when we deal with a physical commodity. A typical example of off-chain collateralised stablecoin is tether gold[8], in this case we have gold bars as off-chain collateral, and the aim of the token is to replicate the price of gold. Like for tokenised funds, we have a decentralized and trustless environment only on the transfer of token part. There is a trust component needed for the issuer and the custodian of the off-chain material.

[7] over-collateralization means that for every unit of value I get, I must deposit a proportional amount given by the over-collateralization ratio. For example, if I want 100$, I need to deposit at least 150$ in order to have an over-collateralization ratio of 150%

[8] gold.tether.to

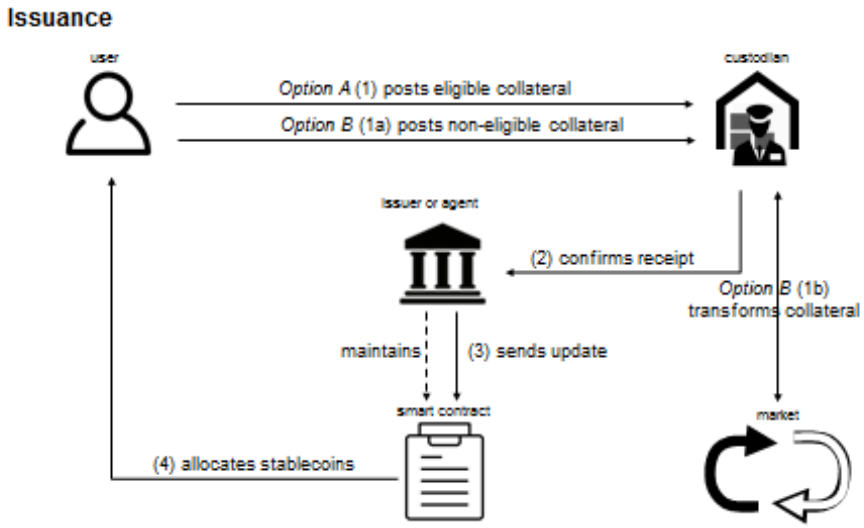


Figure 2.11: Issuance process for Off-chain collateral [5]

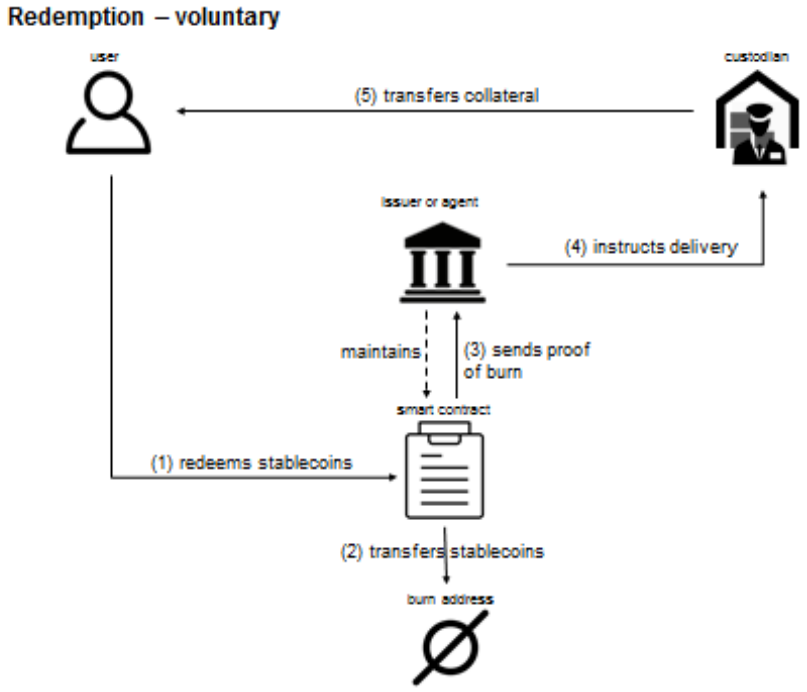


Figure 2.12: Voluntary Redemption process for Off-chain collateral [5]

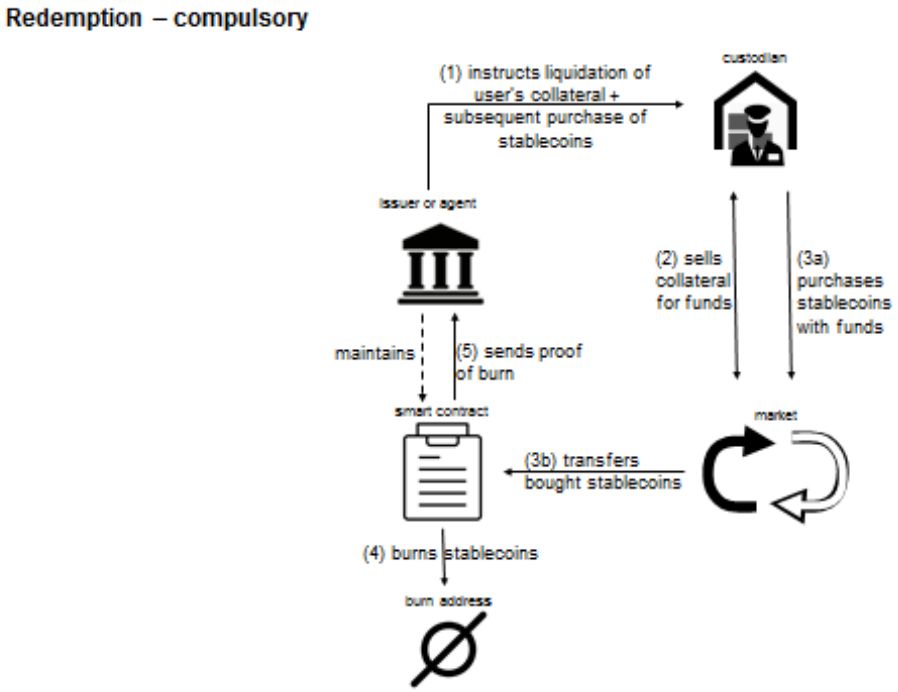


Figure 2.13: Voluntary Compulsory process for Off-chain collateral [5]

On-chain collateralised stablecoins

In this case the collateral is directly present on the Blockchain. This makes thing easier since everything can be managed by a smart contract, in a fully decentralized manner, without the trust of a third party. Stablecoin of this type can vary the type of collateral and the over-collateralisation ratio.

Here there is no accountable party (issuer or custodian), the **issuance** is done in the following manner: a user sends the on-chain collateral directly to the smart contract that governs the scheme, after that the smart contract creates the stablecoin and sends them to the user. The stablecoin amount is decided by the user and must respect the over-collateralization ratio.

The **redeeming** process can be voluntary or compulsory, voluntary in the case the user simply wants back his collateral, and compulsory in the case that the collateral drop its price below a pre-determined over-collateralization ratio. As usual, we provide four schemas for Issuance, Transfer and Redemption in figures 2.14 , 2.15 , 2.16 and 2.17.

In this case blockchain technology is used for transferring tokens, but also for creating the new ones, since everything is managed by a smart contract. This is the first example of fully trustless and decentralized stablecoin. An example of on-chain collateralised

stablecoin is Dai from MakerDAO [9] (A description of the Dai mechanism is done in section 2.3.1). The stability of Dai comes from the market, if the price of the collateral drops in a fast way and the price of Dai moves away from one, the protocol will liquidate the open positions and will create an arbitrage opportunity thanks to the liquidity fee in order to move the price back to one.

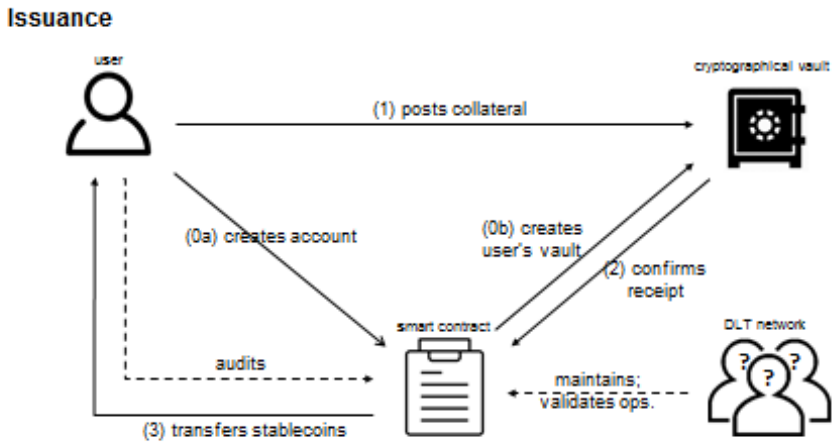


Figure 2.14: Issuance process for On-chain collateral [5]

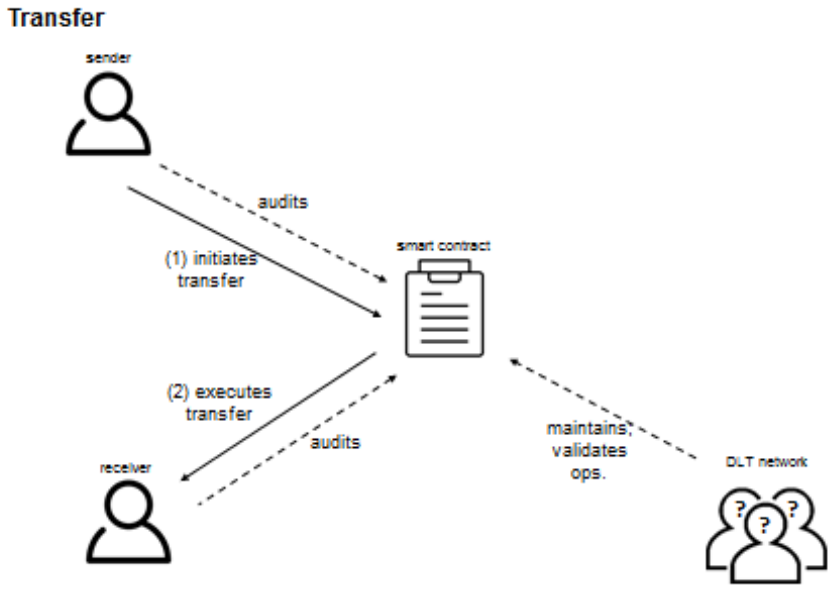


Figure 2.15: Transfer process for On-chain collateral [5]

