Diana Elena Ion

Promo 2022 – Master EIT Digital en Sécurité Numérique

AIT Austrian Institute of Technology GmbH

Vienna, Austria

01.03.2021 - 31.08.2021

Master Thesis

Decentralized Finance Analysis

Superviseur EURECOM: Antonio Faonio Dr. Assistant Professor

Superviseur entreprise: Bernhard Haslhofer Dr. Univ. Lecturer

Rapport de stage confidentiel / Confidential thesis report
OUI / YES ☐ *NON / NO* ☒

EURECOM

Abstract

The multitude of Decentralized Finance (DeFi) protocols that appeared in the last couple of years has brought a wide range of financial products, with new protocols building on the previous ones by either integrating them or simply forking the available open-source code and start developing on top of it. This creates a highly interconnected ecosystem, with many inter-dependent parts, similar to Lego pieces. This paper tries to get a glimpse inside this ecosystem and see how different protocols are composed and how they might interact with each other. DeFi composability might also present risks, this is however out of scope for this paper. The proposed analysis represents only the first phase of a much larger study, and it aims at gaining an initial understanding of the involved protocols, establishing a methodology for data collection and processing, building and analysing a small-scale network of smart contract interactions. The preliminary results were consistent with the high-level observations regarding the composability, as well as being in line with previous studies on Ethereum network measurements.

Résumé

La multitude des protocoles de finance décentralisée (DeFi) implémentés au cours de ces dernières années apportait des divers applications financières avec des nouveaux protocoles implémentés basés sur l'intégration des anciens protocoles ou la manipulation du code source des anciens protocoles et le développement sur la base de ces derniers. Ceci créait un écosystème fortement connecté, avec plusieurs parties inter-dépendantes, similaires aux pièces de Lego. A travers ce papier, nous essayons de donner une idée sur cet écosystème et voir la composition de différents protocoles et leurs manière d'interagir entre eux. Il est à noter que la composabilité de DeFi peut aussi présenter des risques, cependant ceci ne fait pas partie du focus de ce papier. L'analyse proposée représente seulement une première phase d'une large étude, et a pour but d'avoir une compréhension initiale des protocoles impliqués à travers l'établissement d'une méthodologie pour la collection des données et leurs traitement, en construisant et en analysant un petit réseau d'envergure limitée d'interactions de "smart contracts" . Les résultats préliminaires étaient consistants, mais avec une vision superficielle à l'égard du facteur de composablitié. En effet, les résultats s'alignaient avec les études précédents sur les mesures des réseaux Ethereum.

Contents

Chapter 1

Introduction

A decentralized solution for digital currency was an issue many have tried to tackle for decades, however, it was not until 2009 when Bitcoin was introduced by the now infamous Satoshi Nakamoto. This mysterious character whose online presence lasted only three years, as he abruptly disappeared on April 23 2011, gave us a technology whose full potential has yet to be discovered, as many agree more than 10 years after Bitcoin's launch.

Ethereum built upon Bitcoin's key features and introduced a new paradigm in the blockchain space. The most important innovation of Ethereum is the Ethereum Virtual Machine (EVM), which runs smart contracts. These pieces of code can embed existing and new business logic into the blockchain, and new applications have emerged.

Not many technologies have seen a similar level of hype as blockchain has. According to Gartner's Blockchain Spectrum [34], four phases of blockchain solutions are identified, together with five key characteristics of blockchain: distribution, encryption, immutability, tokenization and decentralization. The first phase, blockchain-enabling, focuses on creating the technologies on which blockchain builds upon. Cryptography, peer-to-peer networks, etc. are some of the building blocks. Blockchain-inspired solutions represent the second phase, where only three of the five elements are used, with tokenization and decentralization being considered not mature enough. According to the timeline envisioned by Gartner, this is the phase we are currently in. The next one, blockchain-complete solutions, it is predicted to begin around 2023, with applications that use all five key elements. Finally, the last phase is about blockchain-enhanced solutions that will use other breakthrough technologies as well, combinaing blockchain with the Internet of Things or artificial intelligence, for example.

Decentralized finance, the main topic of this work, consisted initially of a few isolated protocols, but quickly grew to be the most prominent area for blockchain applications. DeFi tries to escape the complex and intricate traditional financial system, where the lack of transparency and interoperability, as well as high costs, are major sore points. One would argue that DeFi represents already the transition from blockchain-inspired to blockchain-complete applications. However, in terms of adoption, there is still a long way to go until DeFi sees the same level of users as centralized financial institutions.

Even though the DeFi ecosystem is relatively new, there have already been a number of important studies. Harvey et al. [19] provide a comprehensive study of the DeFi ecosystem, its components, building blocks and benefits, as well as its risks. Amler et al. [3] give an

overview of the existing DeFi products and analyze its advantages over traditional finance. Even though this part will not be discussed in this paper, DeFi presents a range of risks and vulnerabilities. There have been studies focusing on describing the attacks which have already taken place or describe hypothetical vulnerable situation. Gudgeon et al. [17] focus on how weak design implementations can affect the DeFi landscape. Another range of attacks such as frontrunning, stemming from the creation of highly specialized bots, is explored by Daian et al. in [12]. In terms of network analysis, Lee et al. explore different interactions in the Ethereum blockchain in [23], of particular interest for this paper being the analysis and measurements of the contract-to-contract network.

1.1 Aims

The main research question this paper aims to answer is how are DeFi protocols interconnected. The relationships between them are important in understanding the dynamics and risks within this ecosystem. Having the Lego concept in mind, we want to see to what extent these financial applications do rely on one another. We already know of services such as oracles that are being used by many protocols, however, we want to explore the more complex interactions taking place in DeFi. Trying to visualize a small-scale representation of the Ether flow between DeFi smart contracts and quantitatively characterizing the network with some key network metrics will constitute the analysis put forward in this work.

1.2 Structure

The paper begins with the chapter 'The path to DeFi' which explores the elements that led to DeFi, starting from the first forms of money to the Ethereum platform. The shortcomings of the centralized financial systems are analyzed, to establish the motives behind the emerge of DeFi applications. Next, the blockchain technology is described, with a focus on Ethereum and its building blocks. The next chapter, 'Decentralized Finance', analyses the elements, both in terms of infrastructure and financial primitives, upon which DeFi is built. Some key moments in the history of DeFi are highlighted as well. The last two chapters are concerned with the actual network analysis, limitations, future work and results.

Chapter 2

The path to DeFi

2.1 Money

The fuel of the world's economy, the root of all financial systems, money has evolved throughout history, under many forms, travelling the world at unimaginable speeds and conquering the world. Whether people like to admit it or not, our entire lives revolve around money. Different types of assets were categorized as early forms of money because they were used as a medium of exchange in economic transactions. Before the creation of this common medium of exchange, people were simply trading goods based on matched needs. This barter system, "as old as the man itself" [13] had the advantage of a real exchange of value for both parties, not shells, tokens or promises, but actual, tangible goods.

Functions of money

There are three main functions currencies need to fulfill. The medium of exchange feature was already mentioned, the other two are: unit of account and store of value.

- **medium of exchange** - different countries can have different sovereign currencies which act as the main medium of exchange in the respective territory. The introduction of currencies as intermediaries between the commodities or services traded by people made commerce a lot more efficient than in the barter system period. There was a natural evolution of the things people regarded as currencies along history such that we can now articulate a set of properties we use to determine a good medium of exchange. These features are: durability, transportability, divisibility, fungibility and non-counterfeitability.
- **unit of account** - the currency is used in denoting the price of other products and services. Measuring the value of all the other goods and economic activities in terms of the same unit makes accounting easier. It is crucial for the unit of account to have a good stability. National currencies tend to lose value with time due to inflation and it is the responsibility of national central banks to maintain the monetary stability.
- **store of value** - a good store of value enable people to accumulate wealth over time when holding on to the currency or asset. It is impossible to predict with certainty the

future value of the accumulated assets, be it gold or fiat currency, therefore, there is no perfect store of value. Anything for which there are expectations of stable future supply and demand can act as a store of value, from gold or diamonds to stocks, bonds or real estate.

2.2 Centralized Financial Systems

Managing money became a complex task over time and the need for specialized entities brought us to our current financial system. An intricate maze of institutions ranging from central banks to insurance companies which, essentially, provide access to financial instruments for their customers. Financial systems are responsible for creating links between people with available funds and the ones in need of investment. Acting as intermediaries, financial entities stimulate economic activities and ensure the circulation of the current money supply [36].

Even though the form of money people use today, fiat currencies issued and backed by national governments, has changed over time, the infrastructure of the financial system has mostly seen incremental innovations which aimed to reduce cost and friction in the existing systems. A high number of inefficiencies in the current systems arise because of the intermediation provided by financial institutions. The middleman character introduces new costs for all involved parties.

Harvey et al. [19] identify five major problems of centralized financial systems:

1. **centralized control**
2. **limited access**
3. **inefficiency**
4. **lack of interoperability**
5. **opacity**

In many countries the financial landscape is dominated by a couple of major players. Martin Schmalz shows in his Harvard Business Review article [39] how the same asset management firms are found among the top 5 shareholders of the top 6 largest US banks. This concentration of power in the hands of a few harms competition. The strong players can agree on high or low prices for certain financial instruments in the name of profit. As many people usually interact with only one bank for all their financial needs, and considering the complicated process of moving assets from one bank to another, they might find themselves trapped. The centralization does not manifest itself only at shareholders level, we see a large concentration of assets held by the top banks in both the United States and the United Kingdom. Figure 2.1 shows the assets of the five largest banks as percentage of the total commercial banking assets in the US[1]. For the UK[2], the situation is even more centralized, figure 2.2 displays the asset share of the top three banks.

[1] ttps://fred.stlouisfed.org/series/DDOI06USA156NWDB, July 13, 2021

[2] https://fred.stlouisfed.org/series/DDOI01GBA156NWDB, July 13, 2021

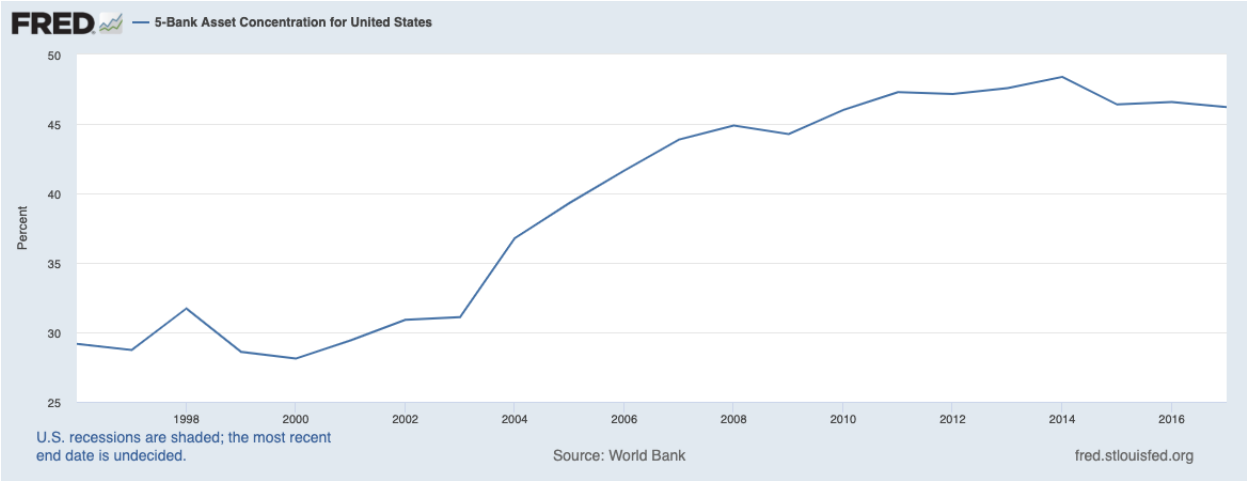


Figure 2.1. World Bank, 5-Bank Asset Concentration for United States [DDOI06USA156NWDB], retrieved from FRED, Federal Reserve Bank of St. Louis

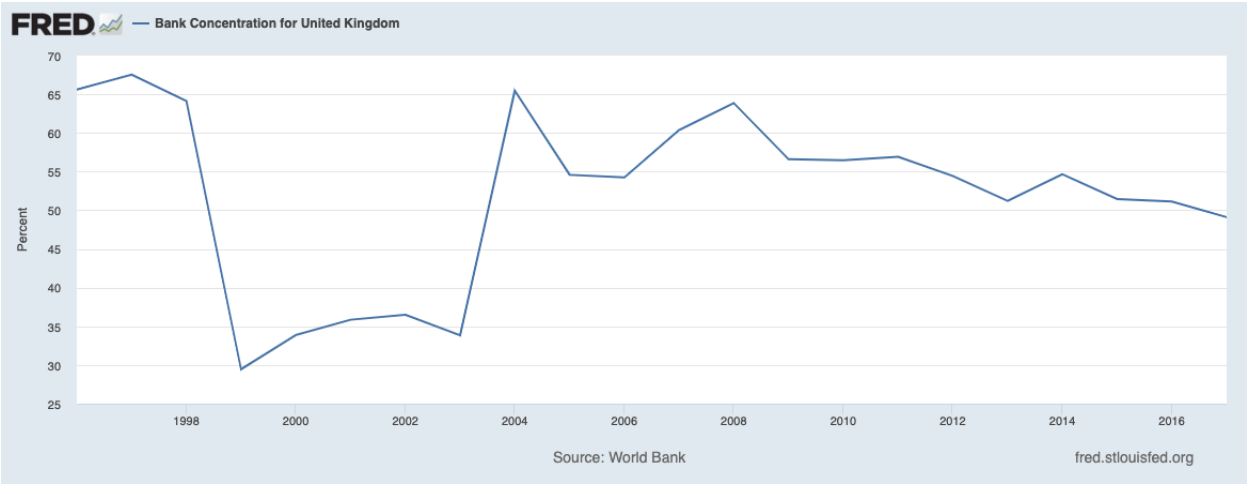


Figure 2.2. World Bank, Bank Concentration for United Kingdom [DDOI01GBA156NWDB], retrieved from FRED, Federal Reserve Bank of St. Louis

The scale of limited access to financial services in the Gloabl Findex database [14] is alarming. There are still 1.7 billion unbanked people around the world. This figure improved from the 2 billion found in 2014. However, we are still talking about a huge population segment who cannot participate in the global economy, access loans or insurance. According to the report, among the cited reasons for not having a bank account was the lack of sufficient funds, cost and distance and lack of documentation and trust in the financial system.

Financial institutions act as middlemen and are in charge of verifying the claims and identities of all the parties they do business with. The attestation process can be tedious

and complicating, with the middleman always charging a high fee for its services. One significant inefficiency can be observed in the stock market where trades take at least two business days to settle[3]. High transactions fees, lack of security, impossibility of performing microtransactions [19] are other examples of hurdles in the current system.