[9] makerdao.com

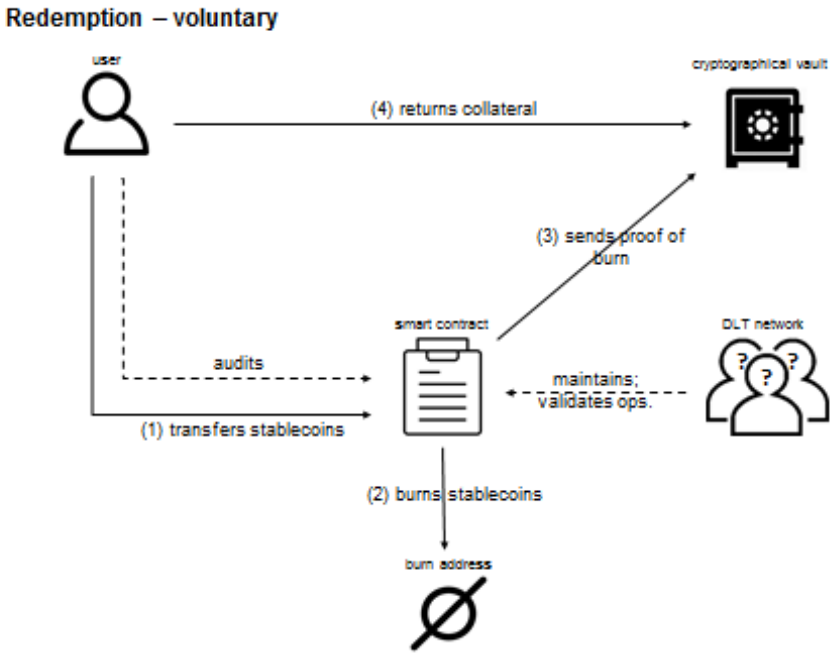


Figure 2.16: Voluntary Redemption process for On-chain collateral [5]

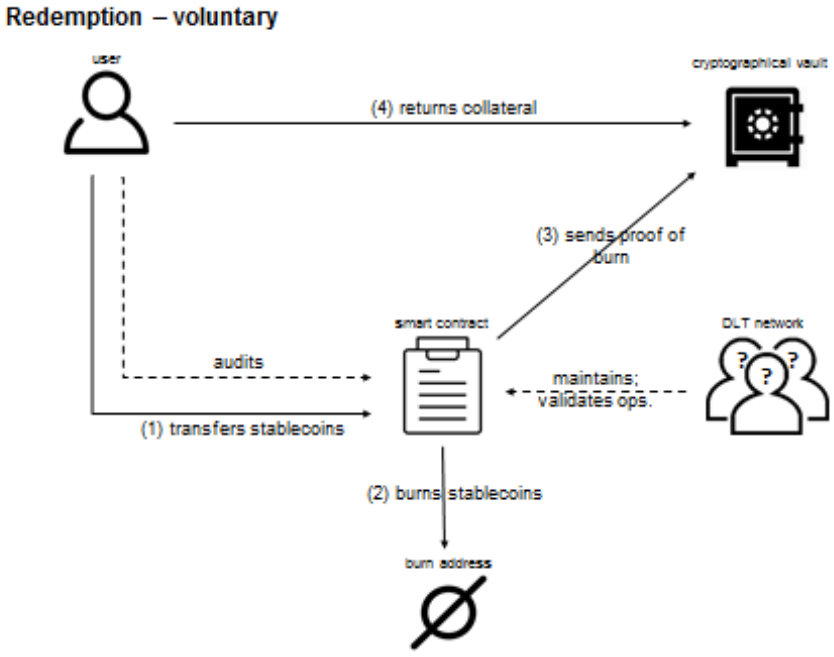


Figure 2.17: Compulsory Redemption process for On-chain collateral [5]

Algorithmic stablecoins

Algorithmic stablecoins are not fully backed by the collateral they meant to represent. In fact, they try to stabilize the market price in some algorithmic way, so this is a wide field since several possibilities have been discovered. There is a smart contract that manages issuance and redemption in order to maintain parity with the currency of reference. The information on excess of demand or supply is reported by some market data providers (in jargon "oracles"[10]) that bring to blockchain information from the off-chain world.
The **Issuance** process is different for almost any algorithmic stablecoin, in general stablecoin are given in exchange of some cryptoasset. The **Redemption** part simply does not exist since there is no assets to redeem. There is a **Contraction** phase when there is an excess of supply, and it's similar to compulsory redemption. In a contraction phase, we have more stablecoin on the market than the effective demand so the price start dropping. In this case the smart contract must reduce the supply, in order to do that, generally the smart contract buys stablecoins from users and burn them. In order to buy stablecoin it can use some reserve or sell some rights to future revenues. We provide two general schemas for Issuance and Contraction in figures 2.18 and 2.19 .

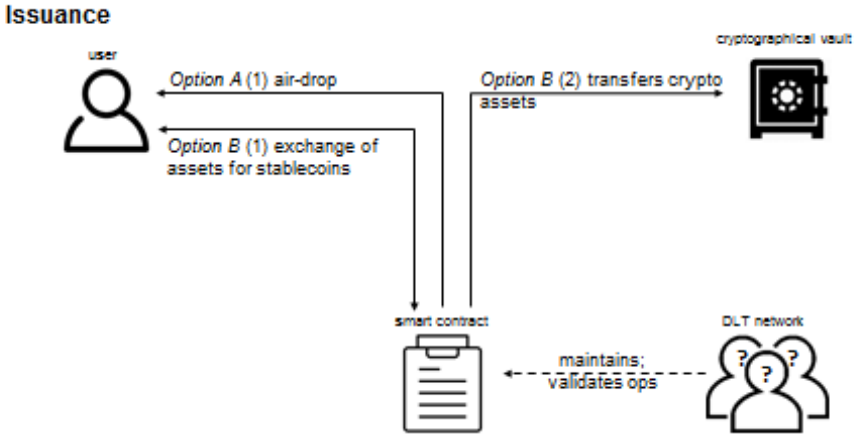


Figure 2.18: Issuance of algorithmic stablecoin [5]

[10]An Oracle is a third party service that captures data from the world e.g. election result, weather, plane delay etc. in order to provide to a smart contract information. In order to avoid manipulation and centralization is better to use a decentralized network of oracles.

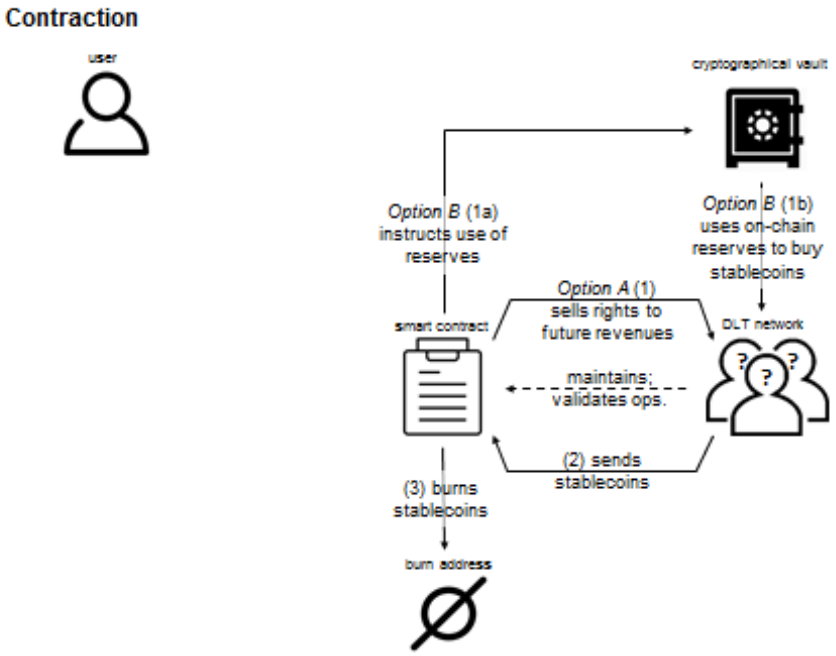


Figure 2.19: Contraction of algorithmic stablecoin [5]

We will further explore the algorithmic stablecoins in the next section since it is possible to classify them by the kind of stability algorithm they adopt. As an example of Algorithmic stablecoin Terra stands out since it has been the largest one by market capitalization. Terra algorithm will be presented in details in the next chapters.

Summary

As a summary of what we have said before, we can observe the table in figure **??**, in order to have an overview over the ECB classification schema:

Summary table of stablecoin characteristics

	issued on the receipt of:	"collateralised" by:	redeemable at:
Tokenised funds	funds (i.e. cash, deposits or electronic money)	funds and/or close substitutes (i.e. secure, low-risk, liquid assets [12])	market value of the collateral at the time of redemption or face value of the stablecoin
Off-chain collateralised stablecoin	assets held through an accountable entity (e.g. securities, commodities, or crypto-assets in custody with an intermediary)	assets held through an accountable entity (e.g. securities, commodities, or crypto-assets in custody with an intermediary)	market value of the collateral at the time of redemption
On-chain collateralised stablecoin	crypto-assets held directly on the distributed ledger	crypto-assets held directly on the distributed ledger	market value of the collateral at the time of redemption
Algorithmic stablecoins	crypto-assets or given away for free	no collateral – value of stablecoin is based purely on the expectation of its future market value	not redeemable

Figure 2.20: Summary of the classification provided by ECB [9]

2.2.2. MIT classification

Now we are going to briefly explore the classification schema provided by [12], which is developed by a research team at MIT.
They define a tripartite classification schema:

- Claim based
- Good-Faith based
- Technology based

In **Claim based** there is a direct legal right to redeem the stablecoins against a predefined amount or value of a reference asset. We can consider as an example USDC.
In **Good-Faith based** there is no legal obligation, but users must trust the good business practice of the issuer. For example, TrustToken emits TUSD which is a stablecoin backed by the US dollar, but in its legal terms is written: "the Company itself does not guarantee any right of redemption or exchange of TrueCurrency tokens for fiat currency"[11]
The last case: **Technology based** stablecoins rely on a technology to autonomously induce price stabilization. There is no legal claim or user's faith in the good intentions of the issuer.
For example, all on-chain collateralized stablecoin (like Dai) and the algorithmic stablecoin (like Terra) belong to this category.

The authors of [12] provide also the position of these three categories in the International Monetary Fund's money tree [2],in figure 2.21.

[11]https://www.trusttoken.com/terms-of-use

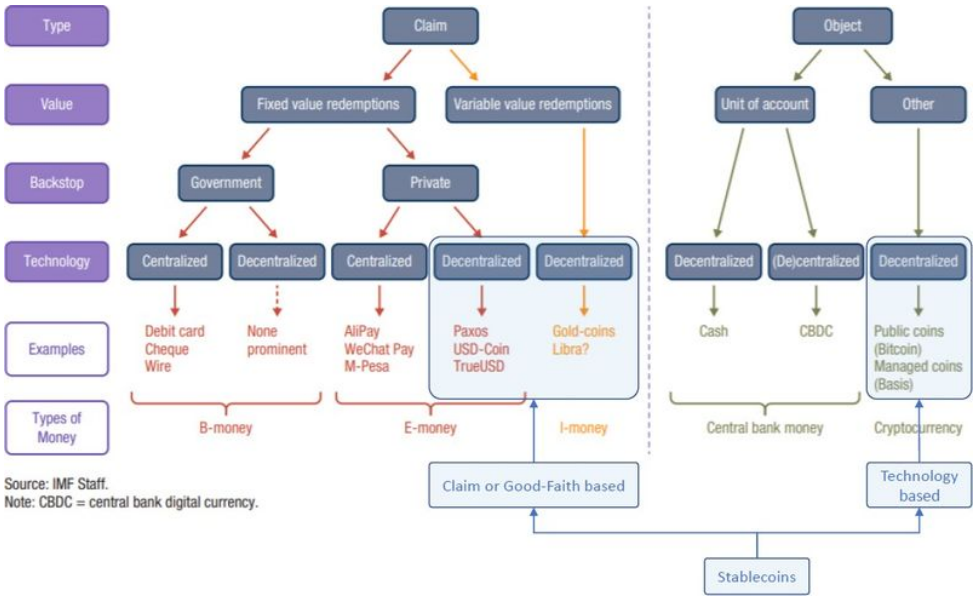


Figure 2.21: IMF money tree and stablecoin [12]

This kind of classification is very different from the ECB's one. The point of view of the ECB is the one of the regulator and policymaker, so it's very important to understand what is the asset that is behind the stablecoin and who has the responsibility for that asset. Instead, the research team of MIT has a more high-level classification since its differentiating stablecoins on the necessity of a legal system in order to use them.

So now we have two possible classifications, now we are going to explore the only common part which is the one related to the algorithmic stablecoins. These are the most complex ones and also the more interesting from the mathematical and economical point of view since they have the ability to move prices in the market via algorithms.

2.3. Algorithmic stablecoin

In this chapter we're going to examine the algorithmic stablecoin, this is the most innovative stablecoin since does not require the use of a fiat currency, and it is completely decentralized.
In order to properly explain how an algorithmic stablecoin works, let's first consider full-reserve bank. In this scenario a user deposit an amount of cash in the bank, then the bank emits the same amount of cash in digital notes that will be created for the user account. Basically a user can deposit cash assets in the equity part of the bank and the bank will issue digital notes to users and writes them as liabilities. This basic model describes stablecoin like Tether, USDC and every full fiat-backed stablecoin.
In this scenario every liability is backed by an asset in reserve, as we can see in figure 2.22.

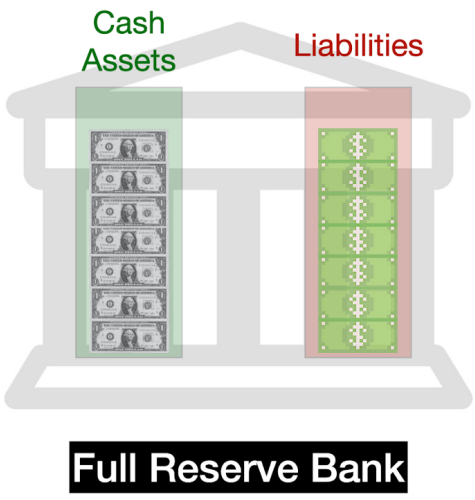


Figure 2.22: Full Reserve Bank

Now we are going to more complex models of stablecoins:

- Full reserved Crypto Stablecoin
- Algorithmic Central Banks
- Seigniorage Shares Stablecoins

2.3.1. Full reserved Crypto Stablecoin

We want to remain in the crypto space, in order to completely avoid fiat currencies, or other off-chain collateral. A first idea could be to create something like a full reserve bank where the equity part is given by crypto assets. But, as we saw before, crypto assets are very volatile, so if the price goes down, then the stablecoin issued by this system will be undercollateralized. And the price will drop below one.
The solution to this problem is to have more equity value than liability, in order to be able to absorb market shocks by the surplus of the equity part. Let's say for example that for every 100 $ of stablecoin minted the smart contract has a deposit of 150 $ in value of any cryptoasset, in this case the stablecoin can absorb a drop of 50$ in value of the crypto asset and still being collateralized.

Dai A classical example for this type of stablecoin is Dai (the same Dai described before), which aims to keep its value equal to one US dollar, Dai is regulated purely by a smart contract that can be changed with the governance token MKR issued by MakerDAO[12] (DAO stands for Decentralized Autonomous Organization). Let's see how DAI works (for simplicity let's assume that 1 ETH = 150 $):

- A user deposit one Ether to Maker's smart contract, so it's creating a CDP (Collateralised Debt Position)
- Dai has a 150% collateralization rate, this means that against the user's ETH that has a value of 150 $ the smart contract will issue maximum 100 Dai.
- Let's say that the user request 50 Dai, so the collateralization ratio is 150/50 = 3 and it is respected since greater than 1.5.
- Now there are two possibilities: the price of ETH rises or drops
  - If ether rises, then the position is even more over-collateralized and Dai becomes stronger. If the price is at premium (greater than one dollar) the maker mechanism incentivizes users to create more Dai in order to grow the supply and reduce the price.
  - If ether drops, the maker mechanism will liquidate the CDP with an auction in exchange of Dai in order to remove the position before the value in ether is less than the amount of Dai emitted.

[12] makerdao.com

- If the user wants his ETH back, he simply needs to pay back the amount that he took out with the addition of a minor fee.

Actually, now Dai supports a multi-collateral system, so there is the possibility to use different crypto assets in order to mint Dai. More about the mechanism of Dai can be found in the official whitepaper [13].

2.3.2. Algorithmic Central Banks

This kind of stablecoin has no redeemable asset in a strict way, the smart contract that manages it defend the peg directly on the market. We're going to analyze two algorithmic central banks stablecoins: Fei and Celo (And in the next chapter we'll go into detail with Terra).

Fei USD - FEI

Fei defends its peg directly in the market. At the beginning Fei has an amount of cryptocurrency as equity and an equivalent amount in value as Fei USD (the stablecoin issued by Fei) as a liability. So 1 Fei USD is backed by 1 $ in value of other cryptocurrency, for simplicity let's consider ETH. Now there are two possibilities on the market:

- ETH price goes up so the Fei USD has a value greater than 1 $.
- ETH price goes down and the Fei USD price goes below 1 $.

If the price of Fei USD is above 1 $ the protocol mints more Fei USD, so now we have more Fei USD as equity and an equal amount of Fei USD as liability, with the new amount of Fei in the equity the central bank buy directly on the market ETH for Fei USD, so the now we have more equity in equity part of the bank, and more Fei USD in the market, this operation moves the market price of Fei USD back to 1 $.

If the price of Fei USD is below 1 $ the protocol will use some ETH present in the equity and buys some Fei USD in the market, in that way there will be less Fei USD available in the market and the price will go up, back to 1 $.

A visual description of what happens if the price broke the peg is presented in Figure 2.23 and 2.24.