The lack of interoperability is probably the most common issue bank customers face on a regular basis. International transfers can still take between 1 and 4 business days to complete[4]. Moreover, as it was mentioned previously, changing the bank one has used for a long time is complicated as each bank has its own ledger. Small improvements have been implemented, however, a disruptive innovation is needed to drastically change the infrastructure of the system.

Finally, transparency is hard to achieve in such a complex and intricate system. Finding the best interest rate for taking a loan has to be done by the customer or by a specialized third party, banks know it is not in their best interest to admit to their customers that other banks offer a better interest rate. As such, people who are always using the same bank will miss out on opportunities to lower their costs.

2.3 Blockchain history

Every tale about blockchain starts by referencing the Bitcoin Protocol [31] as the stepping stone of everything that followed in this space. Without downsizing Nakamoto's contribution, his innovation built upon decades of research in cryptography, digital cash and distributed systems, to just name a few.

These roots can be traced back to the 1980s when the personal computers started to reach a wider audience. During this time a new movement called the 'cypherpunk movement' was born. As envisioned in the 1993 manifesto [27], cypherpunks were aiming for anonymous systems and electronic money. A pioneer of the movement towards electronic money was the cryptographer David Chaum who, in 1983, published a paper called 'Blind Signatures for Untraceable Payments' [8] proposing a method for anonymous payments. He also tried to bring digital money to the public when he founded DigiCash[5] in 1989 and partnered with several banks in trying to speed up the adoption. Unfortunately, the project failed after almost 10 years, but his legacy continued to live on in the crypto culture.

In parallel with David Chaum's endeavours, Stuart Haber and W. Scott Stornetta described a tamper-resistant system [18] for registering time stamped documents in 1991. The structure they proposed bears a striking resemblance to what people now call blockchain, a cryptographically secure chain of blocks. Their idea to chain together hash values of actual documents was further expanded in a 1993 paper [6] where Merkle Trees were introduced to reduce the verification cost from N to logN and to allow several documents to be stored into one block.

Others who helped paved the way to Bitcoin were Wei Dai with his B-money paper [11], intended to be an anonymous and distributed electronic cash system and Adam Back

[3] https://www.investopedia.com/ask/answers/what-do-t1-t2-and-t3-mean/

[4] https://fexco.com/fexco/news/how-long-international-bank-transfers-take/, 14 July 2021

[5] https://www.chaum.com/ecash/

with Hashcash [5], which introduced a proof-of-work algorithm, later being also used in Bitcoin.

In 2009 the foundation for peer-to-peer financial services was laid when Bitcoin was released into the world by the enigmatic, and still unknown, Satoshi Nakamoto. Bitcoin was the first blockchain application, launched after the 2008 financial crisis, it aimed to make centralized financial institutions obsolete. Solving the double-spending problem using a peer-to-peer network and the proof-of-work algorithm was what put Bitcoin ahead of other past initiatives. Profiting off the first-mover advantage, the project gained rapid traction and sparked people's interest in the crypto world.

2.4 Blockchain features

As Alex and Don Tapscott explain in their "Blockchain Revolution" book [41], the blockchain can be characterised as the Internet of Value. This is in contrast with the traditional Internet used in the last decades where we exchanged information. Unfortunately, the old model could not provide enough guarantees for peer-to-peer transfers of value, which is why there was a need for intermediaries, such as banks, to ensure the integrity of our payments.

To be able to discard intermediaries, trust has to be ensured by alternative means and in blockchain this is achieved by combining some key characteristics [47]:

- **ledger** - an append only data structure where data can be stored without the risk of being modified or deleted

- **secure** - the strong cryptographic basis upon which blockchain is built - hash functions, digital signatures, etc, - ensures a high level of security for the contained information as well as for its integrity

- **shared** - in theory everyone can choose to join the network as a peer node and download the entire blockchain history and check its validity, in practice though, as time goes by and more and more blocks are appended, the required storage capacity to run a blockchain node might deter regular users: 250 Gb for Ethereum [6] and 343 Gb for Bitcoin[7] as of June 2021. Still, this feature ensures transparency between participants

- **distributed** - a distributed topology resembles a fully connected graph where each node is connected to all the other nodes. There is no difference between nodes in terms of authority, even though their individual computing power can differ drastically. The more distributed a network is, the more resilient to attacks it becomes. Also, availability of the network increases since no one can take down or destroy all nodes at once. As long as at least one fully synchronized node remains available, all the others can reconstruct their local blockchain history again

[6] https://blockchair.com/ethereum/charts/blockchain-size

[7] https://blockchair.com/bitcoin/charts/blockchain-size

2.5 Permissionless vs Permissioned

As this paper explores the Ethereum DeFi ecosystem, we are concerned only with permissionless blockchain networks. Nevertheless, understanding the key differences between permissionless and permissioned gives us a deeper understanding of the opportunities, as well as the limitations, of each type.

Permissionless blockchain networks are open to anyone since they are released as open source software. There is no high authority to manage the individuals' right to write to the blockchain, meaning to publish blocks or to read blockchain data. Of course, by being open to anyone, permissionless networks also attract malicious users trying to craft transactions that will bring them financial gains. The blockchain networks employ a protection mechanism which assumes the majority of nodes are not-malicious and that they hold more than half of the computing power of the network. This mechanism is called consensus protocol and will be touched upon shortly. Examples of permissionless blockchain networks are Ethereum, Bitcoin, Litecoin, etc.

Permissioned networks, as the name suggests, rely on some form of authority - centralized or decentralized - to determine who has the right to append new blocks. This approach has use cases in fields like banking, supply chain, etc, where the identity of the participants has to be established beforehand. Write access is usually restricted to allowed parties, but read access could be open to everybody or restricted to certain parties. Since all participants are known, the consensus mechanism for permissioned blockchains is usually faster than for permissionless networks since any misbehaving node can simply be excluded from the network. Some prominent permissioned networks are Hyperledger, Corda, Quorum, etc.

2.6 Consensus mechanism

For public and decentralized blockchains such as Ethereum and Bitcoin, the participants have to agree upon who has the right to publish the next block. Since the nodes do not trust each other as they are only known by their public address, they need a set of rules to unequivocally choose the next node to publish a block. Participants are incentivized by financial gains to be the chosen one, as the winner has the right to collect the transaction fees and/or a block reward. The consensus mechanism is what makes this group of mutually distrusting parties work together.

Two problems have to be solved in blockchain applications: double spending and the Byzantine Generals Problem [30]. Unlike fiat currency where you cannot use the same $5 for two different transactions, digital currencies with no central authority suffer from this problem. To solve it, all transactions have to be validated by all nodes. Whenever users join the network, their individual local copies start from the same ground-truth: the genesis block[8]. All the subsequent blocks are added to the genesis block such that the whole history can be traced back to this initial state. The Byzantine Generals Problem is widely known in distributed systems [21]. It describes a situation where some of the nodes are malicious and send conflicting information to the other peers. Satoshi Nakamoto

[8] https://www.blockchain.com/btc/block/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

introduced his solution to this problem in the Bitcoin whitepaper, namely Proof-of-Work. Since then, other mechanisms have tried to replace PoW because of environmental concerns. According to the Cambridge Bitcoin Electricity Consumption Index, the Bitcoin network consumes more electricity annually than Austria[9].

2.6.1 Proof-of-Work (PoW)

The main idea behind PoW is to let nodes engage in a hashing power competition. Nodes perform huge amounts of computations trying to solve a hard cryptographic problem. Specifically, participants attempt to find a nonce (number used once) by looping through all the possible values (0 to 2^32). This nonce, when combined with the hash of the previous block and the transactions chosen by the miner for his candidate block, and fed to a hash function, must produce a hash starting with a predefined number of zeros. The leading number of zeros represents the difficulty level. Each additional zero increases the difficulty and thus, the number of computations required to find the solution. In Bitcoin, difficulty is adjusted approximately every 2016 blocks to maintain the average block time at 10 minutes. The same happens in Ethereum where the difficulty is a function of an average block finding time.

Once the nonce has been found, the block is broadcasted to the network. The other nodes determine whether the new block fits into their known block sequence. If the block is valid, it will become the latest block in the chain. Participants with a higher hash rate are more probable to be the ones publishing the next block. The formula describing the probability of a node to find the next block in a network with N participants [33] can be expressed as:

$$p_i = \frac{c_i}{\sum_{j=1}^{N} c_j} \tag{2.1}$$

c_i represents the hash rate of $node_i$. Individual nodes try to increase their hash rate to have more chances of winning the race and receive the rewards. This has led to more usage of electricity and growing concerns about the environmental impact of PoW. However, since block creation is so expensive, it acts as a deterrent for attackers since a successful attempt at rewriting the blockchain history requires the attacker to build the longest chain. This is not possible without controlling more than half of the network hash rate power.

2.6.2 Proof-of-Stake (PoS)

PoS is the most popular choice when coming to PoW replacements. In PoS, the concept of coin age was introduced. Coin age is calculated as the value multiplied by the time period since the coin creation [30]. The probability to be selected as leader and have the right to publish the next block increases with the amount of coin age the user controls. This stake-based approach completely eliminates the dependence on the nodes' computational power, thus solving the huge electricity consumption issue.

[9] https://cbeci.org/cbeci/comparisons

The Follow-the-Satoshi (FTS) algorithm is an example of a PoS algorithm where each token is indexed. Then a hash function being fed a seed will output a token index and the current owner of that token is elected as leader. In a similar manner to the PoW relation, the probability of a node being selected as leader can be expressed as:

$$p_i = \frac{s_i}{\sum_{j=1}^{N} s_j} \tag{2.2}$$

s_i is the stake of participant i. Since the election process is much faster, PoS has a smaller block time and, therefore, the transaction throughput (the number of transactions a network can process per second) is increased. The transaction throughput is related to the block time as in equation 2.3. Therefore, PoS makes blockchains faster.

$$T_x/s = \frac{Block_{size}}{T_{x_{size}} \times Block_{time}} \tag{2.3}$$

2.7 Ethereum Platform

The blockchain used by Bitcoin proved to be too restrictive for more general use cases. This led to the birth of Ethereum, the first and most prominent smart contract platform. Ethereum is different from Bitcoin as it allows users to program and create their own operation instead of offering just a predefined set of options. The core component of this architecture is the Ethereum Virtual Machine (EVM) which offers a sandbox environment where code of random complexity can be executed.

As explained in [4], "Ethereum is a deterministic but practically unbounded state machine, consisting of a globally accessible singleton state and a virtual machine that applies changes to that state". The code of smart contracts can encode business transactions which alter the blockchain state. Particularly important for this thesis is the ability to create complex financial instruments that deal with token transfers and much more. This 'power of the code' means we can deploy completely autonomous applications that will always act deterministically when triggered, according to the conditions defined in code.

Of course, the realm of applications enabled by smart contracts is not limited to the financial sector. For years there has been active research and development in many diverse areas such as supply chain, gaming, healthcare, etc.

2.7.1 Accounts

The Ethereum global state consists of many entities that are communicating through a message passing framework. These entities are called accounts. Each account is identified by a unique 20-byte address and holds an internal state. There are two types of accounts in Ethereum: externally owned accounts (EOAs) and code/contract accounts. On one hand, externally owned accounts only hold the owner's ether and are controlled by a private key. On the other hand, code accounts contain instructions that control the behaviour of the account [4].

The major difference between the two types of accounts is that only externally owned accounts can initiate transactions on their own. An EOA can either send messages to another

EOA or to a contract account by using its private key to generate and sign transactions. Received messages may fire a predefined sequence of steps in code of the smart contract and different actions may take place during the transaction execution (token transfers, internal state changes, computations, etc.,). However, if any of the required steps fails, all the changes which took place up to that point are reverted. Any funds, other than the gas used up to the stop point, are sent back to the originating address as though the transaction would not have happened. We call this concept atomicity. The feature is especially used in DeFi where many actions which try to take advantage of arbitrage opportunities are chained together and submitted to the blockchain as a single transaction.

2.7.2 Transactions

Transactions are the starting point of any interaction which alters the blockchain state. We mentioned earlier that Ethereum can be viewed as a global state machine, only using transactions can this state change. A transaction can be thought of as being a single instruction which is created, cryptographically signed, serialized and submitted to the blockchain by an EOA. There are two types of transactions: message calls and contract creations.

When looking closer to the components of a transaction, we see the following fields can be found in both types [46]:

- **nonce** - counter representing the number of transaction initiated by the sender, used for mitigating reply attacks
- **gasPrice** - the amount of Wei (1 Wei = 1^−18 ETH is the smallest Ethereum subsenomination) per gas unit the sender agrees to pay for the transaction execution
- **gasLimit** - the maximum amount of gas that can be used for executing the transaction
- **to** - the 20-byte address of the recipient. In case of contract creation, it is empty (zero)
- **value** - the number of Wei that will be deducted from the sender's balance and transferred to the recipient address. In case of contract creation, an initial balance for the new smart contract will be set
- **v,r,s** - these values correspond to the signature of the transaction and are used for identifying the sender

Two more components need to be mentioned, the first one, **init**, is exclusively associated with transactions resulting in contract creation while the other, **data**, may only exist within a message call.

- **init** - represents a byte array storing the code used to initialize the new contract account. This piece of code is run only once at contract creation and then is discarded. From its execution, another code fragment called body is returned. The body will be permanently linked to the contract account

- **data** - a byte array containing the parameters of the message call. For example, a function from a smart contract might expect as parameter an integer representing an id

It was mentioned that only EOAs can create a transaction, but this statement is true only from an outer perspective. A contract can send messages to other contracts that exist in the same scope. These internal transactions are triggered by a parent transaction sent from an EOA. This may create a chain of calls from contract to contract, but these calls are limited by the gasLimit field of the parent transaction because this field is not present in the messages between contract accounts. The present study also investigates the internal transaction generated by calls made to DeFi smart contracts from our watchlist.

When transactions are submitted to the blockchain, they first end up in what is called the memory pool (mempool) before miners assemble them into blocks. Mining nodes are quite in a privileged position since they actively listen for new transactions and can read them before. Since miners receive the transaction fee after executing the transactions, they will prefer the ones offering a competitive gas price. The transactions are sent in plain text to the mempool and this allows miners to parse them and extract information they can further use to frontrun the respective transaction or even execute other attacks (ex: sandwich attack) from which they can directly profit. Any realization of this scenario is called Miner Extractable Value (MEV).

2.7.3 Blocks

Relevant pieces of information form what we call a 'block' in the network. In Ethereum, a block contains a header, information about the transactions it includes and a set of other blocks' headers (these blocks, called ommers, have the same parent as the current block's parent's parent).

Considering that, in Ethereum, blocks are added to the network much faster (15 seconds) than other blockchains (Bitcoin 10 minutes), more competing blocks are mined. Because only one of them can be added, the other blocks remain "orphaned". A solution to also include these blocks in the main chain is for miners to add their header in their current block. In order to be valid, an ommer has to be at most six generations older than the current block [46].

Some of the more relevant components of a block's header are:

- **parentHash** - the hash of the parent block's header, this link makes the set of blocks a chain
- **ommerHash** - the hash of the list of ommers added to the block
- **beneficiary** - the account address of the block's miner
- **difficulty** - the difficulty level of the block
- **number** - a counter for all the previous blocks starting from the genesis block which has the number 0
- **gasLimit** - the current limit of gas per block

- **gasUsed** - the total gas used by the transactions included in the block
- **timestamp** - the Unix time at the block's creation
- **mixHash** - a 256-hash which, together with the nonce, shows that enough computation has been put into mining this block
- **nonce** - a 64-bit value which is combined with mixHash

2.7.4 Gas

Since transactions have to be run by all network nodes to be validated, a Turing-complete language can easily enable software bugs that could result in the transaction running indefinitely, commonly known as the halting problem. Whether accidentally or intentionally, an infinite loop in a smart contract would essentially result in a denial of service for the platform. To combat this issue, Ethereum introduced the concept of gas fees. Each instruction executed in the Ethereum Virtual Machine has an associated cost measured in gas units.

When a transaction is created, two of the fields which need to be set are the gasPrice and the gasLimit. While the transaction is executed, the gas units for all instructions are summed up and multiplied with the specified gasPrice, resulting in the total gas fee. The gasLimit represents the maximum amount of computational steps the transaction can go through before it runs out of gas and stops. This mechanism ensures no transaction will run indefinitely as it becomes prohibitively expensive. The gasPrice, given in Wei, represents the price a user is willing to pay per unit of gas. It has a major impact on how quickly the respective transaction will be included in a block since miners give priority to the transactions having the highest gasPrice. In addition to being a metering mechanism, gas fees are an incentive for the miners as well since they are the ones collecting the fees.

2.7.5 Smart Contracts

The term 'smart contract' was first defined by Nick Szabo in 1994 as "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."[10]

Smart contracts are collections of code and data (or methods and state) which are deployed on the blockchain using the 'contract creation' type of transaction. Since transactions have to be executed by each network node, all participants need to end up in the same state after the execution, meaning the smart contract code has to be deterministic. To achieve this, smart contracts can only work with the data given as input. Data from outside the blockchain can be fed by oracles which are discussed in a later section.

Before being deployed, smart contracts have to be compiled. From the compilation process, the most important artifacts are the bytecode and the interface. The smart contracts

[10] https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

are compiled from the high-level language used by developers to machine code so that they can be run by every node in the EVM. As bytecode is not human-readable, developers need something in-between to allow them to interact with the deployed smart contracts. This is achieved using the ABI (application binary interface) which defines a standard scheme (JSON format) for representing the smart contract code. Calls to deployed smart contracts are done using the ABI.

Once deployed on the blockchain, the smart contract code cannot be altered and remains there as long as the network exists. Only the bytecode is stored on the blockchain, not also the ABI.

2.7.6 Ethereum clients

In order for users to join the Ethereum network, they first need to download an Ethereum client[11] and "run" their own node. A client is a piece of open-source software which implements Ethereum's technical specifications. Clients have been written in many different programming languages: Go, Rust, C#, Java, etc. The goal is to have many diverse clients such that there is no dominant client to create a potential single point of failure.

The Ethereum yellow paper [46] details how the networks should function, but there is no standard blockchain implementation. The freedom the developers from the Ethereum community had in building the software in any language they wanted created diversity in client implementations. This fact is crucial since each client can have bugs and, because not all users run the same client, the issues can be contained quicker and not affect the whole network.

At their core, all clients provide essential services like joining the P2P Ethereum network, synchronizing a local copy of the blockchain history, sending out new transactions, and creating/managing accounts. A full list of the most used Ethereum clients can be checked in the Ethereum documentation.

2.7.7 Types of nodes

Clients can be tuned to consume blockchain data in a particular way. This can create different types of nodes. Depending on what users want to achieve, they can choose how to synchronize their node as well, examples are fast, full, warp, snap, etc.

Full nodes

Full nodes store the whole blockchain data and validate all blocks, participating in the mining process. In case of node failure, a full node is queried to reconstruct the blockchain state. Clients provide APIs so that applications can query full nodes for data. Depending on the synchronization method, the initial synchronization can last from hours to days.

[11] https://ethereum.org/en/developers/docs/nodes-and-clients/#clients

Light nodes

The Ethereum network is constantly expanding which consequently results in an overwhelming increase in the amount of data needed to be stored by a full node. Scalability represents a dire concern surrounding the Ethereum network. One immediate solution to this issue is running a light node instead of a full one. The same clients that offer full node participation also provide this light option of syncing as an alternative. These nodes cannot see the pending transactions so they cannot take part in the mining process.

Archive nodes

This type of node stores the same data as a full node, but also keeps all the historical state. The hardware necessary for running an archive node makes sense for businesses like wallet vendors or blockchain explorers. The nodes not being synced using the archive method, will contain pruned data. Still, full nodes can reconstruct historical states on demand.

Chapter 3

Decentralized Finance

3.1 Building blocks

3.1.1 Cryptocurrency

Bitcoin, a cryptocurrency, was the first blockchain application back in 2009. Cryptocurrencies rely on cryptographic primitives such as public-key cryptography to ensure users can only spend their own assets. Each account consists of two keys. On one hand, there is the public key from which the account address is derived, is publicly known and used to receive tokens. On the other hand, the private key has to be kept secret as knowledge of the private key is needed to spend the coins. The main purpose of a cryptocurrency is to mimic fiat money in the digital world: offer people a medium of exchange, unit of account and store of value, while overcoming the shortcomings of centrally issued currencies.

In addition, the blockchain layer on top of which cryptocurrencies work provides protection against the 'double-spend' problem. Digital assets are easy to copy in general, this feature hindered digital currencies implementations since there was no secure way of ensuring a user could not spend the same cryptocurrency more than once. The append-only character of blockchain, together with the transaction verification mechanism will invalidate double spending attempts.

3.1.2 Oracles

Blockchains platforms are closed ecosystems. A smart contract's possible knowledge domain is limited to the data already residing in the EVM state.This fact has two consequences, as mentioned in [4]: firstly, there is no reliable source of randomness inside the EVM and secondly, data from the outside world can only be sent to the blockchain as transaction input. Eliminating randomness sources is crucial to ensure the code execution remains deterministic and all nodes running a certain transaction would trigger the exact same state changes. The second issue greatly reduces the utility of any smart contract since many applications need some external data to base decisions on.

Oracles try to bridge the gap between the off-chain world and the smart contract platform by bringing extrinsic information (stock price, exchange rates, etc) to blockchain. For DeFi protocols, price oracles are an essential piece to their functionality. As an example,

lending protocols such as MakerDAO[1] need real-time price of assets to determine whether loans have become under-collateralized and need to be liquidated. MakerDAO uses an oracle module[2] consisting of whitelisted oracle addresses and an aggregator contract. The prices broadcasted by the individual oracles are fed to the aggregator which computes the median price. This process is repeated for each collateral type. Other DeFi protocols choose to hook into decentralized oracle networks such as Chainlink[3]. The service provides "interfaces to off-chain resources for both smart contracts and other systems".

B. Liu et al. argue in [26] that the mechanisms behind the oracles deployment, frequency of price updates, aggregation of values from different sources, etc., are ambiguous and not transparent. The paper investigates the oracles used by MakerDAO, Compound, AmpleForth and Synthetix and makes recommendations on oracle design best-practices. The area of oracles is still immature and introduces a high risk for all DeFi protocols.

3.1.3 Stablecoins

The extreme volatility of cryptocurrencies may deter risk-averse users from engaging with DeFi applications. The historical price data of ETH[4] clearly displays this high instability. Therefore, a new class of cryptocurrencies was introduced to address the price volatility issue. They are called stablecoins and are designed to maintain a relatively stable value by being pegged to an underlying asset or currency. The Global Stablecoin Initiatives report[5] published by the International Organization of Securities Commissions mentions stablecoins can differ by the type of asset they are pegged to:

- fiat-backed - this was the first type of stablecoins and the most popular. They could be collateralized by one or even more fiat currencies. The off-chain reserve of fiat is usually kept by a regulated entity and in this case we have centralized custodial stablecoins. The circulating supply of the stablecoin has to be reflected by the related fiat currencies found in custody. The largest fiat-collateralized stablecoin by market capitalization is Tether[6] (USDT) with a value of $62B[7] (June 2021), being the third largest cryptocurrency behind Bitcoin and Ethereum. Another example is USDC[8] with a market cap of $21B. Both USDT and USDC are pegged 1:1 to the US dollar. They are especially used in DeFi protocols to generate yield while avoiding the adverse effects of market volatility. However, when using these stablecoins we should not forget the risks of centralization.

[1] https://makerdao.com/en/

[2] https://docs.makerdao.com/smart-contract-modules/oracle-module

[3] https://research.chain.link/whitepaper-v2.pdf

[4] https://coinmarketcap.com/currencies/ethereum/historical-data/

[5] https://www.iosco.org/library/pubdocs/pdf/IOSCOPD650.pdf

[6] https://tether.to/

[7] https://coincodex.com/cryptocurrencies/sector/stablecoins/

[8] https://www.circle.com/en/usdc

- backed by other real-world assets such as commodities, financial instruments, etc. - ensuring a stable price can also be done by using commodities such as gold or silver as collateral for stablecoins. Tether Gold[9] (XAUT) is an example where 1 XAUT = 1 troy fine ounce of physical gold. The market cap of XAUT is $158.76M at the time of writing.

- crypto-collateralized - these stablecoins are backed by an overcollateralized[10] amount of another cryptocurrency. These are decentralized non-custodial stablecoins, where the reserves are stored in smart contracts. The most common example of a crypto-collateralized stablecoin is DAI[11], created by MakerDAO. It is designed to maintain 1:1 parity with the US dollar while its value is backed mostly by Ethereum (ETH) locked up in the Maker collateral vault contract. The market cap of DAI is $5.26B and there are mechanisms in place to keep the price close to $1 USD. Another interesting crypto-collateralized stablecoins is sUSD[12], introduced by Synthetix, whose value also tracks the US dollar. To mint sUSD users need to stake Synthetix network tokens (SNX). Unlike DAI whose price is soft-pegged to USD, sUSD is hard-pegged through the exchange functionality of Synthetix.

- algorithmically controlled - these stablecoins are special in that they are uncollateralized. Their price is regulated using algorithmic expansion and contraction of supply. Ampleforth[13] (AMPL) is an example of such a stablecoin. The price-volatility is translated into supply volatility in Ampleforth. When the price goes above $1, the users' wallet balances increase proportionally and when the price goes below $1, the balances decrease accordingly. These adjustments are done daily. As stated on the website: "AMPL is an independent financial primitive that does not rely on centralized collateral or lenders of last resort. It's like Bitcoin, except it can be used in contracts".

3.1.4 Decentralized Applications and DAOs

Decentralized applications (dApps)[14] are a crucial element of the DeFi space. Unlike regular applications, dApps live on a smart contract platform like Ethereum. The main advantages of dApps over traditional software applications stem from the underlying blockchain infrastructure: permissionless nature and censorship-resistance. Anyone having an Ethereum wallet can interact with a dApp, as long as the smart contract conditions are met, once it has been deployed.

---

[9] https://gold.tether.to/

[10] "Over-collateralization (OC) is the provision of collateral that is worth more than enough to cover potential losses in cases of default." - https://www.investopedia.com/terms/o/overcollateralization.asp

[11] https://developer.makerdao.com/dai/1/

[12] https://research.binance.com/en/projects/susd

[13] https://www.ampleforth.org/

[14] https://ethereum.org/en#what-are-dapps

Decentralized autonomous organizations (DAOs)[15] are managed by a group of people where every decision has to be approved by a majority. There is no single owner, nor CEO since the ownership is shared among its members. All the logic behind changes and upgrades are encoded in smart contracts. If there is a proposal to spend an amount of money from the smart contract custody for investing in a DeFi protocol, the members have to decide together if they will do it because the smart contract will not allow individual users to withdraw funds. The great transparency and openness make DAOs a suited choice for trustless cooperation.

3.1.5 Initial Coin Offerings

Initial Coin Offerings or ICOs have become widely known as an user-friendly financing mechanism. Traditionally, when a company needed funding, it could have gone to either debt or equity markets. ICOs opened a new avenue where projects being in their early stages can organize crowdsales in return for utility tokens. A nice team and a polished whitepaper used to be enough to raise millions of dollars during the ICO boom between 2017-2018. As many of the advertised projects proved to actually be scams, people are now more cautious. As of July 2021, the list of upcoming DeFi ICOs on Ethereum had 31 entries[16] on icomarks.com.

Year	Number of ICOs	Total funds raised
2014	2	$16,032,802
2015	3	$6,084,000
2016	29	$90,250,273
2017	875	$6,226,689,449
2018	1253	$7,812,150,041
2019	109	$371,209,025

Table 3.1. ICO data taken from https://www.icodata.io/

3.1.6 Tokens

The Token Taxonomy Framework (TTF) was launched in 2019 by the Interwork Alliance[17]. The main purpose of the TTF is to establish a knowledge base for the token economy and, therefore, is an import step towards the Alliance's mission to "empower organizations to adopt and use token-powered distributed services in their day-to-day business operations"[18]. The TTF is platform-agnostic and it does not take any stand regarding the tokens

[15] https://ethereum.org/en/dao/

[16] https://icomarks.com/icos/defi?platform=ethereum&status=upcoming&whitelist=&kyc=&bounty=&mvp=&email_confirmed=

[17] https://interwork.org/

[18] https://interwork.org/about-us/

implementation, as only the specification is considered [40].

Token Features

Even though tokens can be created for different applications and purposes, there is a set of common features that all tokens share: valuable, representative, digital, discrete, and authentic. We say tokens are valuable because usually they can be evaluated in terms of a widely accepted standard, mainly the US dollar. By representative we refer to how tokens show the ownership or claim of someone to an asset, be it digital or physical. Because tokens live in the digital realm, usually recorded on blockchain, we say they are digital in nature. The discrete property should not be confused with fungibility. Here, by discrete we mean that each unit of a token exists independently of any other unit. Also, anyone should have the same view of a given token. The authenticity of tokens stems from the blockchain layer, being both public and permissionless, together with the consensus protocol, it enables us to verify the authenticity of each token in the same way we do for paper money.

Fungibility

Broadly speaking, there are two types of tokens: fungible and non-fungible. Fungible tokens are modelled after the fiat currencies we use in our everyday life. They can be divided depending on the declared number of decimals, with individual units being interchangeable and identical to each other, exactly like two newly minted bills of $1. Meanwhile, non-fungible tokens were introduced to represent the ownership over a unique asset. One could draw a comparison to a piece of artwork, even if created by the same artist, no two pictures can be totally identical.

Standards

As applications started to be built on Ethereum, the need for interoperability grew. The Ethereum community introduced Ethereum Improvement Proposals (EIPs)[19] which are design documents describing standards for the platform. There are different types of EIPs, the most relevant type for this thesis is ERC (Ethereum Request for Comments) which defines "application-level standards and conventions, including contract standards[..]"[20]. These are interfaces providing a core set of functionalities which should be implemented by every token smart contract. The first such interface was ERC-20[21] for fungible tokens. It was followed by the ERC-721[22] standard for non-fungible tokens and continued with ERC-1155[23] to add support for multi-token contracts.