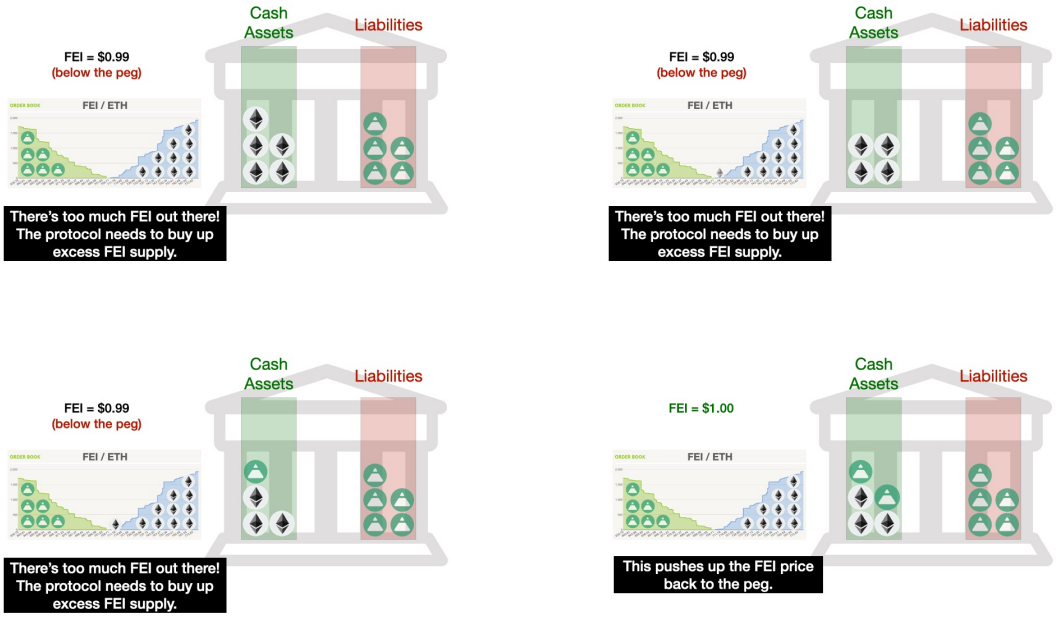


Figure 2.23: Price below the peg

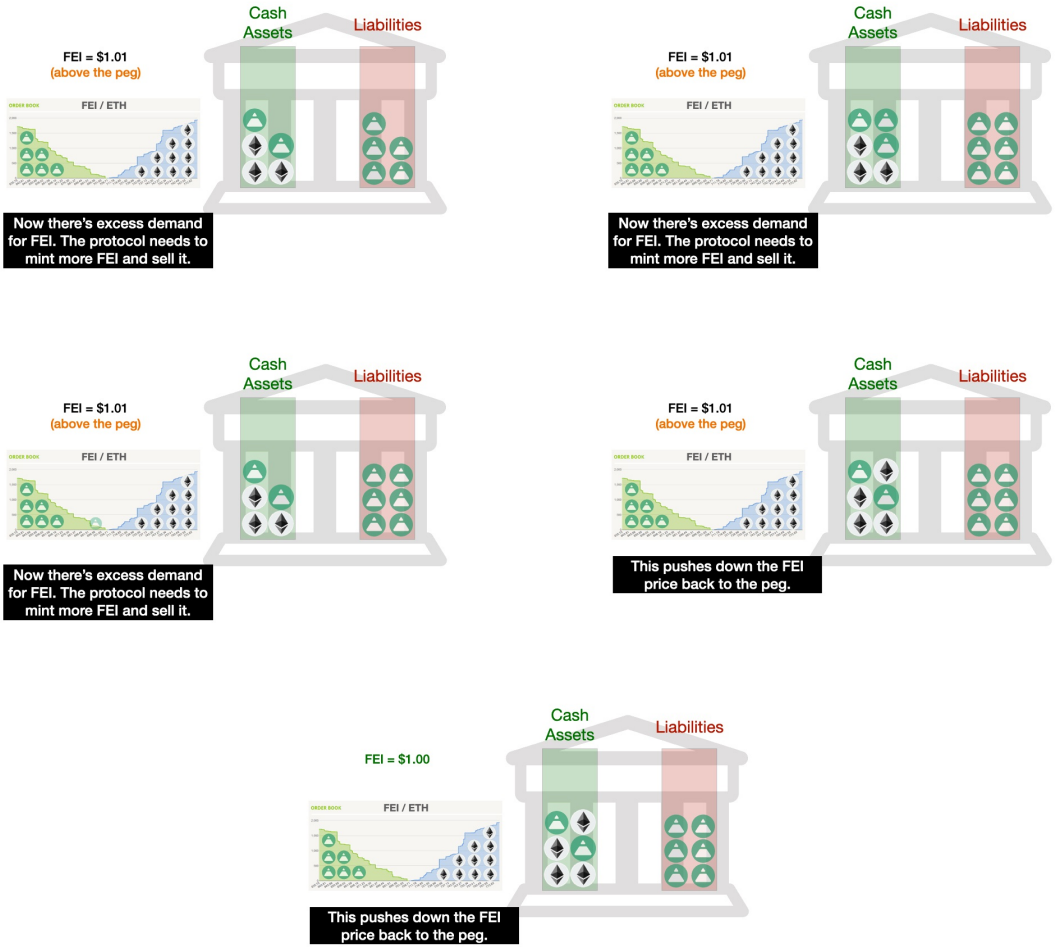


Figure 2.24: Price above the peg

Celo Dollar - CUSD

Celo Dollar works exactly like Fei USD since it's an algorithmic central bank, with the only difference that the equity part stay always over-collateralized. And if Celo's assets dip below 200% of its liabilities the system re-capitalize by collecting transaction fees on CELO transfers.

Algorithmic Central Banks have a similar way of working, they differentiate with options like: assets that can be held and rules over possible collateralization.

2.3.3. Seigniorage Shares Stablecoins

Seigniorage Shares Stablecoins actually do not have nothing that covers the value. To better explain how they work we'll make the example of Basis Cash

Basis Cash - BAC

Basis Cash works along with other cryptocurrency: Basis Share (BAS) and Basis Bond (BAC).
Let's suppose that the price of 1 BAC is 1 $.

- If the price of 1 BAC rises above 1 $ then the system will generate more BACs and gives them to the BAS holders, so with this new amount of BACs on the market the price should go back to 1 $.
- If the price drops below 1 $ then the system will generate new BABs and sell them for BACs, then it will burn the BACs in order to reduce the supply, and move the price back to 1 $.

So the BAS are like shares that generate dividends (paid in BACs) when the economy face an expansionary phase. On the other hand, why should someone buy BABs? Because after the emission of BABs the in the next expansionary phase the system will give the BACs surplus to the BABs holders instead of the BAS ones.

There is also the possibility that a stablecoin stays between two or more categories, for example FRAX [13] is defined as "a cryptocurrency being partially backed by collateral and partially stabilized algorithmically". So there is not a perfect classification schema.

[13] frax.finance

2.4. Role of stablecoin in DeFi

In this section we are going to provide a definition and an example of Decentralized Finance (DeFi). This is an important subject for stablecoins since they are one of the main pillars of the DeFi. In DeFi we want to provide financial services without intermediaries, and most users are interested in financial services in their local fiat currencies. So having the possibility to access a token that has a stable price with respect to the fiat currency of reference is fundamental in the DeFi applications.
Let's now present what DeFi means: DeFi stands for Decentralized Finance. In particular, it refers to the provision of financial services characterised by the reduction or elimination of the role of intermediaries through the use of distributed ledger technologies such as blockchain and smart contracts. More about DeFi can be seen at [4].
With the introduction of smart contract some developers create applications based on smart contracts and Blockchain that allow some financial services. For example: exchange between different token or lending/borrowing services.
Now we discuss about a typical DeFi application: DEX (Decentralized Exchange).

2.4.1. Decentralized Exchange

An exchange is essentially an application that allows us to trade token for another token. We can consider a common stock exchange where we buy or sell stocks for money, in DEX we exchange cryptoasset for other cryptoasset. Let's consider that over a blockchain there are two token: TKN1 and TKN2, some users may want to exchange a certain amount of TKN1 for TKN2, and some other users vice-versa. In the classical Order Book model we would have to match demand and offer. In the decentralized case we can rely on automated market makers (AMM) where Liquidity Pools (LP) are being exploited. Now let's start from the beginning, what is a liquidity pool?

A liquidity pool is a smart contract that allows users to deposit two token (TKN1 and TKN2) in the same quantity of value For example, consider 1 TKN1 = 10$ and 1 TKN2 = 20$, if I want to deposit 10 TKN1 I need also to deposit 5 TKN2. So the value of the two token is balanced in the liquidity pool. In exchange of my tokens I will get other token called LP-token that will represent my participation in the liquidity pool. Now we have a smart contract that has half of its value composed by TKN1 and the other half by TKN2. If a user wants to exchange TKN1 for TKN2 he will need to send the request to the smart contract and the desired amount of TKN1. In exchange, he will receive TKN2. Now the value in the liquidity pool is no more balanced, in that case the smart contract will change the value of TKN1 and TKN2, since we deposit TKN1 now there are more

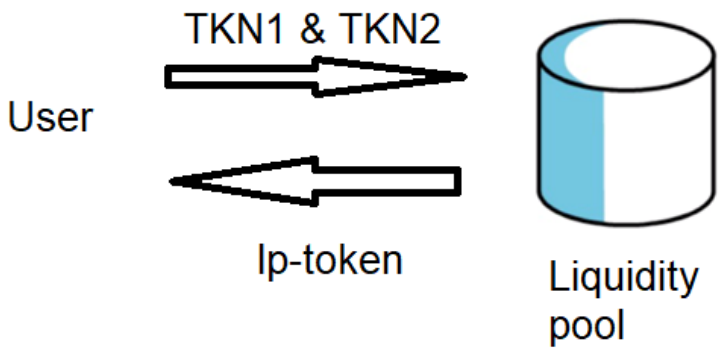


Figure 2.25: Schema of a liquidity pool

than at the beginning, so the price of TKN1 will drop, same argument in the different direction for TKN2 that will become more expensive.
There are multiple ways to compute the new price, one of the most famous DEX, like Uniswap[14] keeps the product of the value of the two token constant. Other DEX, like Curve[15] uses more complex formulas that can be found in [7]. We'll make an example with the constant since it's easier. The new price is calculated thanks to the formula:

$$x \times y = k$$

where x is the amount of TKN1 and y is the amount of TKN2.
Suppose,as before, that the price of TKN1 is 10$ and the price of TKN2 is 20$. Suppose that in the liquidity pool there are 1000 TKN1 and 500 TKN2. The constant given to the formula is given by $k = 1000 \times 500 = 500'000$. Now if we want to exchange 100 TKN1 for some TKN2, we can do the following: after we add 100 TKN1 the amount in the pool of TKN1 will be 1'100. Then we divide k by the new amount of TKN1 in order to get how many TKN2 the user will receive, in this case $500'000/1'100 \approx 455$ so the DEX will give to the user $500 - 454 = 45$ TKN2.
Now we can also compute the new prices of the items in relation to the US dollar. We have now 1100 TKN1 and 455 TKN2, the initial value of the two token was 10'000 $ of

[14]Uniswap.org
[15]curve.fi

TKN1 and 10'000 $ of TKN2. If we divide the US dollar values by the number of token we get the new prices:

$$TKN1 : 10'000/1100 = 9.09$$

$$TKN2 : 10'000/455 = 21.98$$

The prices change according to the new availability of tokens, so since we have more TKN1 its price drops and since we have less TKN2 its price rises.
Now a new arbitrage opportunity is created since I could buy TKN1 at 9.09 and sell it on another exchange for 10 realizing a risk free profit. This opportunity will re-balance the value of the two token inside the liquidity pool. Liquidity pool usually has an enormous amount of value of tokens, so generally even big transaction do not affect the price in a significant way.
More detail about how DEX works can be found in [11].

Stablecoin has a key role in DeFi since it's the main bridge between the Fiat money and the various cryptocurrencies. Just think about the possibility given by a decentralized exchange, when you can buy cryptocurrencies with fiat-like token. Everything that can be tokenized on the blockchain can be traded in a decentralized way. If we consider the borrowing/lending protocol we can use a cryptocurrency like ETH as collateral to borrow some stablecoin, and then use these stablecoin to buy additional ETH so we have a leveraged position. Another example, could be made with application like Curve or Anchor Protocol where you can deposit stablecoin in order to get some interest, in this way we avoid the risk of very volatile market.
The possible applications are virtually infinite and several are being discovered day by day.

3 | Use case: Terra

The aim of this chapter is to explore the Terra 's stablecoins and its mechanism to maintain the peg to a currency of reference. The choice of Terra comes from the fact that it is a paradigmatic example of an "algorithmic central bank" system for stablecoins. This system has demonstrated to behave very well in adverse market conditions, since, except for two days, its value has been stable and more important the market capitalization grew from the initial 13 million USD to the actual 11 billion USD. Moreover Terra has a very interesting mechanism to maintain stability that will be explained later.
Terra is a set of stablecoins, not a single stablecoin, so there exist Terra USD, Terra EUR and many more. During the next chapters we will refer at Terra as one single stablecoin that can be one of the many available in the Terra ecosystem.

Now we're going to explore in detail how the Terra mechanism works. Most of the work presented in this chapter comes from the official Terra whitepaper [8].
The main drivers of Terra projects are these two following:

- Stability, since nobody wants a payment method with huge volatility.
- Efficient fiscal policy, in order to give money to promising application that encourage the use of the Terra platform as mean of payments.

We are now going to explain them in detail.

3.1. Stability

We need to go through three main aspects:

- How to define price-stability?
- How to measure it?
- How to achieve it?

Since most goods are consumed domestically one single stablecoin is not enough, we should create a family of cryptocurrencies pegged to the world's major currencies. So we'll have

several Terra currencies pegged to USD, EUR, CNY, JPY, KRW and the IMF SDR [1], and other currencies can be added by user voting.
Other than the Terra currencies we have also Luna, which is a volatile native coin on the Terra blockchain, and its purpose is to be the reward given to miners and also the main coin to pay the fees with.
The system supports atomic swaps[2] among Terra currencies at market exchange rates, this allows all the Terra's currency to share liquidity and macroeconomic fluctuations.
So we define stability as the *ability of every Terra regional coin to reflect the respective regional currency* (e.g. UST should reflect USD).
As we said before, we will refer to Terra as a single currency.