Because from the aforementioned interfaces the most common one in the DeFi space is the ERC-20, we will explore its core functionalities in more detail. The methods which all

[19] https://eips.ethereum.org/

[20] https://eips.ethereum.org/erc

[21] https://eips.ethereum.org/EIPS/eip-20

[22] https://eips.ethereum.org/EIPS/eip-721

[23] https://eips.ethereum.org/EIPS/eip-1155

ERC-20 compliant token contract has to implement are the following:

- **totalSupply()** - returns the total supply of the token
- **balanceOf(owner)** - returns the balance of the owner
- **transfer(to, value)** - transfers the value amount of tokens from the caller's balance to the to address
- **transferFrom(from, to, value)** - allows contracts to transfer tokens on a user's behalf since the amount is not deducted from the caller's account balance, but from the from address. To not trigger an error, the user with the from address must have previously approved the caller to transfer a certain amount of tokens on his behalf
- **approve(spender, value)** - allows the spender to transfer from the caller's balance up to the value amount
- **allowance(owner, spender)** - returns the amount the spender can withdraw from the owner's account

3.1.7 Types of Fungible Tokens

Depending on their purpose, fungible tokens can be further broken down into categories such as utility tokens, security tokens, and governance tokens.

Utility tokens

This type of token is usually created for a specific purpose within a decentralized application or platform. In order to use some functionality of a system, users need to acquire the system token first. Even ETH can be considered as a utility token for the Ethereum ecosystem, especially in the beginning where dApps were mainly offering services in exchange for ETH. Now, the trend is for each new application to come with its own token to decouple itself from ETH economics, creating scarcity/demand on its own. For example, BAT[24] is a utility token for the digital advertising industry where publishers are rewarded for content creation and users for their attention. Other DeFi examples are SNX[25], used as collateral in Synthetix and the cTokens[26] created by Compound to represent the users' stake in a certain asset pool.

Security (Equity) tokens

After topping more than $2 trillion in April 2021[27], the cryptocurrency market is still way behind the global equity market which sits at over $100 trillion[28]. However, as Diloitte

[24] https://basicattentiontoken.org/#bat-utility

[25] https://docs.synthetix.io/tokens/

[26] https://compound.finance/docs/ctokens

[27] [https://coinmarketcap.com/charts/

[28] https://www.statista.com/statistics/376681/global-equity-market-capitalization-by-region/

identified back in 2018 [22], any real tradable asset can be represented on the blockchain through tokenization. These tokens are usually issued during events known as security token offerings or STOs to distinguish them from ICOs where all kinds of tokens (utility, payment, etc) can be issued. Holding a security token proves the owner's share in an entity which can be anything from companies to artworks. The P2P network allows participants to easily trade security tokens on secondary markets and disregard the intermediaries and 3rd party costs from traditional stock trading. Polymath[29] and Securitize[30] are just two examples of start-ups helping private companies raise money by issuing digital security tokens.

Governance tokens

Governance tokens give their owners voting power within a protocol. This type of token became particularly significant for decentralized financial applications, as the protocol creators aim to distribute the voting right among the participants in the true spirit of decentralization. As mentioned earlier, smart contracts are immutable and changes are widely scrutinised before being adopted. Usually, protocol developers create a governance module which dictates how changes can be applied through majority voting schemes. Depending on the adopted strategy and the planned use of the governance token, they can be created with either static, inflationary or even deflationary supply [19].

As explored by Jensen et al. in [20], there are also multiple methods of distributing governance tokens, trade-offs between decentralization, incentivized participation and secure venture capital are considered. The 3.2 table shows how Balancer, Compound, Uniswap and Yearn Finance have distributed their governance tokens. Initially, the founders like to retain a portion of the voting rights and distribute them to their internal team or, even keep them for future development in a treasury smart contract. Rewarding investors with governance tokens is also common, while incentivizing participation seems to be the area covered by most governance tokens.

Protocol	Initial Allocation				
	Retrospective users	Participation incentives	Founders & Team	Investors & Advisors	Ecosystem Treasury
Balancer	0%	65%	5%	25%	5%
Compound	0%	42.37%	26.05%	23.76%	7.82%
Uniswap	15%	45%	21.82%	18.18%	0%
Yearn Finance	0%	100%	0%	0%	0%

Table 3.2. Different allocation strategies for governance tokens

[29] https://polymath.network/

[30] https://www.securitize.io/

3.1.8 Wallets

Initially, blockchain wallets were only used to manage cryptocurrency balances and exchange funds easily. As the technology advanced, wallets have evolved as well to allow easy integration with different kind of tokens and decentralized applications. At a basic level, a cryptocurrency wallet is like a keychain. It does not store your assets itself, but it holds pairs of private and public keys. Externally owned accounts were previously discusses and it was mentioned that each EOA has both a private and a public key. Wallets are used to generate this key pair. The wallet uses the public key to derive the Ethereum address of the account and, by querying the blockchain, it displays the user's balance. Also, the wallet application is used to sign blockchain transactions with the private key.

Depending on whether the wallet is connected to the Internet, there are hot wallets and cold wallets [25]. Also, if only the user holds the private key we say the wallet is non-custodial. Cold wallets are all non-custodial wallets while hot ones can be either custodial or non-custodial.

- **hot wallets** - these are online wallets, simple and easy to set up, represent the most popular choice among users. Whenever someone creates an account on an exchange, downloads a desktop wallet, or installs a mobile wallet on a smartphone, a hot wallet is created. People engaging in daily activities involving cryptocurrencies, need the speed of a hot wallet, as transaction signing is seemlessly done with a few clicks.
- **cold wallets** - these wallets are not connected to the Internet, providing an increased level of security for the private key. Hardware and papers wallets are both example of cold wallets. Hardware wallets come as small devices, similar to USB sticks, where private keys are stored. Tokens are sent to a hardware wallet directly from a hot wallet. However, when tokens from the hardware wallet need to be spent, the device is connected to the internet via its dedicated software and the transaction is signed. The paper wallets are not as common anymore. They consist in a printed version of the private key together with the wallet address. Sending cryptocurrency to a paper wallet is the same as for the hardware wallet, while transferring tokens out implies importing the paper wallet in a hot wallet.

DeFi Wallets

With the emergence of DeFi applications, the need for user-friendly gateways grew.Even though the entry point into the crypto world mostly happens through a centralized exchange like Binance or Coinbase, further interactions with DeFi protocols require a non-custodial wallet as the user needs to sign transactions and this can only be done by having access to the private key. Exchange wallets are hot wallets most of the times, however, reputable exchanges will store the majority of their users' funds in cold hardware wallets to avoid security incidents. This feature is not usually available in other types of hot wallets like mobile-based or web-based.

We can thus infer the main characteristics of the DeFi wallets [7]:

- **non-custodial** - the user holds the private key and is in total control of the funds
- **key-based** - the wallet consists of a unique keypair, as opposed to centralized wallets where the user only has a unique address (public key) attached to his account. The

private key is usually represented as a 12-word seed phrase and users are responsible for backing it up

- **accessible** - DeFi wallets can handle multiple types of assets found in DeFi applications: ERC-20 tokens, non-fungible tokens, stablecoins, and ETH of course
- **compatible** - many wallets, especially the mobile ones, now provide custom integrations for DeFi protocols such that users can do all transactions seemlessly

3.2 Financial Primitives

3.2.1 Custody

A common topic among many DeFi protocols is the ability to pool users' funds together. In general, this would require a certain level of trust in the DeFi app and the team behind it. However, because the funds are escrowed in smart contracts whose code is made available to the public, the process is streamlined. Other than developers looking for bugs in open-source code, many DeFi projects choose to undergo code audits by established third party companies. This increases the trust people have in a particular protocol and their willingness to allow smart contracts to control some of their tokens for certain financial operations. In order to manage different tokens, a smart contracts has to be programmed to handle the interface of the supported token type. As discussed previously, the ERC-20 standard for fungible tokens and ERC-721 for non-fungible tokens.

The custody of a smart contract over users' funds opens up new possibilities for other financial instruments such as collateralized loans, insurance funds, token swaps, yield farming, etc [19].

3.2.2 Supply adjustments

Specifically applicable to fungible tokens, supply adjustment mechanism can be embedded in the smart contract implementing the ERC-20 interface. Tokens can be programmed to have an adjustable supply instead of a fixed one. Increasing the supply is done through minting new tokens and decreasing it happens through burning some of the existing tokens.

- **mint** - the number of tokens in circulation is increased through minting. Tokens are created with a clear purpose, programmed into the smart contract design. Usually, minting new tokens acts as incentive for certain user actions such as supplying assets to a liquidity pool or as a reward mechanism. Stablecoins increase the supply to move the price downward (decrease scarcity)
- **burn** - reduction of a token's supply translated into decreasing the number of token in circulation. Technically, rendering a token unusable by sending it to to an address or smart contract incapable of spending it. Sometimes this happens accidentally and many funds are lost in this way. Intentionally burning tokens as part of financial strategies is not uncommon in DeFi protocols. As an example, the Compound protocol uses cTokens to represent the stake of a particular user in an asset pool. When the user want to withdraw his funds, his cTokens are burned. Stablecoins can be burned to drive the price up (increase scarcity).

When the supply of a token is updated regularly depending on the token's price, we say we deal with an elastic token and the mechanism for adjusting the supply is referred to as a rebase. To peg a token to a specific price, rebases take place whenever the price slips away from the target price. Elastic tokens seems to be simillar to stablecoins, however, there are some differences. Stablecoins are not created for profit, their only purpose is to eliminate price volatility. Meanwhile, elastic tokens follow profit as their market capitalization grows and new tokens are minted. Another difference is that the supply of elastic tokens is dynamically shifting to target an un-collateralized peg price, while stablecoins have a semi-fixed supply [44].

3.2.3 Incentives

Stimulating the right user behaviour is a widely deployed strategy in the DeFi ecosystem. Encouraging a user to take certain actions deemed useful is possible with what we call positive incentives. Similarly, negative incentives are meant to discourage users from engaging in a detrimental behaviour. Incentives are payments or fees, usually denominated in the platform token or another utility token specifically created for this purpose. We can distinguish between staking rewards and direct payments [19].

Staking rewards

These positive incentives act as rewards for users who have a stake in the protocol. The rewards usually depend on the stake amount and on duration. For example, Yearn Finance has an Earn[31] service which aims to achieve the highest yield for the supported tokens. Users can deposit DAI, USDC, USDT, TUSD, sUSD, or wBTC and the protocols will programmatically monitor exchanges and constantly shift the tokens between them to earn the most interest. The Compound protocols also rewards any interaction such as borrowing, withdrawing or repaying a loan with its governance token, COMP[32].

Staking penalties

Staking penalties can be seen as negative incentives. By slashing - removing a portions of a user's stake, some unacceptable behaviour is punished. The removed amount can represent either the entirety of users' balance or just a part. Also, the conditions under which slashing is triggered are well known, defined in the protocols documentation.

Undercollateralized loans are a major trigger for liquidation, a process of selling/auctioning the collateral in an attempt to recover the debt value. A loan becomes undercollateralized when the collateral value becomes less than a predefined collateral-to-borrow ratio known as liquidation threshold. Network participants are incentivized to monitor loans and trigger liquidations if conditions are met. In Compound, theoretically, anyone can be a liquidator and repay the debt to receive the collateral at a discount. However, in practice, there are specifically designed bots which quickly grab the most profitable liquidations and only leave unprofitable ones for regular users.

[31] https://v1.yearn.finance/earn

[32] https://compound.finance/docs/comptroller#refresh-comp-speeds

Perez et al. [35] provide an interesting perspective into liquidation efficiency on Compound and, also, the change in user behaviour brought about by the introduction of COMP, the governance token. According to their analysis, users started to take higher risks and borrow more assets as they were receiving COMP in return. The COMP token was creating a positive cash flow for borrowers as well when its value was greater than the interest the borrowers had to pay. With more loans, the potential for liquidation opportunities surged. To be able to capture as much of the collateral as possible, bots had to become highly specialized at detecting unhealthy loans. In 2019, only 26% of the positions which became prone to liquidation were indeed closed during the same Ethereum block. The figure rose to 70% in 2020, showing the huge leap in efficiency.

Market contractions can also trigger a slashing event. This mostly happens with algorithmic stablecoins. To keep the price close to $1, users' balances can be decreased when the price falls behind. Not a penalty in the usual sense, it still represents a removal of funds, burning them in this case.

Direct rewards

It was mentioned in the previous section that anyone can liquidate an undercollateralized loan. A truly efficient system would have a fully autonomous process in place. However, as discussed in 2, the Ethereum platform is like a state machine where only externally owned accounts can trigger state changes. EOAs can be controlled by either real persons or bots. The EOAs providing maintenance services in the DeFi ecosystem are called keepers. Direct rewards are payments, positive incentives, received by keepers. Liquidating undercollateralized loans is only one example, many other maintenance tasks are required in DeFi protocols. Constantly feeding price updates to an oracle is another behaviour incentivized by direct rewards.

Fees

Fees can be seen as negative direct incentives as users usually pay fees when engaging with DeFi services. Fees are a funding mechanism, to keep the platforms running, but a percentage of them may be redistributed to some of the users, especially the liquidity providers, as rewards. These fees should not be confused with the gas fees paid for each Ethereum transaction. A concrete example is the 0.3% fee that Uniswap[33] charges for swapping tokens.

3.2.4 Swap

Swapping one cryptocurrency for another one might seem like a trivial thing but it was quite difficult to do until recently. Users would first need to exchange one token for fiat and then buy the second token to realize the swap. DeFi brought the need for new swapping mechanisms to allow traders to execute the exchange seamlessly . With their capability of embedding contractual rules, smart contract can help users swap their tokens. The contract can escrow the tokens and do the swap when all parties agree, however, the traders

[33] https://uniswap.org/docs/v2/advanced-topics/fees/

retain control over their assets until all conditions are met. Swapping on Ethereum takes place through decentralized exchanges (DEXes) which can either use the order matching approach for providing liquidity or, use the more recent mechanism of an Automated Market Maker (AMM).

Order Book Matching

An order book contains a list of bids and asks. A bid is an order to buy whereas an ask is an order to sell. The lowest price from the sell orders together with the highest bid form the top of the book and their difference is known as the spread. On one hand, traders can buy or sell on spot for the best available price through market orders. The DEX pairs the buyers and the sellers with the top of the book orders. On the other hand, limit orders allow users to specify the exact price for their order and wait until the DEX will find a match.

The order books can be stored both off-chain or on-chain. Off-chain means only the final trade takes place on-chain while the orders are stored on centralized servers called relayers. This is a drawback as third party servers can be censored. On-chain order book DEXes do not suffer from censorship, however, the cost for users is higher since they need to pay the gas fee for both placing and canceling orders. 0x[34] is an example of off-chain order book DEX, while fully on-chain swaps are offered by platforms such as Kyber[35].

This type of decentralized exchanges work well in liquid markets.A high trading volume will keep the spread low and execute large orders without much slippage [9]. We see this approach working especially well for established centralized exchanges like Binance[36] where trade volume is significant. Order book DEXes also have a couple of drawbacks which should not be overlooked.

Firstly, market manipulation techniques such as pump and dumb or wash trading are possible and there is no legal solution to punish such behaviour. Bad actors can post many buy orders in an attempt to influence the market sentiment and drive the price higher. An honest trader might think there is a huge demand for the respective asset and place a buy order himself, only to find out the malicious parties have canceled their buy orders, thus leaving the honest user deal with a high slippage.

Secondly, on-chain order book exchanges are susceptible to front-running. This attack happens when a miner monitoring the memory pool sees a transaction for an exchange and tries to profit from it by including his own transaction before it. If a trader submits a large order to buy a token, this would normally drive the price up and thus, the miner would make a profit if he would by the token before the observed buy order gets executed. Daian et al. [12] study how this information asymmetry affects decentralized exchanges. Their study focuses on arbitrage, a particular type of front-running. Arbitrage bots engage in priority gas auctions to make sure their transactions are executed at the right moment.

[34] https://0x.org/

[35] https://kyber.network/

[36] https://www.binance.com/en

Automated Market Making

An automated market maker is a algorithmic agent, commonly a smart contract, which provides liquidity in markets. It stores funds on both sides of a trading pair in liquidity pools and, depending on a deterministic algorithm, quotes swap prices to users. The ratio of the stored assets is a major component in determining the pricing function. The less amount available from a token, the more expansive it becomes. The most popular DEX using the AMM model is Uniswap[37]. To determine the trading price, Uniswap uses a simple formula called the constant product rule[38]: the product of the two token balances has to remain constant, therefore, whenever someone withdraws some amount of the first asset, some tokens of the second type have to be sold. Uniswap will be discussed in more detailed in a further section.

One big advantage of this kind of decentralzed exchanges is the fact they are always available and instant. As loong as the smart contract has some liquidity, users can buy token X without waiting for someone to sell it. Another plus for AMMs is enabling users to become liquidity providers and earn a share of trading fees.

One drawback of AMMs is the risk of high slippage for orders covering a high portion of the liquidity pool. Chen provides a great example [9] where an order representing half of the pool liquidity will double the price of the tokens. Maintaining the slippage under 1% would require each order to be 100x smaller than the liquidity pool. For liquidity providers, another present risk is that of impermanent loss. This can happen when the price shifts such that it would have been more profitable to just hold on to the tokens than giving it to the liquidity pool. Therefore, being a liquidity provider is only profitable when the earnings from the trading fees are higher than the impermanent loss.

3.2.5 Collateralized Loans

The first DeFi applications started out by offering the same services as traditional financial institutions, but doing it in a transparent, user-friendly and cost-effective manner. As such, debt and lending are common mechanisms across many DeFi protocols. Because Ethereum is open and users do not need to identify themselves, decentralized lending platforms allow participants to borrow assets and provide liquidity in permisionless and trustless manner [38].

Since identities are not established, there is an inherent counterparty risk for all DeFi loans. To prevent the borrowers from simply disappearing after obtaining the funds, loans which are not repaid during the same transaction have to be backed by an amount of collateral. The most common approach taken by lending platforms is to set a collateralization threshold higher than 100%, meaning the loans are overcollateralized. This threshold is especially increased for highly volatile assets used as collateral.Whenever the collateral losses value and the loan becomes undercollateralized, the position becomes candidate for liquidation.

There are different flavours of collateralized loans platforms [38]:

[37] https://uniswap.org/

[38] https://uniswap.org/docs/v2/core-concepts/swaps/

- **collateralized debt positions** - the collateral can be used to back the value of synthetic token. First, assets are locked in a smart contract and then the synthetic token is issued, funded by the debt position. Borrowers can thus generate and use liquidity in form of synthetic tokens while maintaining market exposure through the locked collateral.

- **collateralized debt markets** - in this case, no new token is created for the borrower, instead he receives the assets lent by someone else. Matching lenders and borrowers can be done in different ways: either peer-to-peer or pooled matching. On one hand, P2P collateralized debt markets directly match people willing to lend certain cryptoassets with the ones looking to borrow the same assets. The liquidity provider will start to earn interest only after a match is found. The interest rate is fixed and agreed upon by both parties. On the other hand, pooled matching protocols put all the funds from liquidity providers in a single smart contract called liquidity pool. Each token has its own liquidity pool. Variable interest rates can be programmed into the smart contract to take into account the supply and demand. Unlike the P2P matching, lenders start to earn interest immediately after depositing funds into the liquidity pool. However, only when the asset is scarce and in great demand, the loans become expensive and interest rates are high. Otherwise, if widely available, loans are cheap and lenders do not earn much from the interest.

3.2.6 Flash Loans

It was mentioned in the previous section that loans which are not repaid during the same transaction have to be backed by collateral. Obviously, this implies that borrowing funds, using them and repaying the loan, all in the same Ethereum transaction, can be done without any collateral. As seen previously, lending is risky. The fear of defaulting is countered by locking up some of the user's funds. However, when everything happens during the same transaction, the risk of defaulting is non-existent, so there is no need for collateral. This type of uncollateralized loan, typically known as flash loan, can be described using a simple pattern.

For taking out a flash loan, a smart contract is needed to encode all the operations to be done during this atomic transaction. First, a user borrows some funds. The assets are transferred to the smart contract. Once the assets are available, the pre-defined operations, interactions with other protocols where the borrowed tokens can be leveraged for different profit yielding financial strategies, are triggered. Upon successful completion of all operations, the funds are transferred back to the flash loan provider. Only if all funds, optionally together with some service fee, are returned to the provider, the transaction is said to be completed. However, if any step fails, all the previous operations which took place until that point are reverted to their initial state. Therefore, there is no counterparty risk in a flash loan, hence the non-collateral characteristic.

Wang et al. [43] identified some flash loan applications: arbitrage, wash trading, flash liquidity and collateral swap. In traditional settings some are illegal (wash trading) but, little to no regulations have been put in place for the DeFi ecosystem.

3.3 DeFi Key Moments

After its launch in 2015, Ethereum became the preferred smart contract platform, attracting more and more developers willing to build on top of it applications ranging from games (CryptoKitties) to financial ones (MakerDAO). Also introduced in 2015, the ERC20 token standard became the basis of every new token developed on Ethereum.

We will talk more about Maker in the next sections, but it is worth mentioning here as well. The project was initially formed in 2014, before the launch of Ethereum. The aim of the project was to create a permissionless credit system where users could take out loans against cryptocurrency collateral. The protocol was launched on Ethereum in 2017 and it became an inspiration for projects to come.

Another project which was an early pioneer, very popular in 2017, was EtherDelta. Among the first decentralised exchanges to appear on Ethereum, it allowed users to exchange ERC20 tokens between them, based on an order-book system. EtherDelta was especially used during the ICO boom in 2017, but, at the end of the year the exchange was hacked.

Initial Coin Offerings (ICOs) represented a new way of raising capital in exchange for tokens. Although there were many over appreciated projects, some of the ones launched back then are now successfully part of the DeFi space, among the prominent examples are Aave - a lending and borrowing protocol and Synthetix - a protocol which enables the creation of DeFi derivatives.

Around this time the idea of users interacting with a smart contract holding funds emerged, this new paradigm proved more efficient, especially for financial decentralised applications, as users would interact only with a smart contract and not directly with other users.

From 2018 Bitcoin entered into a bear market, with the price of all cryptocurrencies following the same trend. Many DeFi projects were actively being developed during this time, culminating with the launch of Compound and the first version of Uniswap on the Ethereum mainnet in November 2018. Uniswap was the first major decentralized exchange based on the concept of Automated Market Maker (AMM), as opposed to the traditional order matching approach.

Liquidity incentives programs started to appear as well, Synthetix launched its first such program in July 2019. These programs encouraged users to participate in certain protocols by providing liquidity and being rewarded with other tokens which represented their stake.

On March 11th, 2020 the World Health Organization declared COVID-19 a pandemic[39]. As fear and uncertainty amplified, the price of Ether dropped by more than 30% the next day. March 12th, 2020 became known as the Black Thursday in the crypto world. This sharp price decline meant that many loans taken against Ether were at risk of becoming under-collateralized. The gas price increased as many users were trying to supply more Ether to close their loans. The DeFi space quickly recovered and gained even more momentum. In May 2020 Compound started its liquidity mining initiative, rewarding the users

[39]Coronavirus confirmed as pandemic by World Health Organization, https://www.bbc.com/news/world-51839944

with the COMP token for lending and borrowing on the platform. This market making strategy brought more liquidity into DeFi, opening the way to what is now called the 'DeFi Summer'.The total value locked (USD) in DeFi grew from $1.072B on May 31st 2020 to an all time high of 86.191B[40] on May 12th, 2021.

[40] https://defipulse.com/

Chapter 4

Analysis

4.1 Methodology

The aim of our analysis is to explore the composability present in the DeFi ecosystem. As different types of protocols interact with each other, we want to observe whether clear patterns emerge. As a fist step, we relay on the data provided by DeFi pulse[1] and classify the protocols, according to the services they provide, into different sectors: lending, decentralized exchanges, derivatives and assets[2]. From each category we consider the first three protocols[3] in terms of Total Value Loked (TVL)[4] as of March 2021. The next step is to compile a list of deployed smart contract addresses per protocol. Having these Ethereum addresses, we can next query the blockchain and extract the transactions in which the concerned addresses are involved. Once we have the transactions, both internal and external, we try to leverage a graph database to build a network and further analyse it. This process is exemplified in figure 4.1.

The next sections will discuss the protocols which were chosen for this study, the data collection procedure, network construction and the actual analysis.

4.2 DeFi Protocols

This section aims to provide an overview of the main protocols included into our analysis. However, due to space limitations, some of them are only briefly discussed, the focus being on the ones that have had a significant impact on the ecosystem, being first-movers and bringing innovative products and services.

---

[1] https://defipulse.com/

[2] There is also a 'payments' category on DeFi Pulse, but we chose not to include it into our analysis since only a small fraction of applications are under this category, mostly layer 2 applications

[3] for DEXes there are 4 protocols because Curve FInance uses a different programming language for its smart contracts (Vyper), but it was later included as well since there was no observed difference due to this fact

[4] DeFi Pulse calculates this value by monitoring the balance of ETH and ERC-20 tokens from the smart contracts of each protocol. The balances are then multiplied by the assets price in USD

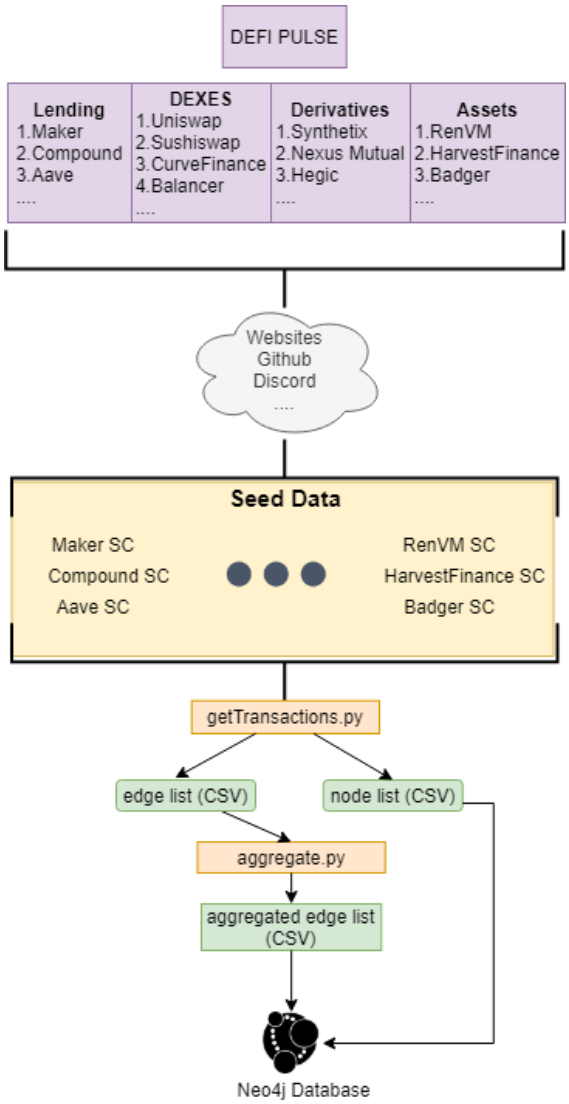


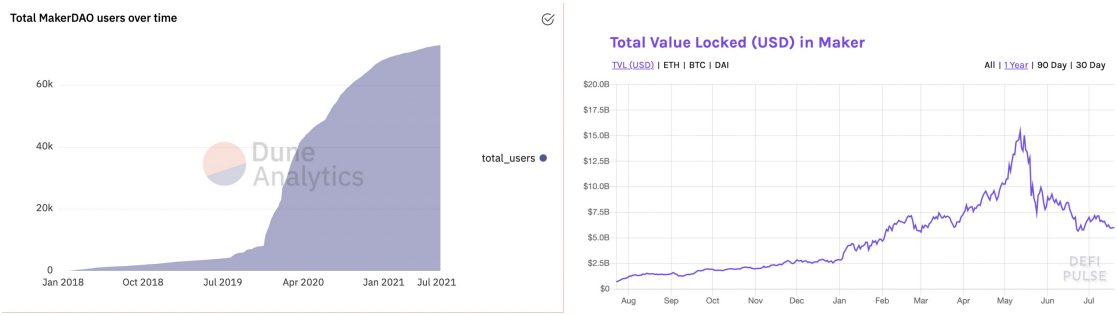
Figure 4.1. Methodology Flowchart

4.2.1 Lending

The explored lending protocols are Maker, Compound and Aave. Table 4.1 presents a summary of the main statistics about the three protocols. Aave became dominant in terms of TVL, while Maker retains the highest percentage of circulating ETH locked in its system. All protocols have a governance module where protocol token holders are given voting rights, all aiming towards a truly decentralized autonomous organization (DAO) to take all protocol-related decisions.