How can the system measure stability? With miner oracles[3], since the price of Terra in the secondary market is exogenous to the blockchain, a decentralized price oracle to find the true exchange rate is necessary. The mechanism for the price oracle is the following:

- For each Terra currency, miners submit a vote for the current exchange rate in the target currency.
- Every n blocks, the system computes the weighted medians.
- An amount of Terra is given to those who voted within one standard deviation from the median. Miners who voted outside may be punished keeping their stakes.

The major issue with decentralized oracles is the risk that miners (that assume also the role of oracles) can take profit from coordinate false price vote. In order to avoid this risk the system limits the vote to a subset of miners, since for them a successful coordination on the price would result in a loss greater of their the staked Luna greater than the potential fraudulent profit. We remark that Luna stakes are time-locked for 28-days, so check on bad actions can be done with some delay.

Now let's get to the crucial point: how to achieve stability.
Since the market of Terra is open, it follows the rules of supply and demand:

[1]The SDR is an international reserve asset created by the IMF to supplement the official reserves of its member countries. The SDR is not a currency. It is a potential claim on the freely usable currencies of IMF members. As such, SDRs can provide a country with liquidity. A basket of currencies defines the SDR: the US dollar, Euro, Chinese Yuan, Japanese Yen, and the British Pound[1]

[2]"Atomic swaps are automatic exchange contracts that allow two parties to trade tokens, even from two different blockchains" from coindesk.com

[3]An Oracle is a third party service that capture data from the world, e.g. election result, weather, plane delay etc. in order to provide a smart contract information. In order to avoid manipulation and centralization is better to use a decentralized network of oracles.

- If the money supply contract, all other conditions held equal, then we'll have a higher relative currency price level.
- If the money supply expands, all other conditions held equal, then we'll have a lower relative currency price level.

Given those basic economic concepts, we should understand how is possible to modify the Terra and Luna supply. Firstly, we notice that the value of Terra comes from its capabilities of replicating and maintaining a stable price. For Luna its value comes from the whole ecosystem, since Luna is the native token, it is the "fuel" of the Terra blockchain: transaction fees are paid in Luna, staking is done with Luna, and several applications use Luna. So when Terra value drops then we need to transfer some value from Luna to Terra, and vice versa if the value of Luna increase we need to transfer some value from Terra to Luna. This kind of value transfer is possible thanks to the Terra algorithm that can create new Terra and Luna and control two levers: the transaction fees and the Luna burning rate.

We recall that Terra blockchain has a proof of stake consensus model, so in order to produce new blocks and get rewarded a miner need to "stake" an amount of Luna, more Luna staked mean more possibilities to be elected as the next block creator.

The plan for Terra is to let miners absorb Terra contraction costs through mining power dilution in the short term, so during a contraction, the system will mint and auction more mining power to buy back and burn Terra, so the supply of Terra will be contracted.
This means that if there is too much Terra in circulation, then more Luna will be printed, with this new Luna the system will buy back the excess of Terra from the markets. This operation is not good from the miner's point of view for two reasons:

- Producing new Luna will create an increased offer of Luna and so the price will drop, so the staked funds of miners will have a decreased value.
- Since there is more Luna available the probability of being elected as creator of the next block is smaller if you not increase the amount of the stake.

Briefly the miners absorb the costs of Terra volatility in the short term, and they'll be compensated for this in the long-term. This concept will be explained in detail in the next section.

3.1.1. Miners absorb short-term Terra volatility

The Terra blockchain has a Proof of Stake (PoS) mechanism, that means that miners need to stake the native cryptocurrency Luna in order to be able to create new blocks. To create a new block the system will elect a block producer from the set of miners, this block leader will produce the next block aggregating transactions and ensuring that messages are distributed in a short time frame with high fault tolerance. In order to elect the new block creator the system will take into account the amount of staked Luna. So we can consider Luna as a representative of the mining power in the Terra network.
Luna is also the immediate defense against fluctuations in Terra's price. Let's see what happens when Terra USD (UST)'s price move away from 1 USD:

- If UST's price goes below 1 USD, then user can send 1 UST to the system and receive 1 UST's worth of Luna.
- If UST's price goes above 1 USD, then user can send 1 USD's worth of Luna to the system and receive 1 UST.

The system will always respect the target exchange rate of 1/1 irrespectively of the market condition. So if the price in the secondary market move from the desired peg, then an arbitrageur can easily make risk-free money.
If the price of one UST is 0.9$ then we could buy it for 1 $ of Luna and sell this Luna immediately, gaining a risk-profit of 0.1 $.

With this in mind, we can consider the operations that grant price stability as a transfer of value from Luna to Terra and vice versa.
Luna is the key to maintain price stability:

- To buy 1 UST, the protocol mints and sells Luna worth 1 USD
- By selling 1 UST, the protocol earns Luna worth 1 USD

In this way volatility is moved from Terra price to Luna supply. We should consider that the increase in Luna supply will present a problem for the miners, since their staked Luna are worth a smaller portion of total available mining power post-contraction.

3.1.2. Miners long-term stable rewards

Miners provide stability and security to Terra, so it's fundamental that they are compensated with long-term stable rewards. A key feature of Terra network is to provide stable demand for mining, and to achieve this, the protocol offer stable and predictable rewards,

in all economic scenarios: booms and busts.
There are two ways of rewarding miners:

- Transaction fees: every Terra transaction pays a small fee to miners, the default is 0.1% of the transaction, and are capped at the lower between 1% of the transaction and 1 SDR.
- Seigniorage: if we have an increase in Terra demand, the system will mint Terra and then earns Luna. This is called seigniorage since the protocol is gaining value simply printing new Terra. After that, the system burns a portion of the earned Luna in order to make mining power scarcer. The remaining part of seigniorage goes to the Treasury (more about the Treasury will be explained in the Fiscal Policy section).

Let's look the situation from the miner's point of view, after fixed costs the profit (or loss) for a *single unit of mining power* (1 Luna) is given by rewards minus the cost of the work for that unit, formally:

$$P(t) = \frac{TotalRewards(t)}{LunaSupply(t)} - UnitMiningCost(t)$$

We want this value to be positive and predictable. Most of the uncertainty lies in the first term also called *unit mining rewards*, since stable unit mining rewards produce a stable demand for mining. Both rewards and Luna supply are uncertain, the rewards come from fees on transactions, so rewards increase when the economy grows. And Luna supply tends to decrease when the economy shrinks, since new Luna are issued to buy back Terra. So unit mining rewards move in the same direction of the economy.

The system has two possibilities to maintain a stable mining demand in the long-term: changing the transaction fees and/or the rate of Luna burn, in order to oppose changes in unit mining rewards.
The idea is:

- If unit mining rewards are increasing:
  - Decrease fees.
  - Decrease Luna burn rate.
- If unit mining rewards are decreasing:
  - Increase fees.
  - Increase Luna burn rate.

Fees and the rate of Luna burn are adjusted every week. Now we define:

- f_t is the transaction fee;
- b_t is the Luna burn (what percentage of seigniorage does the protocol use to buy back and burn Luna);
- R_t the unit mining rewards, all of them at time t.

The system that define the rule for adjusting the values of f and b is:

$$\begin{aligned} f_{t+1} &= (1 + g_f)\frac{R_{t-1}}{R_t} f_t, \\ b_{t+1} &= (1 + g_b)\frac{R_{t-1}}{R_t} b_t. \end{aligned} \tag{3.1}$$

For example, if unit mining rewards were cut in half, then the fees would double in response, conversely, if unit mining rewards were to double fees would be cut in half as a response. The result is scaled by a factor $1 + g_f$ for fee and $1 + g_b$ for burn rate, that permits gradual growth.

3.2. Growth-driven Fiscal Policy

With high-volatile cryptocurrencies also the related smart contracts can be useless, since the payouts will be denominated in a volatile asset. In the Terra ecosystem, the presence of stablecoins could unlock huge potential given by smart contracts. So other than the stability mechanism Terra offers a stable dApp (decentralized application) platform oriented to building financial applications that use Terra stablecoins as underlying. And these Terra dApps will help to bring growth and stability to the Terra ecosystem. We're going to explore how the Treasury implements Terra's fiscal spending policy. The focus of the Treasury is to allocate resources derived from seigniorage to dApp. In order to receive resources a dApp needs to register as an entity that operates on the Terra network. The funding procedure is the following:

- dApp applies for an account with the Treasury, here we have several metadata in order to correctly identify the dApp.
- Luna validators vote on regular basis to accept or reject new dApp applications. (There is a quorum of 1/3 in order to approve a dApp).
- For every voting session, Luna validators have the right to request that a dApp is blacklisted (even on this occasion there is the quorum of 1/3).

The funding of a dApp is determined by the validators voting in each funding period in accordance with a weight that is assigned to each dApp. The weights are done in order to maximize the impact of the stimulus on the economy, there are two criteria to determine spending allocations: robust economic activity and efficient use of funding. We consider these two factors into a single weight w_t given by the formula below, so we define:

- TV_t is the dApp's transaction volume.
- F_t is the Treasury funding received.

The notation $*$ denotes moving average. The funding weight w_t is:

$$w_t = (1 - \lambda)TV_t^* + \lambda \frac{\Delta TV_t^*}{F_{t-1}^*}$$

The first term is proportional to TV_t^*, so this term describes the economic activity. The second term is proportional to $\Delta TV_t^*/F_{t-1}^*$, with the numerator we describe the trend in transaction volume, the denominator is the average funding amount received in the last funding period, so with this term in general we are describing how the economic activity is changing with respect to the past funding. The parameter λ is used to balance between the two terms.

In this chapter we have described the key elements of the Terra ecosystem, in particular we have gone through:

- How the stability mechanism work.
- Key factors like transaction fee and Luna burn rate.
- How the seigniorage is used for the stability mechanism and for promoting dApp.

In the next chapter, we will develop a model to describe the Terra ecosystem and we will run some simulations in different conditions, in order to stress test the system and check if the peg to the fiat currency will last under adverse market conditions.

4 | Terra stability stress test

In this chapter we're going to develop a model and simulate some aspects of the Terra protocol, in particular one model for Terra transactions/demand and another one for the Luna price. This model will help us to see how the protocol behaves in adverse market condition, in particular, we will try to figure our if the two "levers" of transaction fees and burning rate are enough to stabilize the price of Terra. In order to do that we will run a baseline scenario where we will define some risk threshold and we will stress some parameters in order to simulate several possible scenarios and then check if the above thresholds are broken. This will constitute a stability stress test for Terra.
Most of this work will take inspiration from the Stability Stress Test run by Nicholas Platias and Marco Di Maggio in [16].

The model will have discrete time-steps, where at each time-step we determine whether Terra demand has increased or decreased. Then we determine the current market price for Luna thanks to our model. Then we have two options:

- Issue new Terra if the demand of Terra has increased. Sell those Terra for Luna, and then burn Luna according to the current burn rate. Deposit in the Treasure the remaining Luna.
- Buy back excess of Terra supply and burn them, by issuing new Luna if Terra demand has decreased.