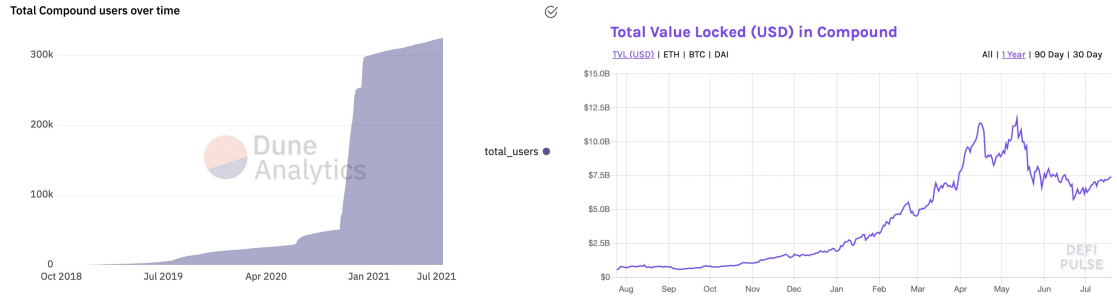
Protocol	TVL	ETH locked	% Supply locked	Protocol token
Maker	$6.01B	2.48M	2.12%	MKR
Compound	$7.37B	1.28M	1.10%	COMP
Aave	$10.81B	1.75M	1.50%	AAVE

Table 4.1. Key statistics for lending protocols. Source: Defi Pulse. Data as per 19^{th} July 2021. Retrieved from https://defipulse.com/



(a) (b)

Figure 4.2. (a) Total MakerDAO users over time. Source: Dune Analytics, Data from 19^{th} July 2021. Retrieved from https://duneanalytics.com/queries/2951/5696 (b) Maker - Total Value Locked (TVL) in USD. Source: Defi Pulse. Data as per 19^{th} July 2021. Retrieved from https://defipulse.com/maker/

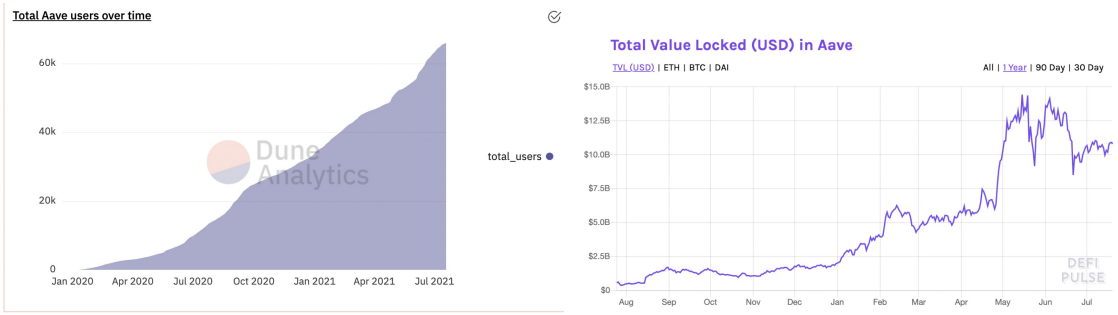


(a) (b)

Figure 4.3. (a) Total Compound users over time. Source: Dune Analytics, Data from 19^{th} July 2021. Retrieved from https://duneanalytics.com/queries/1010/5530 (b) Compound - Total Value Locked (TVL) in USD. Source: Defi Pulse. Data as per 19^{th} July 2021. Retrieved from https://defipulse.com/compound/

Maker

Maker was the first DeFi protocol to gain wide adoption. Maker is a two-token system. On one hand there is Dai, the flagship product, a collateral-backed stablecoin, soft-pegged



(a) (b)

Figure 4.4. (a) Total Aave users over time. Source: Dune Analytics, Data from 19^{th} July 2021. Retrieved from https://duneanalytics.com/queries/2994/5785 (b) Aave - Total Value Locked (TVL) in USD. Source: Defi Pulse. Data as per 19^{th} July 2021. Retrieved from https://defipulse.com/aave/

to the US dollar. On the other hand, there is the governance token, MKR, which allows holders to vote on protocol changes. Users can generate liquidity in the form of Dai after locking collateral into smart contracts called Maker Vaults [28]. Initially, only ETH could be provided as collateral, and the generated DAI was called Single-Collateral Dai or SAI. Since November 2019, Maker added support for multiple collateral types which have been previously approved by the MKR holders. To overcome price volatility of collateral, the collateralization ratio is well above 100%, according to https://daistats.com/ the ratio is 165.23% as of July 2021.

The generated Dai can be used as any other cryptocurrency, sold and bought on exchanges, or as a payment method for goods and services, however, the Maker protocol offers a savings feature for Dai holders called the Dai Savings Date (DSR). When users lock their Dai into the DSR contract, the amount starts to accrue interest in accordance with the DSR set by MKR holders. To retrieve the original collateral from the Maker Vaults, users have to pay back the generated Dai, plus the accrued interest called stability fee.

In case some Maker Vaults lose their value by depreciation of the locked collateral, they can be liquidated through auctions. Each vault has its own liquidation ratio which is the collateral-to-debt ratio at which the vault is at risk of liquidation, determined through Maker governance for each collateral type [28]. There are also a number of key actors to maintain the system. Keepers are entities, usually automated, that trigger liquidations and help with keeping the Dai price close to $1 by selling when the price is too high and buying when the price is too low. When auctions are held after a vault is liquidated, the keepers can buy the collateral at discounted prices. Other participants in the good running of the system are oracles, which ensure real-time price information for the collateral types is available so that liquidations are triggered at the right moment and the health of the system is maintained.

Compound

Compound is a decentralized protocol that enables liquidity providers to lend their assets and earn interest and borrowers to quickly obtain liquidity without negotiating terms like duration or maturity. Money markets are created for each Ethereum asset type. Compound pools together all the supplied assets of a certain type and, depending on the supply and demand for that asset, it algorithmically determines the interest rate. This aggregation provides substantially more liquidity than peer-to-peer lending platforms [24].

Whenever a user supplies an asset to a Compound pool, his share in the respective pool is tokenized and represented by an amount of newly minted cTokens. These are ERC-20 tokens that earn interest in time, the holder being able to redeem more of the underlying collateral in the future. There is an exchange rate between each type of asset and its corresponding cToken, for example, 1 USDC = 45.284196407894825 cUSDC[5] or 1 cUSDC = 0.022034101717091 USDC. When a market is launched, the cToken exchange rate for it starts at 0.02000, therefore we clearly see how one can earn interest in this case as cUSDC accumulates the interest paid by borrowers.

In order to borrow tokens from Compound, other types of assets have to first be supplied to be used as collateral. Each type of collateral has its own collateral factor, ranging from 0 to 1, representing what percentage of the underlying asset can be borrowed. These collateral factors can be changed by Compound Governance. Low collateral factors characterize small and illiquid assets, deemed as risky collateral. Meanwhile, liquid tokens with high market capitalizaiton make good collateral and have a high factor[24]. For example, if a user supplied 1000 Dai in the Dai pool, and considering the 75% collateral factor[6], it means the user can withdraw 750 Dai worth of other assets. Once the necessary collateral is supplied, the user has to enter the market for token he wants to borrow. This is done to notify the protocol that the supplied assets are to be used as collateral and, thus, cannot be withdrawn anymore until the debt is paid. The interest rate is variable, recomputed with every block.

Similar to the key actors discussed for MakerDAO, keepers and oracles are also present in Compound. Here, keepers receive a percentage of the collateral of the liquidated position.

Aave

Aave emerged under its current form from its predecessor, ETHLend, back in 2018. Aave has very similar features to the ones found in Compound, however, it added certain features that set it apart and helped the platform achieve market dominance.

There is a similar mechanism for lenders where they receive aTokens that represent the locked assets. Unlike Compound, Aave's aTokens are in a 1:1 ratio to the underlying asset and the earned interest is distributed to the lenders by increasing their aToken balance directly.

Another difference between the two protocols is the fact that Aave allows users to switch between variable and fixed interest rates. The number of markets is significantly higher for Aave, users having more options to choose from. At the time of writing, there were 26

[5] https://compound.finance/markets/USDC, accessed at 19^{th} July 2021

[6] https://compound.finance/markets/DAI

markets for Ethereum assets on Aave, while for Compound only 14 are listed. Also, Aave extended its offering by working with Polygon network[7] to create a layer 2 solution[8][2]. As such, Aave is now classified as a multi-chain system.

Probably the major contribution of Aave to the DeFi ecosystem is the introduction of flash loans in 2020. Refer to section 3.2.6 for a complete description of flash loans.

4.2.2 Decentralized exchanges

The chosen protocols from this category are Uniswap, CurveFinance, Balancer and Sushiswap. The focus is on Uniswap as it was the first DEX to earn a huge user base and it became an invaluable component in the DeFi ecosystem. The other protocols are discussed in comparison with Uniswap.

At the time of writing, all the mentioned DEX protocols have governance tokens. Uniswap, Balancer and CurveFinance use DAOs for their governance, while SushiSwap is yet to transition to a DAO.

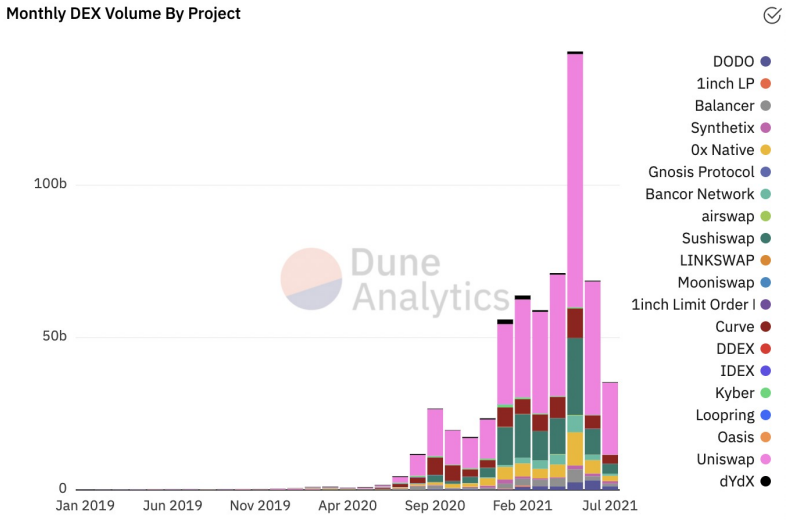


Figure 4.5. Monthly DEX Volume By Project. Source: Dune Analytics. Data from 20^{th} July 2021. Retrieved from https://duneanalytics.com/queries/1847/3261

Uniswap

The first version of Uniswap allowed users to swap ETH for other ERC-20 tokens by directly interacting with smart contracts called exchange contracts. These functioned as automated market maker between ETH and an ERC-20 token. A ETH-ERC20 exchange contract

[7] https://polygon.technology/

[8] In Ethereum a layer 2 application is built on top of the original blockchain infrastructure with the aim of scaling and increasing the transaction throughput

Protocol	TVL	Gov token	DAO
Uniswap	$4.72B	UNI	Yes
Curve Finance	$6.77B	CRV	Yes
SushiSwap	$2.35B	SUSHI	No
Balancer	$537.7M	BAL	Yes

Table 4.2. TVL and governance details for DEX protocols. TVL Source: Defi Pulse. Data as per 20^{th} July 2021. Retrieved from https://defipulse.com/

would contain liquidity on both sides. As Uniswap smart contracts are not upgradable, old versions will continue to exists in parallel with new versions as long as the Ethereum blockchain exists. The second version, Uniswap V2 [1], allows swaps between any two ERC-20 tokens.

Anyone can create a Uniswap pair if the desired contract does not already exist. Next, liquidity has to be provided to the smart contract pair. This happens when users send tokens of both types to the contract. Anyone who does this becomes a liquidity provider (LP) and receives an LP token for the respective pool. As Uniswap charges 0.3% for each swap and stores these fees into a reserve, LPs can burn their LP tokens to redeem their share of the total reserve.

Uniswap uses a constant product formula $x \times y = k$ where k is referred to as the invariant since it is not changed by trades. If a Uniswap pair consists of tokens A and B, to withdraw an amount of token A, an equivalent amount of token B has to be deposited to maintain the invariant.

The figure 4.6 shows how traders and liquidity providers interact with a Uniswap pair smart contract.

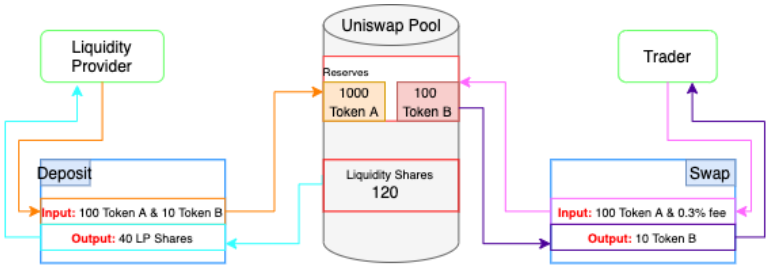


Figure 4.6. Example of traders and LPs interaction with Uniswap Pair SC

One of the most interesting and desirable features of Uniswap are the flash swaps[9]. Users can withdraw up to the full reserves of any ERC20 token, use it however they want, maybe to profit off some arbitrage opportunity, and return the amount at the end of the transaction. The returned amount can be either the withdrawn token type plus a small fee or the equivalent amount of the corresponding token pair.

[9] https://uniswap.org/docs/v2/core-concepts/flash-swaps/

In terms of governance, Uniswap uses a DAO where holders of the protocol token, UNI, can vote on proposals and take decisions in a decentralized manner. There is a minimum threshold of 10M UNI to be held by an address in order to be able to propose governance actions. For a brief period of type after launching the UNI token in September 2020, liquidity providers to four of the Uniswap pools could stake their LP tokens into an yield farming contract to earn even more rewards.

Curve Finance

The core concepts on Uniswap are also found in Curve Finance[10]. The Automated Market Maker of Curve is specialized on stablecoin trading. According to [15], the constant product rule employed by Uniswap does not perform well in case the price of the two tokens is correlated, or in the same range. Curve introduced a new automated liquidity provider for stablecoins, called StableSwap, which uses a different formula for the invariant to keep the slippage at minimum. Curve Finance has fees ranging from 0.04% to 0.4%, so swaps can be significantly cheaper than on Uniswap.

The Total Value Locked shown on DeFi Pulse for Curve Finance is higher than the one for Uniswap and this is due to the yield farming[11] present in Curve. Unutilized deposits from liquidity pools are locked into other DeFi protocols to generate additional revenue.

Similar to the other protocols, Curve has a DAO for goveernance, its token being CRV. Liquidity providers receive CRV and they can participate in voting and decision making, moreover, they can also stake their CRV to receive trading fees.

SushiSwap

SushiSwap[12] was created as a fork of Uniswap in August 2020. It differes from Uniswap in the way rewards are distributed. In Uniswap, the entire 0.3% fees for all swaps goes to liquidity providers, while in Sushiswap, only 0.25% is distributed to LPs, the rest of 0.05% is rewarded to holders of the SUSHI protocol token [10].

In addition, the yield farming for the SUSHI token is still going on SushiSwap, while Uniswap stopped its similar UNI program. When users stake their SUSHI tokens, they receive xSUSHI which is another token representing their share of the 0.05% pool coming from trading fees.

Balancer

Like Uniswap, Balancer was introduced as yet another AMM. However, Balancer sets itself apart by allowing users to create multi-token pools or even private pools. Multi-token pools can have different token ratios. For example, a pool with 3 tokens can have the ratios set to 25%, 50% and 25%. Such a pool will be rebalanced whenever a trade occurs, as people are naturally incentivized to profit from these arbitrage opportunities.

---

[10] https://curve.fi/

[11] Yield farming, also known as liquidity mining, refers to additional rewards earned by liquidity providers via staking

[12] https://sushi.com/

Weighted pools were the first type of pool introduced in Balalncer V1. They are a generalization of Uniswap's constant product AMM. The $x \times y = k$ becomes $\prod_t B_t^{W_t} = V$ [29] for a pool with t tokens, where B and W represent the balance and weight, respectively, for each token t and V is the invariant. For assets which need to trade at parity, such as stablecoins, Balancer uses the same approach as Curve Finance, namely the StableSwap invariant.

The launch of Balancer V2 in May 2021 brought about a new set of innovations. The architecture of the pools has been changed. There is now a single Vault smart contract holding all the assets and handling all token transfers, while individual Pool contracts are responsible only for the swap and liquidity provision/removal logic. This change aimed to reduce gas fees for multi-hop trades. In addition, it allows Balancer to offer flash loans.

A unique feature of Balancer, the private pools can be customized by their creators who can set the trading fee in the range from 0.0001% to 10%. This fee will be proportionally distributed to all LPs of the respective pool. Private pools are extremely useful for asset managers with large porfolios.

4.2.3 Derivatives

Derivatives are financial instruments that, as the name suggests, derive their value from the price of an underlying asset which can be anything from stocks or bonds to commodities or cryptocurrencies. The chosen protocols from this category are Synthetix, Nexus Mutual and Hegic, with the main focus being on the first one. Synthetix was the first major platform offering derivative products and it quickly became one of the dominant forces of the ecosystem.

Protocol	**TVL**	**Token**
Synthetix	$1.14B	SNX
Nexus Mutual	$292.5M	NXM
Hegic	$33M	HEGIC

Table 4.3. Total Value Locked (USD) and Protocol tokens in Derivatives. Source: Defi Pulse. Data as per 21st July 2021. Retrieved from https://defipulse.com/

Synthetix

First launched back in 2018 under the name Havven, the project was later rebranded as Synthetix, a decentralized platform for issuance of synthetic assets, also called Synths, on Ethereum. Synths are ERC-20 tokens that track other assets, even from different markets. The system allows users to benefit from the price movements of the underlying asset without actually owning it. In simple terms, off-chain assets can be replicated on-chain via Synths. Inter-blockchain liquidity can also be exploited by creating synthetic assets that track cryptocurrencies from other blockchains.

It is of critical importance to maintain the price of Synths on par with the assets they track. The Synths' price can sometimes fluctuate as they can be traded on secondary markets. There are three mechanisms, described in the Synthetix whitepaper [42], to maintain

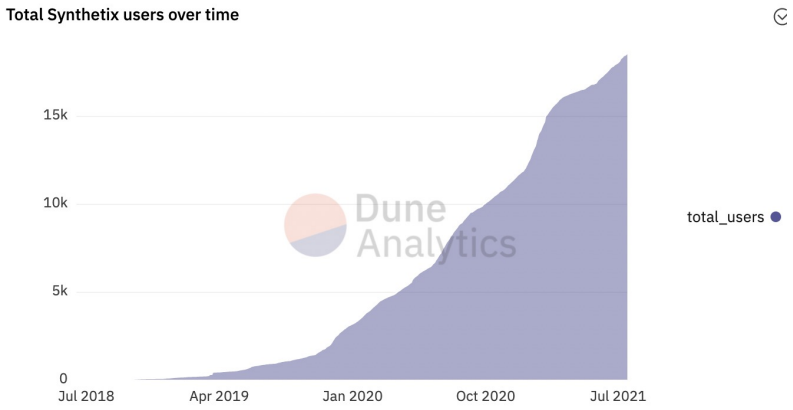
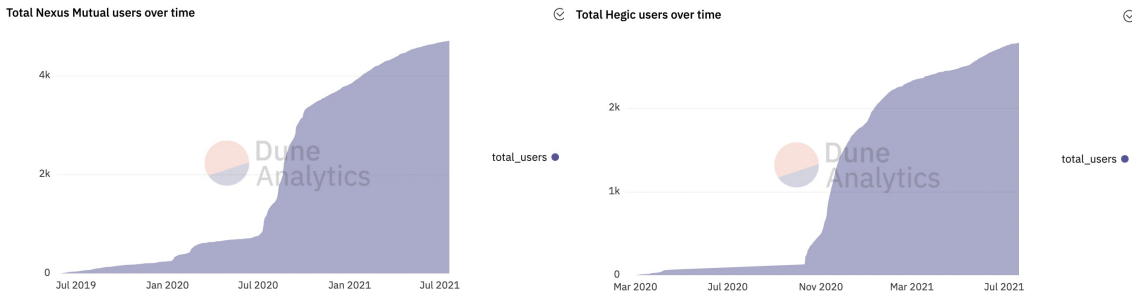


Figure 4.7. Total Synthetix users over time. Source: Dune Analytics, Data from 21^{st} July 2021. Retrieved from https://duneanalytics.com/queries/2961/5722



(a) (b)

Figure 4.8. (a) Total Nexus Mutual users over time. Source: Dune Analytics, Data from 21^{st} July 2021. Retrieved from https://duneanalytics.com/queries/2964/5729 (b) Total Hegic users over time. Source: Dune Analytics, Data from 21^{st} July 2021. Retrieved from https://duneanalytics.com/queries/13349/26740

the peg. One of them, which highlights the great interconnection between different DeFi protocols, is the sETH liquidity pool on Uniswap. Liquidity providers to the sETH/ETH pool on Uniswap also receive a portion of the SNX generated through the inflationary monetary policy on a weekly basis.

To mint Synths, users have to deposit an amount of Synthetix Network Token (SNX) as collateral. The collateralization ratio is currently at 750%, but it can be changed via governance. Until recently all Synths could be backed only by SNX, but there is a trial running for also accepting ETH as collateral, with a collateralization ratio of 150% [42].

There are incentives for SNX holders to stkae their tokens for minting Synths. Whenever someone exchanges one Synth for another one, a trading fee, usually 0.3% goes to a fee pool from which SNX stakers can claim their reward, according to their share, weekly.

The system of oracles is crucial for the good functioning of the system. Chanlink is

responsible for providing accurate off-chain information to Synthetix.

Nexus Mutual

Nexus Mutual[13] is a decentralized protocol, providing an alternative to traditional insurance for the DeFi ecosystem. The system functions like a mutual insurance company, a risk sharing pool, owned by its members. Any profits made by the pool are distributed among the members. Due to the more complicated legal framework around the insurance business, to become a member of Nexus Mutual, people have to pay a membership fee of 0.002 ETH and undergo a Know Your Customer (KYC) check. The cover products include yield token cover, protecting against de-pegging of yield-bearing token, protocol cover, mitigating the risk of a protocol hack, and custody cover for funds stored on centralized exchanged.

Nexus Mutual has its own token, NXM, which is a bit different from the other tokens discussed in this paper. It can only be bought by members on the Nexus Mutual platform and used there to buy insurance covers and also participate in the protocol governance. NXM holders can vote whether to cover certain smart contracts, a risk assessment process. In addition, NXM can only be transferred between protocol members. However, there is a wrapped version which can be traded outside the system.

Hegic

Hegic[14] was launched in 2020 as a permissionless and non-custodial alternative to options[15] contracts. Call options allow assets to be bought by the holder at a set price within a given period, while put options enable the holder to sell the underlying asset at a predefined price in a certain timeframe. To achieve this, Hegic offers hedge contracts which are on-chain contracts behaving like an option. Users can use hedge contracts to protect themselves from price downsides. For example, a Hegic user fearing of a sudden decline in ETH price can buy a put hedge contract which allows him to swap ETH for Dai at a set price, called strike, until the put contract expires [45].

Users can also provide stablecoins to the Hegic liquidity pool and receive writeASSET tokens to represent their share. LPs earn the premiums obtained from holders of hedge contracts. The HEGIC token is used for governance as well as for distribution of settlement fees.

4.2.4 Assets

The protocols from this category deal with inter-blockchain liquidity and automatic yield farming strategies. The concerned protocols for this paper are RenVM, Badger DAO and Harvest Finance.

---

[13] https://nexusmutual.io/

[14] https://www.hegic.co/

[15] Options are financial instruments that are derivatives based on the value of underlying securities such as stocks. An options contract offers the buyer the opportunity to buy or sell—depending on the type of contract they hold—the underlying asset. Source: https://www.investopedia.com/terms/o/option.asp

Protocol	TVL	Token
RenVM	$372.3M	REN
Badger DAO	$334.3M	BADGER
Harvest Finance	$310.2M	FARM

Table 4.4. Total Value Locked (USD) and Protocol tokens in Assets. Source: Defi Pulse. Data as per 22^{nd} July 2021. Retrieved from https://defipulse.com/

RenVM

RenVM[16] is a decentralized protocol for tokenizing cryptocurrencies which are not native to the Ethereum blockchain. ERC-20 tokens, are created to represent assets such as Bitcoin, Bitcoin Cash, Zcash, etc. Essentially, RenVM offers access to inter-blockchain liquidity.

It is important to understand the bigger picture of how RenVM fits within the DeFi ecosystem. Many DeFi projects can easily interact with each other since most of them are built on Ethereum. However, not all people wishing to use Ethereum dApps dispose of Ethereum-based liquidity, many having their money locked in other cryptocurrencies such as Bitcoin or Bitcoin Cash. Therefore, they first need to sell their assets and buy ETH or ERC-20 tokens. One other solution was for them to issue synthetic tokens, see 4.2.3. Another approach is tokenization, where users can lock up an asset and receive an ERC-20 token, a one-to-one representation of the locked assets [48].

There are other platforms offering wrapped versions of cryptocurrencies coming from other blockchains such as wrapped Bitcoin[17], with locked Bitcoin being kept in custody by a centralized party. RenVM, on the contrary, does not rely on any centralized custodian. The project employs a network of so-called Darknodes, which share their computing power and storage capacity in exchange for fees. To mitigate network attacks, an amount of 10000 REN (protocol token) has to be deposited before anyone is allowed to run a Darknode. Security and privacy are at the forefront of RenVM, as the project leverages zkSNARKs and secure multi party computation algorithms to execute programs in zero-knowledge [37].

Badger DAO

The goal of Badger DAO[18] is to speed up the use of bitcoin in DeFi. The platform functions like a shared space where developers are encouraged to collaborate and build products with the aim of using Bitcoin as collateral, not just on Ethereum, but other blockchains as well.

Anyone can propose ideas for new products and community members, holders of the governance token, vote and pick the brightest ideas.

Badger DAO's main product is Sett, a vault where users can deposit their tokenized bitcoins to be leveraged across other DeFi protocols for generating yield.

[16] https://renproject.io/

[17] https://wbtc.network/

[18] https://badger.finance/

Harvest Finance

Harvest Finance[19] is a platform where users can safely deposit their assets and let the system automatically find the best yield farming strategies among different DeFi protocols.

Listed on the Harvest Finance website are cryptocurrencies, tokens and token pairs. For each of them, the annual percentage yield is displayed. A user owning Dai, can deposit them in Harvest, get back fDai, which is the yield-bearing version of Dai, and start earning interest. In addition, as seen in other protocols, the fTokens can be also staked to earn the FARM protocol tokens.

4.3 Data Collection

4.3.1 Seed Data

Having chosen the DeFi protocols discussed in 4.2, the next step was to build a list of smart contract addresses for each of them. Considering there was no central repository we could check, the task was entirely done manually by looking into the protocols' documentation, websites, Github repositories, Discord channels, forums, etc. The goal was to have a CSV file for each protocol containing, at least, two columns: smart contract address and label. We called these addresses the seed data. Table 4.5 shows how many smart contracts were identified for each protocol and table A.1 displays the seed data sources.

Protocol	**Type**	**Nr. of SC**
Maker	Lending	130
Compound	Lending	52
Aave	Lending	91
Uniswap	DEX	259
Sushiswap	DEX	6
Curve Finance	DEX	133
Balancer	DEX	9
Synthetix	Derivatives	226
Nexus Mutual	Derivatives	21
Hegic	Derivatives	8
RenVM	Assets	21
Harvest Finance	Assets	37
Badger	Assets	59

Table 4.5. Number of smart contracts gathered per protocol

[19] https://harvest.finance/

4.3.2 Transactions

The next step was to gather the Ethereum transactions, internal and external, containing the said smart contract addresses as either source or destination. Initially, we relied on the free API provided by Etherscan[20] to query for the needed transactions. The limitation of using Etherscan is that for internal transactions, only the ones carrying value (method calls which also transfer ETH) are returned. This last point will prove to be a major obstacle in fully understanding the interactions between both smart contract between different protocols.

We used Python to automate this stage. Two API calls from Etherscan were used, one for external transactions and one for internal ones. From each returned transaction we kept only the relevant fields for our analysis. The script was parameterized so that, based on the current needs, the output could consist of an edge and node list (two CSV files) for either internal, external or both types of transactions. We use the concepts of edge and nodes from graph theory since, at the next stage, these lists are going to be loaded in a graph database so that more specific queries can be made.

Edge list

Here we can differentiate between internal, external and combined edge list, with the latter being just a concatenation of the first two. We chose to make this difference because we later tried to plot only internal calls between protocols or just the external ones.

The response from the Etherscan API for 'normal' (external) transactions is a JSON document with a status field being either 0 (error) or 1 (correct) and a result field consisting of the transactions list. Each transaction has the following information: blockNumber, timeStamp, hash, nonce, blockHash, transactionIndex, from, to, value, gas, gasPrice, isError, txreceipt_status, input, contractAddress, cumulativeGasUsed, gasUsed, confirmations. To construct the external edge list, if the status is 1, the transaction list is looped through and, if the isError field is 0, the to, from, value, timestamp and hash values are stored. From the input field, only the first 10 characters are kept, as these represent the methodID - the signature of the called function. In case the from field is empty, meaning the transaction is a contract creation, the contractAddress will contain the address of the new contract. To keep the edge list consistent, the address of the smart contract is stored as the target node in this instance.

For the internal edge list, the Etherscan API also returns a JSON response, for each internal transaction in the result there are the following fields: blockNumber, timeStamp, hash, from, to, value, contractAddress, input, type, gas, gasUsed, traceId, isError, errCode. The same considerations as for the external transactions regarding the status and the contract creation case remain valid. The same fields as for the external edge list are also kept for the internal list: to, from (or contractAddress), value, timestamp, and hash. There is no methodID field for internal transactions in this API.

All transaction records were obtained in the same day (May 25^{th}) for all protocols, starting from the first blockchain block until the last, sorted in descending order. Table 4.6 shows the earliest and latest date of the transactions from each obtained edge list.