4.1. Methodology

To simulate the Terra ecosystem we will go through the following four steps:

1. Determine the Terra demand at time t, and check if it is increased or decreased (price above or below the peg).
2. Determine the market price for Luna.
3. Now we have two possibilities:

- If Terra demand has increased. The system issues new Terra and sells them in exchange of Luna at current market price. From the Luna earned some will be burned at the current burn rate. The remaining Luna will be sent to the Treasury.
- If Terra demand has decreased. Then the system will buy back Terra by issuing new Luna at current market price.

4. The system determines the unit mining rewards for the current period and changes the fee and the burn rate, according to the new market conditions for the next time step.

4.2. Model of Terra transaction

In this section we will model the Terra demand. Since the main driver for fluctuations in Terra is the change in transaction volume, we model the Terra demand as a stochastic process represented by the transaction volume. The transaction volume can be seen as a gross domestic product (GDP) for an economy. So the Terra's GDP should represent a broad spectrum of scenarios that we can use to stress test the stability mechanism. We consider two possibilities in the Terra economy:

- Macro volatility: due to shift in the underlying economy. For example, in a recession scenario the transaction volume is going to decrease.
- Micro volatility: since the demand might be subject to shocks in each business cycle, due to idiosyncratic shocks.

There is another factor of uncertainty needed: we don't know when the shock will be realized and when the economy will switch business cycle.
In order to group all these factors we consider firstly a Markov-switching process with two states: boom (to represent an upturn) and bust (to represent a downturn).
To model the Terra demand M_t we use a Geometric Brownian Motion, so we have the following stochastic differential equation (SDE):

$$dM_t = \mu_s M_t dt + \sigma M_t dW_t$$

- W_t is a Standard Brownian Motion.
- μ_s is the drift in the state $s \in boom, bust$.

- σ represents the volatility.

The solution of this SDE is:

$$M_t = M_0 \exp\left(\mu_s - \frac{1}{2}\sigma^2\right)t + \sigma W_t$$

This solution is valid for $t > 0$, where M_0 is the Terra demand at the beginning time $t = 0$.
Since we have two possible economical cycles we have two possible μ_s which are μ_{boom} and μ_{bust}. For boom cycle we will use a positive value since we want a growing process in average. The contrary for the bust cycle where we use a negative value, since we want a decreasing value in average.
To better understand this we should consider the average in the state s:

$$E_s[M_t] = M_0 e^{\mu_s t}$$

Now that we have described how the model behaves in the two cycles, we try to model the transition between them. We can represent the demand model as a Markov switching model where the Terra demand M_t follows the stochastic process:

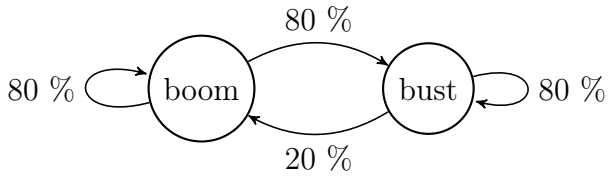
$$M_t = \begin{cases} \mu_{boom} M_t dt + \sigma M_t dW_t & \text{if } s_t = boom \\ \mu_{bust} M_t dt + \sigma M_t dW_t & \text{if } s_t = bust \end{cases}$$

The state s_t follows a Markov chain with the following transition matrix:

$$P = \begin{bmatrix} P(s_t = boom | s_{t-1} = boom) & P(s_t = bust | s_{t-1} = boom) \\ P(s_t = boom | s_{t-1} = bust) & P(s_t = bust | s_{t-1} = bust) \end{bmatrix}$$

$$= \begin{bmatrix} p_{boom,boom} & p_{bust,boom} \\ p_{boom,bust} & p_{bust,bust} \end{bmatrix}$$

in this matrix $P_{s_t,s_{t-1}}$ denotes the transition probabilities which govern the random behavior of the state variable. This Markov process is the solution that we will adopt in order to model business cycle fluctuations.
Let's have a look at a schema to better understand the Markov chain functioning:



The logic of this Markov chain is quite simple. The economy stays in one of two possible states, boom or bust. At each time step (e.g. every year) the economy can remain in the same state with probability 80% or it can change with probability 20%.
The corresponding transition matrix is:

$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix},$$

and playing with the parameters we can show several extreme scenarios in order to comprehend the two main drivers present in our model: stochastic business cycles and short term volatility.
Let's start with a weekly simulation of ten years, where we have $\mu_{boom} = 0.4$, $\mu_{bust} = -0.2$ and volatility $\sigma = 0.02$. We consider this as a scenario with very small volatility, so here we can observe how the business cycle works, to accentuate more this behaviour we exaggerate the values of the transition matrix:

$$P = \begin{bmatrix} 0.67 & 0.33 \\ 0.33 & 0.67 \end{bmatrix}.$$

So we run ten simulations and we get the result in figure 4.1. Notice that the y axis is logarithmic.

Let's now consider the same model but with ten times greater volatility $\sigma = 0.2$. And the following transition matrix:

$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}.$$

We can see the results of ten simulations of ten years in figure 4.2

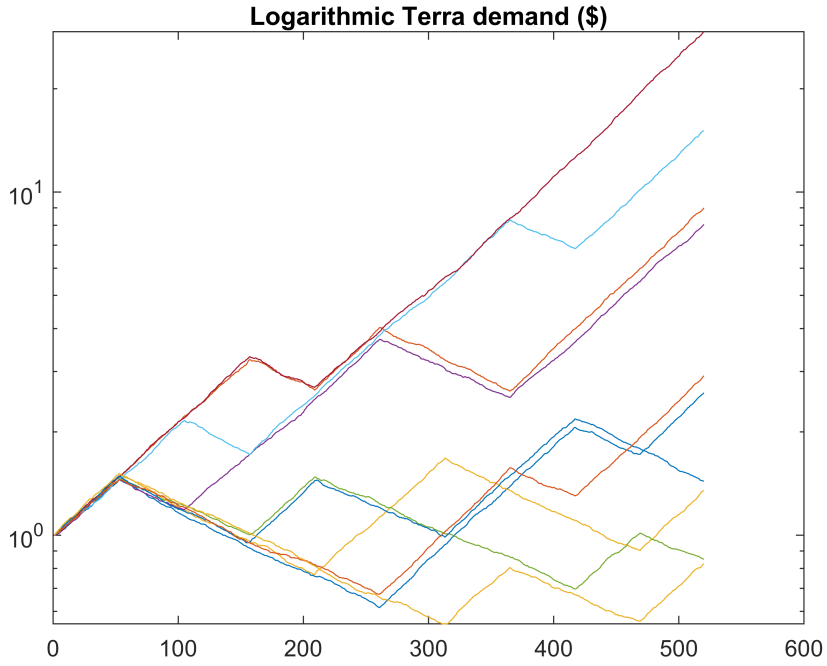


Figure 4.1: Logarithmic Terra demand, emphasis on Macro volatility

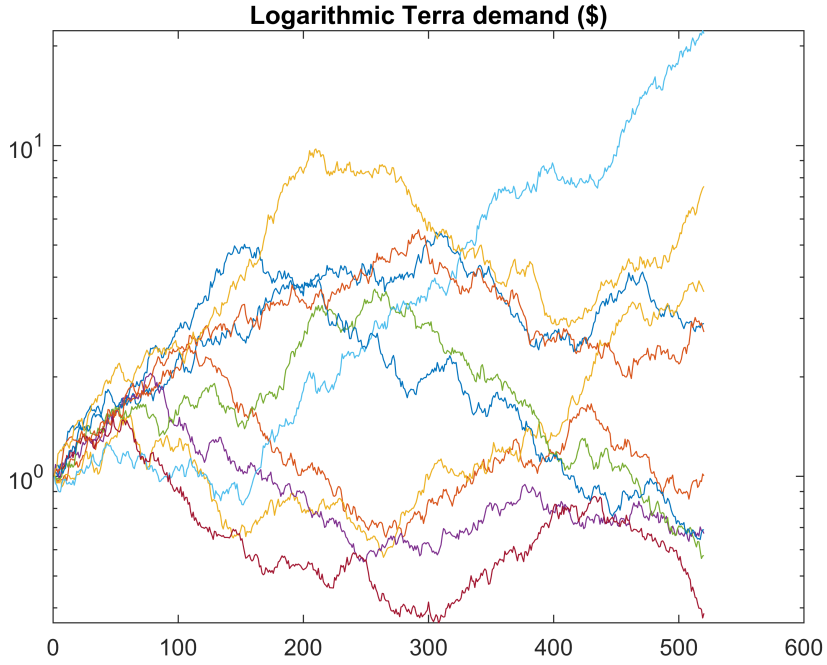


Figure 4.2: Logarithmic Terra demand, emphasis on Micro volatility

4.3. Model of Luna price

A model for pricing Luna should take into account several drivers:

- Expected rewards from fees.
- Fluctuations in the supply of Luna due to changes in Terra demand.
- And the uncertainty intrinsic in Luna.

We already have developed in chapter 4 something that could help us modeling the above list, which is the Profit/Loss of mining $P(t)$:

$$P(t) = \frac{TotalRewards(t)}{LunaSupply(t)} - UnitMiningCost(t)$$

where:

- TotalRewards(t) is the total amount of rewards in fiat currency by the mining process, namely the volume of transactions multiplied by the fee rate at time t.
- LunaSupply(t) is the amount of Luna available at time t.
- UnitMiningCost(t) is the unitary amount of cost per Luna for being a miner at time t.

As a first idea we can consider the price of a unit of Luna as the sum of all the present values of the future profits or losses minus the fixed costs:

$$p(t) = \sum_{t=0}^{\infty} \frac{P(t)}{(1+r)^t} - FixedMiningCost.$$

Using the profits or losses to determine the Luna price is a good idea since it takes into account the rewards and the Luna supply. From the intuitive point of view: Luna would trade at a premium if most of holders believe that Luna would grow fast. Luna would trade discounted if most of holders believe that Luna supply will grow fast.
In order to make the model easier, we remove the cost component (from the unit and from the fixed part), since mining costs vary much less than the remaining components of the model, and also because another feature of the Terra ecosystem is the possibility to delegate Luna to a miner in order to get a part of the fees rewards. From their point of view, there are no mining costs, but only a small discount to pay for the commission.

We consider the price now as:

$$p(t) = \sum_{t=0}^{\infty} \frac{UnitMiningRewards(t)}{(1+r)^t}$$

where

$$UnintMiningRewards(t) = \frac{TotalRewards(t)}{LunaSupply(t)}$$

Computing this sum explicitly is not easy, and another drawback is that it is not easy to define and identify an interest rate r with cryptocurrencies.
So we're doing the following approximation:

$$p(t) = UnitMiningRewards(t) \cdot RewardsMultiple(t)$$

How can we model the $RewardsMultiple$? This factor is moved by two distinct properties of unit mining rewards:

- Growth: higher growth of rewards, higher the rewards multiple (and vice versa)
- Volatility: higher volatility in rewards, lower rewards multiple (and vice versa)

To model the rewards multiple we choose a random walk, each value receives a premium for rewards growth and a discount for rewards volatility relative to the previous one.
The rewards at time t follows a normal distribution with means $\mu(t)$ that depends on the rewards multiple at time $t-1$:

$$RewardsMultiple(t) \sim N(\mu(t), \sigma^2)$$

we define the mean μ_t as:

$$\mu(t) = (1+g(t)) \cdot (1-v(t)) \cdot RewardsMultiple(t-1)$$

$g(t)$ and $v(t)$ are two functions that reflect the change in unit mining rewards and its volatility measured at time t. In fact, when $g(t) > 0$ we apply a premium to the rewards multiple and so to the price. On the other hand, when $v(t) > 0$ we apply a discount.
Let's now define the two functions:

$$g(t) = \left(\frac{ShortRewardAverage(t)}{LongRewardAverage(t)} - 1 \right) \cdot \alpha$$

and

$$v(t) = \left(\frac{ShortReturnVolatility(t)}{LongReturnVolatility(t)} - 1 \right) \cdot \beta$$

The averages are computed on unit mining rewards and volatility is computed based on quarterly logarithmic returns of unit mining rewards. As time frames we use one year for the short period and two years for the long period when computing moving averages.
α and β are two scaling factors between 1% and 5%, we use higher values of α relative to β, since growth usually has a larger effect on the rewards multiple than volatility.
The random walk is restricted to the interval [5,100] in order to make it robust against violent moves in either direction.

4.4. Alternative Model of Luna price

The previous model is the one used in the stress test of [16], and it has a strong economic base. But we would like to provide an alternative, in the previous model we had two sources of randomness, one for the demand and another one to "disturb" the umr in order to obtain the price of Luna. We keep the demand part, but we modify the Luna model since we want a more rigid stress test. In particular, we want to remove the possible correlation between the Terra demand and the Luna price, that is because Luna has grown a lot in terms of popularity and has become a cryptocurrency traded on several exchanges by users that are not necessarily interested in the related stablecoin or in the staking.

So in our model for the price of Luna we will use a new Geometric Brownian Motion, like the one described in the Terra demand, but with $\mu_{boom} = \mu_{bust}$, so we don't have the Markov chain and a cyclic nature.
This assumption is done since Luna is traded on several exchanges so it's used by many users as a speculative asset. This could impact the price behaviour, since most of the buyers/sellers do not use Luna for the staking rewards, but simply with the hope of a rise in price. As we said before, we want to perform a stricter stress test so we would like to eliminate any possible correlation between the Terra demand and the Luna price.

In this model the Luna price will have the following dynamics:

$$dL_t = \mu L_t dt + \sigma_{price} L_t d\tilde{W}_t$$

the solution will be like demand's case:

$$L_t = L_0 \exp\left(\mu - \frac{1}{2}\sigma^2_{price}\right)t + \sigma_{price}\tilde{W}_t$$

It is important to notice that the two brownian motions W_t and $\tilde{W}_t$ are independent, this means that $dW_t \cdot d\tilde{W}_t = 0$.

4.5. Stress Test

We need to define in which condition the peg of Terra could be broken, following the work of [16] we identify two conditions which in combination put the peg at risk of breaking:

- Large amount of Luna produced in a short amount of time
- Substantial drop in Unit Mining Rewards

Both conditions are necessary, if we have only a rapid increase the supply of Luna there is no problem since the Unit Mining Rewards are stable and miners are properly compensated for the dilution of their mining power, briefly even if their Luna have less value from the mining point of view the reward is greater. A decrease in unit mining rewards creates no serious risk if Luna supply does not increase in an unstable way. But if both events happen together, then a death spiral[1] could happen. Now we need to define the above mentioned threshold for Luna supply and of Unit mining rewards.

4.5.1. Baseline scenario

In order to find a proper threshold we need to identify what is a "large amount" and a "substantial decrease". Even this time we refer to [16], where those quantities are identified as "% increase in Luna supply and % decrease in unit mining rewards over any 13 week period (quarterly changes)."
Quarterly changes are chosen because it is hard to establish reliable thresholds at higher frequency due to the noisier nature of the quantities considered. In order to define the risk thresholds, we use the 1% Value at Risk (VaR) for each quantity over 1'000 simulation in a "baseline stress scenario". After those simulations, we define two thresholds:

$$VaR_{supply} = 20\%$$
$$VaR_{umr} = -30\%$$

[1]With "death spiral" we define an event where the system collapses, in particular when some unexpected conditions happen and the algorithm that should maintain stability does not help anymore maintaining the peg, but instead it helps the de-peg. In the Terra ecosystem this happens when both increase in Luna supply and decrease in umr occur.

In figure 4.3 we can observe two histograms for the umr and the Luna supply, and a red line that highlights the VaR:

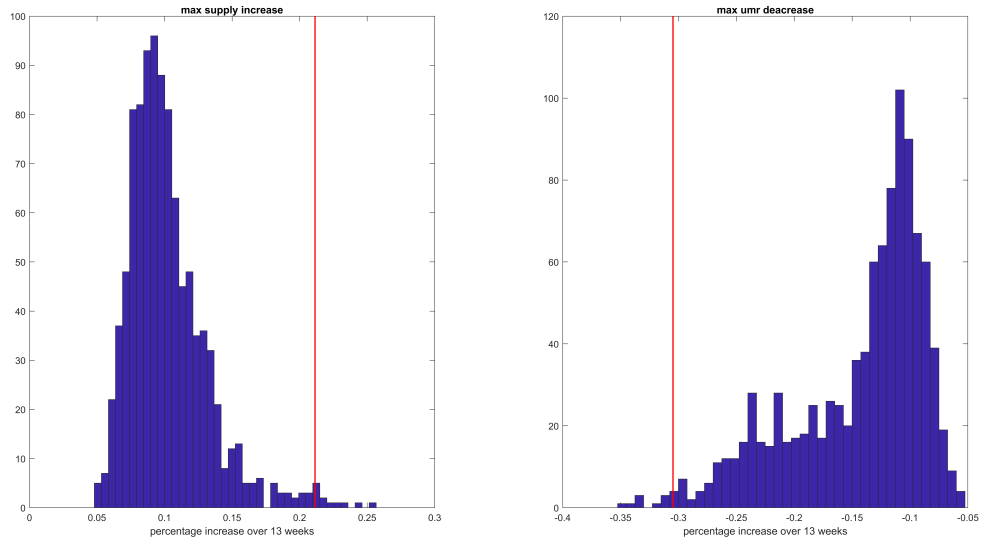


Figure 4.3: Histogram of the 1000 simulations

We should also check a scatter plot of the two quantities, in figure 4.4 we can observe that there is not a single simulation where both the umr and the supply thresholds are broken together, so in this 1'000 simulation there is no de-peg event according to this model.

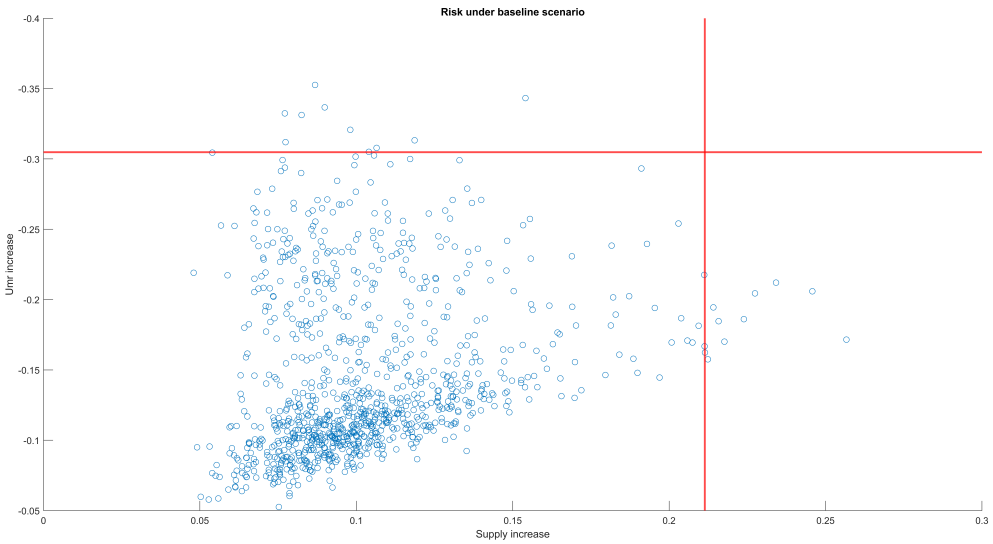


Figure 4.4: Scatter plot of the 1000 simulations (baseline scenario)

Now that we have defined the two quantities and the two thresholds that are responsible for the de-peg, we can run our stress test.

4.5.2. Stress Test

We proceed in the same manner described in [16] but using the Alternative Price Model. In [16] they define two stress variables of the model:

- cyclicality
- volatility

Cyclicality is defined as the difference in severity between boom and bust cycles in the Terra demand model. In particular, we define

$$\mu_{bust} = -\frac{1}{2}\mu_{boom}$$

and the cyclicality is defined as:

$$cyclicality = |\mu_{boom}| - |\mu_{bust}|.$$

The volatility variable is simply the volatility parameter σ of the geometric brownian motion that governs each cycle in the Terra demand model.
With our assumption we found that the cyclicality model is not so relevant since there is not a significant variation in the risk of de-peg. We prefer to use another stress variable which is the volatility of the geometric brownian motion of the Luna price in the Alternative Luna price model, since in that way we can better stress the market fluctuations of the Luna's price.

To perform the stress we create a 10× 10 matrix, where each cell contains a couple of values, one for the volatility of the demand (from 0.2 to 0.4) and the other for the volatility of the Luna price (from 0.2 to 0.65).

For each cell of the matrix we run 1'000 simulation of 10 years each. The stress covers one million years of simulations around several possible market conditions.

The other parameters are the ones provided in [16].

Method The method for a single simulation is the following:

- Simulate ten years of demand
- Simulate ten years of Luna price
- Calculate $\Delta = Demand_t - Demand_{t-1}$
  - If $\Delta > 0$ then $Supply_t = Supply_{t-1} - \frac{\Delta}{price_{t-1}} \cdot burn_t$
  - If $\delta < 0$ then $Supply_t = Supply_{t-1} - \frac{\Delta}{price_{t-1}}$
- Compute Total Rewards: $Rewards_t = Demand_t * fee_{t-1}$
- Compute UMR: $umr_t = \frac{Rewards_t}{Supply_t}$
- Compute the new fee and burn rate (description in chapter three).

After that process, we have a 10 year evolution simulation. We then extract the max umr decrease and the max Luna supply increase over 13 weeks.
The umr max decrease is done with respect to its moving average over 4 weeks since if a miner puts some Luna at stake and want to get the stake back there is a waiting time of 4 weeks, so for the miner is more important the average over 4 weeks rather than the spot value.

After our 100'000 simulations for each cell of the matrix we can count how many times the two thresholds are broken together. If we divide those numbers by 1'000 we can have an estimate of the probability of de-peg and death spiral.

We should also consider that in our 1'000 simulations the maximum of Luna supply increases and the minimum in umr decreases are not necessarily happening at the same time, so our de-peg probability is over-estimated in order to make this stress test more robust.

4.5.3. Results

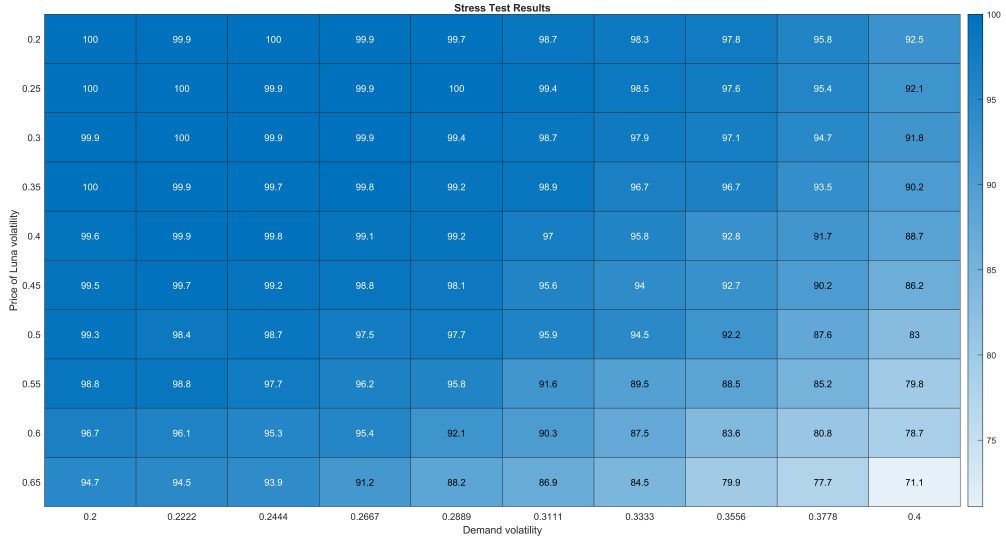


Figure 4.5: Terra Peg Survival

In figure 4.5 we have a heatmap of the before mentioned matrix, here we can observe how the probability of the de-peg vary (dark blue means lower probability of de-deg, and light blue means higher probability of de-peg).
As stated in [16] this kind of stress test take inspiration from the stress tests conducted by the Federal Reserve Board with the original scope of "helping to ensure that large financial institutions will remain solvent in severe recession".

Thanks to our 10×10 matrix we have done one thousand simulation for one hundred possible combinations of market conditions. We will pay attention in particular to the ones with large volatility both in demand and in Luna price.

On the x axis, we have the variation in the volatility of the demand and on the y axis we have the variation on the volatility of the Luna price. As expected, when both risk factors increase then also the probability of breaking the peg increases. A single stress factor alone is not sufficient to pose a serious stability risk, either in high price vol/low demand vol and in high demand vol/low price vol the peg survival probability stays above 92 %. And in the worst case the probability of de-peg stays above 70 %.

5 | Conclusions and future developments

5.0.1. Conclusions

The main objective of this thesis is the study of the stablecoin.

In this work we have given a brief description of some fundamental aspects of the blockchain technology, even from an historical point of view. This is done in order to have the right tools to comprehend the rest of the work.
We then analyzed several aspects of the stablecoins:

- How is it possible to classify them?
- The available stability mechanisms, and in particular we have described in details how the Terra ecosystem works.
- We developed a mathematical model for the Terra ecosystem and performed a stress test in order to check its stability mechanisms.

Classification and stability mechanisms The stablecoin field is very young and it is evolving at a very fast pace. As consequence several classifications came out, and we can conclude that there is not available a single and always correct one. It always depends from the point of view of which is categorizing the various available stablecoins. The ECB tends to classify stablecoins based on the fact that behind them there are some fiat currency or not. The research team of the MIT tends to classify the stablecoins based on the necessity of a legal system in order to have a working stablecoin.
There are several methods available in order to obtain a stable cryptocurrency. Simpler method like creating a token for every unit of fiat currency in a custodial account, so for example we create a certain amount n of tokens and then we deposit n unit of fiat currency in a bank account, this method is the first one ever used to create a stablecoin. Moreover, we have a trickier method like implementing a smart contract that acts as the central bank with other cryptocurrencies as a form of reserves.

The core of the work of this thesis has been dedicated to the stability mechanisms. Fundamentally there are three possibilities to obtain a stablecoin: from fiat currency, from other asset or cryptoasset as collateral (mainly as over-collateral) and purely algorithmic stablecoin (peg is obtained via some open market operations on decentralized crypto-exchanges).
During the years simpler method appeared to be more stable, but with recent developments of algorithmic methods, we can say that even the more sophisticated ones, without the direct use of some collateral, have given proof of strong stability even in a really bad and unstable market condition.

Stress Test We explained a model for the Terra's variables, in particular two stochastic models: one for Terra demand and one for the price of Luna in USD. We provided also a framework to quantify the risk of losing the peg. Then we defined risk thresholds for two key variables which are: Luna supply increase (percentage difference over 13 weeks of the amount of total Luna available) and unit mining rewards decrease (percentage difference over 13 weeks of a measure for the rewards for the miners). We apply two forms of stress in order to test Terra's stability: the volatility of the demand and the volatility of the price of Luna, instead of the original stress test paper where the stability is tested with the volatility of the demand and the cyclicality (difference between the two drifts of the geometric brownian motion of the Terra demand in the two states of the Markov chain). If we compare this stress test and the one done by [16] we can conclude that we come up with the same findings: based on one hundred different stress scenarios (implying one million years' worth of simulations) we can say with high confidence that only one stress factor taken alone is not enough to provide significant risk of de-peg. And the only potential risk comes from a union of both stress factors at the same time in a context of extreme volatility. So even in the test scenario that we have created, where there is no potential correlation between the demand and the price of Luna we conclude that the Terra ecosystem is resilient to huge volatility in the demand of Terra and in the price of Luna.

5.0.2. Future developments

Several future developments are available, for the classification part we should take in consideration that new ways to obtain stablecoins arise day by day. So the classification methods could change consequently. Working with stablecoin require active monitoring on the new available mechanism and possible classifications. The presence of new possible

stability mechanism does not pose a threat to older ones, for example Tether is still around even if new algorithmic stablecoins are available.
More can be done from the stress test point of view. We choose to use three random sources:

- The Markov chain for the demand process.
- The Geometric Brownian Motion for the demand process.
- The Geometric Brownian Motion for the price of Luna.

And in our stress test we have changed only σ_{demand} and σ_{price} of the two GBMs. There is the possibility to stress also the parameters of the Markov chain. In order to model different kinds of macroeconomic cycles.
The various parameters used in our simulation like the drift of the demand, the drift of the price, the transition matrix, the starting points of the fee and burn rate are take directly from the original stress test paper. This paper was published in May 2019, so almost three years passed. The data available right now is quite enough to use some statistical tool in order to better estimate those parameters. For example an interesting option could be found if the duration of a business cycle of one year is a correct assumption, and if it is correct, it would be interesting to find also an estimation for the transition matrix of the Markov chain used to simulate business cycles.

Bibliography

[1] International monetary fund - special drawing rights. URL https://www.imf.org/en/Topics/special-drawing-right.

[2] M. T. Adrian and M. T. M. Griffoli. *The rise of digital money*. International Monetary Fund, 2019.

[3] A. Back et al. Hashcash-a denial of service counter-measure. 2002.

[4] Blockchain and D. A. P. W. U. of Pennsylvania. Defi beyond the hype - the emerging world of decentralized finance. URL https://wifpr.wharton.upenn.edu/wp-content/uploads/2021/05/DeFi-Beyond-the-Hype.pdf.

[5] D. Bullmann, J. Klemm, and A. Pinna. In search for stability in crypto-assets: are stablecoins the solution? *ECB Occasional Paper*, (230), 2019.

[6] V. Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 3(37), 2014.

[7] M. Egorov. Automatic market-making with dynamic peg. URL https://curve.fi/files/crypto-pools-paper.pdf.

[8] M. D. M. N. P. Evan Kereiakes, Do Kwon. Terra money: Stability and adoption. URL https://assets.website-files.com/611153e7af981472d8da199c/618b02d13e938ae1f8ad1e45_Terra_White_paper.pdf.

[9] E. Force et al. Stablecoins: Implications for monetary policy, financial stability, market infrastructure and payments, and banking supervision in the euro area. Technical report, European Central Bank, 2020.

[10] S. Haber and W. S. Stornetta. How to time-stamp a digital document. In *Conference on the Theory and Application of Cryptography*, pages 437–455. Springer, 1990.

[11] L. X. Lin. Deconstructing decentralized exchanges. *Stan. J. Blockchain L. & Pol'y*, 2:1, 2019.

[12] A. Lipton, A. Sardon, F. Schär, and C. Schüpbach. 11. stablecoins, digital currency, and the future of money. In *Building the New Economy*. PubPub, 2020.

[13] MakerDAO. The maker protocol: Makerdao's multi-collateral dai (mcd) system. URL `https://makerdao.com/en/whitepaper/`.

[14] R. C. Merkle. A digital signature based on a conventional encryption function. In *Conference on the theory and application of cryptographic techniques*, pages 369–378. Springer, 1987.

[15] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[16] M. D. M. Nicholas Platias. Terra money:stability stress test. URL `https://agora.terra.money/t/stability-stress-test/55`.

[17] L. Oliveira, L. Zavolokina, I. Bauer, and G. Schwabe. To token or not to token: Tools for understanding blockchain tokens. 2018.

[18] G. M. Pettine. Cryptographic tokens : analysis and classification with a focus on a market index for cryptocurrencies. 2020.

[19] F. Saleh. Blockchain without waste: Proof-of-stake. *The Review of financial studies*, 34(3):1156–1190, 2021.

[20] J. Silverman. Is tether just a scam to enrich bitcoin investors? URL `https://newrepublic.com/article/160905/tether-cryptocurrency-scam-enrich-bitcoin-investors`.

[21] N. Szabo. Smart contracts. URL `https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html`.

[22] Tether. Tether: Fiat currencies on the bitcoin blockchain. URL `https://tether.to/es/whitepaper`.