[20] https://etherscan.io/, Ethereum Block Explorer

Protocol	Earliest Date	Latest Date
Maker	25.11.2017	25.05.2021
Compound	07.05.2019	25.05.2021
Aave	21.07.2020	25.05.2021
Uniswap	05.05.2020	25.05.2021
Sushiswap	04.09.2020	25.05.2021
Curve Finance	10.02.2020	25.05.2021
Balancer	28.02.2020	25.05.2021
Synthetix	10.03.2018	25.05.2021
Nexus Mutual	23.05.2019	25.05.2021
Hegic	06.08.2020	25.05.2021
RenVM	24.03.2020	25.05.2021
Harvest Finance	31.08.2020	25.05.2021
Badger	25.11.2020	25.05.2021

Table 4.6. The earliest and latest transaction dates from each protocol edge list

All the information for the required edge list is stored in pandas DataFrames and then saved as CSV files. To sum up, the columns from the resulting edge file are:

- **internal** edge list - Source, Target, Timestamp, Value (in ETH), Hash (of parent transaction), Type (internal)
- **external** edge list - Source, Target, Timestamp, Value, Type (external), MethodID, Hash, Type (external)
- **combined** edge list - Source, Target, Timestamp, Value, Type, MethodID (0x for internal calls), Hash

Once these lists are created, they usually contain many repeating transactions, from the same source to the same target, which could be easily aggregated to reduce the size of the edge list. An aggregation script takes an edge file and aggregates the contained records based on source, target, type and methodID. The value filed is summed up and a new column is generated to keep track of the transaction count. Therefore, the columns of the aggregated, and final, edge list are: Source, Target, Type, MethodID, Frequency.

Node list

Whenever the first script is run, along with the unaggregated edge list, a node list is compiled as well. The node list contains the addresses found to be either the source or target within the edge list. We know that the smart contracts from the seed data, not necessarily all of them, will also be part of the node list. Therefore, in constructing the node list, we combine it with the labels, the smart contract names, from the seed data. In addition, we try to establish, for each address, whether it belongs to an EOA or CA (see section 2.7.1). This is done by calling the *get_code* method of the Geth client, which returns the code contained at a given Ethereum address. For smart contracts, the returned

value should be different than 0. However, we know this approach can sometimes label CAs as EOAs in case the smart contract was not deployed yet or it has called the selfdestruct method.

To sum up, the node list contains the following columns: Id (Ethereum address), Name (taken from seed data), Label (EOA or CA), isDefi (TRUE for seed data addresses).

4.4 DeFi Network Construction

For easy data querying, a graph database, namely Neo4j, was prefered. All aggregated edge lists, together with the corresponding nodes, were loaded in a local graph database. One major advantage of graph databases is easy data modelling, with a focus on the relationships between entities. Since we have the source and target field in the edge list, the resulting graph will be a directed one.

In Neo4j, nodes and relationships can have attached labels and properties, so that searching for specific patterns in a SQL-like style is possible. Figure 4.9 shows what properties each type of node and relationship contains, while figure 4.10 displays a small sample of how the graph looks like in Neo4j Browser. As observed, there are two categories of nodes:

- **Protocol** - there are 13 such nodes, one for each protocol, with the type being the respective protocol category: lending, dex, derivatives or assets and the name being the actual protocol name

- **Node** - there are 274107 normal nodes, from the nodes list for each protocol. The id is autogenerated by Neo4j, the name is non-null just for the 'known' CAs from seed data, the address is the Ethereum address, the label is either CA or EOA and the isDefi property is set to true for the seed data addresses

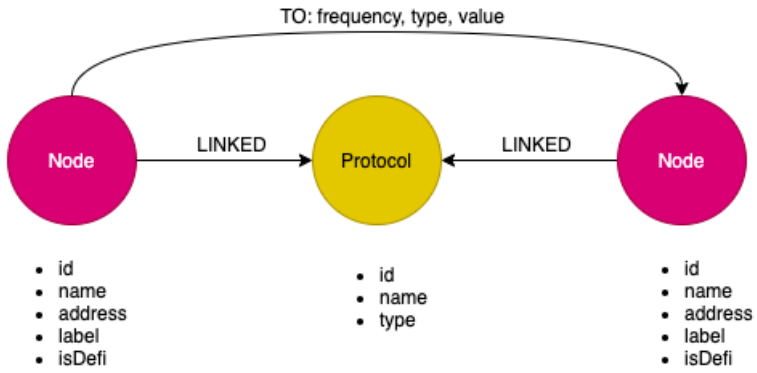


Figure 4.9. Neo4j data model

In terms of relationships, again there are two types:

- **LINKED** - no other property is set for this type. Each Node is linked to at least one Protocol depending in which node lists the respective address is found. For example, an EOA who only interacted with Maker once, will have one outgoing edge to Maker Protocol node, while a Uniswap pair smart contract, used by both Synthetix and Aave, would have three LINKED relationships. There are 369924 LINKED relationships
- **TO** - this relationship models exactly the aggregated transactions from the edge lists. Going from the source Node to the target one, with the frequency representing the number of transactions between the two nodes and the value being summed up across all aggregated transactions. The type is either external or internal. There are 656887 TO relationships

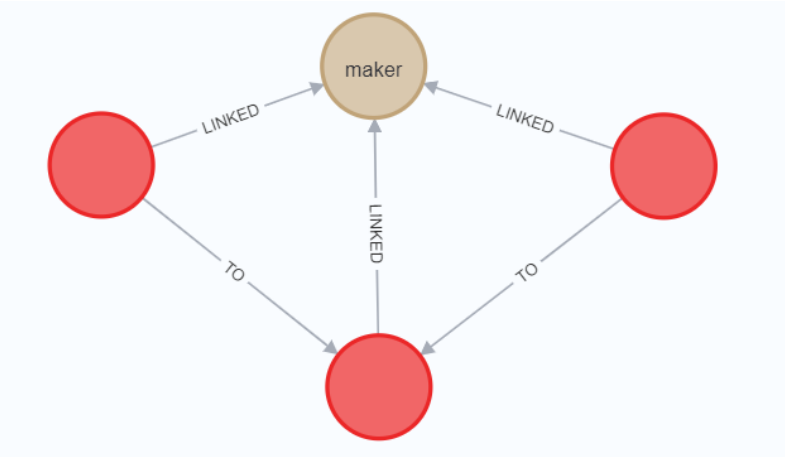


Figure 4.10. Actual Neo4j graph display

4.5 Network Analysis

This section focuses on what links and interactions exist between the chosen DeFi protocols by inspecting the network constructed at the previous step.

4.5.1 DeFi Composability

One of the most praised feature of DeFi is its composability, where different pieces an be easily integrated with each other, led to the famous analogy of Lego parts. Thanks to how smart contracts interact with each other, grouping multiple methods calls, each of them calling different protocols, can be easily done in one transaction. There are services, like Furucombo[21], where users can simply drag and drop blocks representing certain interactions to create a complex strategies. For example, taking advantage of price difference on different platform can yield a profit and it can be done on Furucombo with 4 blocks:

[21] https://furucombo.app/combo

- borrow 100 DAI from platform X through a flashloan
- swap 100 DAI for 12 SNX on platform Y
- swap 12 SNX for 110 DAI on platform Z
- repay the flash loan of 100 DAI on platform X

The multitude of platforms and services allows for a huge number of combinations. Therefore, we will only try to detect simple combinations based on the internal transactions present in the network. Since we only have the ones where an ETH transfer occured, the eventual strategies that can be discovered are limited to the ones involving ETH.

4.5.2 Manual Exploration

Table 4.7 shows how many EOAs and CAs are linked to each protocol. These numbers are an indication of which protocols are most used. We see a higher number of EOAs than CAs, this is the expected result since it reflects the sheer amount of people interacting with DeFi protocols. Another observation comes from the high number of smart contracts linked to the Maker protocol. At a closer look, after randomly sampling 50 smart contract addresses and checking them on Etherscan, we see a pattern emerging. 37 of the addresses have the label 'DSProxy' on Etherscan. Also, all of them have been been deployed by the same address, which turns out to be one from the Maker seed data, a proxy factory.

A factory smart contract is one which has the responsibility of creating other smart contracts that share similar features. It turns out this is a very common pattern used by developers. In Maker, whenever a user interacts for the first time with one of the protocol's applications, a DSProxy[22] smart contract is deployed for him, this is called a profile proxy. There is also a forwarding proxy in Maker which is used for grouping together multiple actions and executing them as a single transaction.

Another insight from Table 4.7 comes form the huge number of addresses linked to Uniswap. This is consistent with the popularity of Uniswap and its role at the core of DeFi ecosystem, as token swaps play a major role.

Having the DeFi network in Neo4j allows querying for certain patterns. More specifically, we first looked for direct interactions between smart contract belonging to different protocols, keeping in mind the limited overview due to only having the internal transaction which carry value. This only yielded intra-protocol results, not inter-protocol as we have hoped. Thus, we extended the query and looked for subnetworks of the form displayed by Figure 4.11.

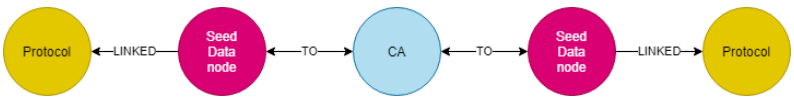


Figure 4.11. Query template for smart network network with one hop

[22] https://docs.makerdao.com/build/dai.js/advanced-configuration/using-ds-proxy

Protocol	Type	# LINKED	# EOAs	# CAs
Maker	Lending	41020	22808	18212
Compound	Lending	41794	41155	639
Aave	Lending	24677	24570	107
Uniswap	DEX	110496	102911	7585
Sushiswap	DEX	17130	16126	1004
Curve Finance	DEX	40898	40446	452
Balancer	DEX	9902	6705	3197
Synthetix	Derivatives	28263	28025	238
Nexus Mutual	Derivatives	7265	7231	34
Hegic	Derivatives	2393	2350	43
RenVM	Assets	11803	11782	21
Harvest Finance	Assets	12916	12477	439
Badger	Assets	21367	21271	96

Table 4.7. The number of EOAs and CAs linked to each protocol

The result is a subgraph of the original network, containing 97 nodes and 246 edges, 12 of these nodes are from the original seed data. Figure 4.13 shows the complete subnetwork[23]. Two labels were added, Compound Ether and Proxy Factory.

Compound Ether or cEther is the smart contract used for supplying ETH to Compound. It functions similarly to the cTokens contracts, but since ETH is not an ERC-20 token, it needs a special contract. Figure 4.12 shows the methods called in the process of supplying ETH. In the mint() method call, ETH is transfered to the cEther contract and the sender receives back cETH.

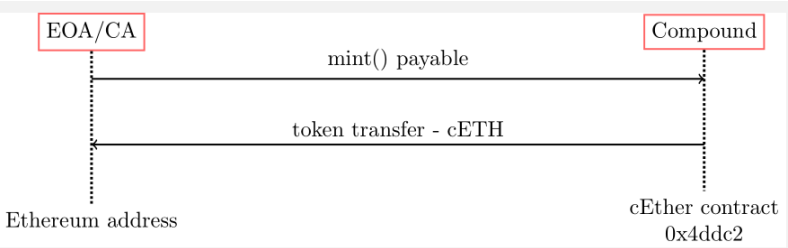


Figure 4.12. Function calls for supplying ETH to Compound

The Proxy Factory, as discussed above, is deploying profile proxies for each Maker user. These proxies can later interact with other contracts, on behalf of the user, to atomically execute a group of actions. Here, we see the interactions between the profile proxies and the cEther contract.

[23]Displayed using Neo4j Bloom

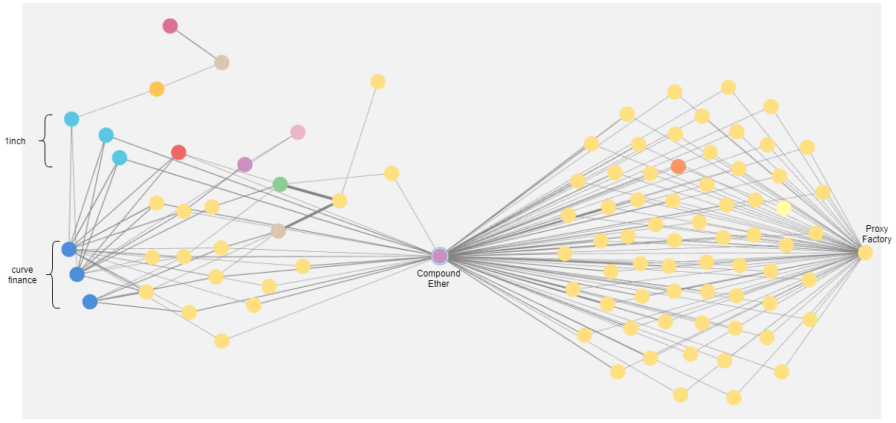


Figure 4.13. Complete smart contract subnetwork with one hop

After carefully inspecting the smart contracts from the subgraph by looking at their details on Etherscan, three contracts belonging to another DeFi protocol were identified. These are colored with light-blue and are part of 1inch[24], which is a decentralized exchange. The edges between Compound Ether and the 1inch nodes are another proof of the DeFi composability.

The dark blue nodes belong to Curve Finance. They are liquidity pools where users can deposit sETH, stETH and ankrETH[25] for yield farming. We see again how easily tokens from one platform are leveraged for profit on another one.

4.5.3 Network Metrics

Visualizing the network gives us only a qualitative description of it. Network metrics provide a mathematical overview of the network structure, a quantitative outline. The problem of centrality, how important nodes are within a network, is of particular interest [32]. There are quit a few measures of centrality, however, we will only focus on degree centrality and betweenness centrality.

For computing the metrics, the smart contracts subnetwork, 97 nodes and 246 edges, is loaded into Python's Networkx[26] package to create a weighted directed graph. The metrics are computed for all 97 nodes, but the focus is on the seed data smart contracts. Table A.2 contains the detailed result for all 12 seed data smart contracts. A brief overview of what these smart contracts do:

- **WETH Gateway** - Aave's smart contract for wrapping Ethereum. An address can send ETH to this address and receive wETH, an equivalent ERC-20 token which can be used in DeFi applications

[24] https://1inch.io/

[25] These are ERC-20 tokens received by users from other protocols in exchange for locking their ETH

[26] https://networkx.org/

- **Uniswap Router** - an intermediary smart contract used to interact with a liquidity pool. Among others, it contains methods for adding and removing liquidity, and swapping tokens
- **Hegic WBTC Options** - used to buy put and call options on WBTC. The contract uses Uniswap to swap ETH to WBTC
- **Maximillion** - a Compound utility contract for repaying cEther borrows
- **Depot** - Synthetix' smart contract for exchanging ETH for sUDS and users having sUSD to deposit it
- **devMultisig** - used in Badger's governance. It keeps contract upgradability rights, can change products parameters, manage permissions and control the treasury
- **Proxy Actions** - one of the wrappers used by Maker. It contains a set of proxy functions that can be used via the DSProxy profile smart contract

The three Curve finance smart contracts, together with the Proxy Factory smart contract from Maker and Compound Ether, were discussed in the previous sections.

The degree centrality of a node is the same as the node's degree, namely the number of direct connections with other nodes. In the context of a directed graph, is formed of the number of incoming edges (in-degree) and the number of outgoing edges (out-degree). Figure 4.14 shows both of these degrees. It is a very intuitive measure of centrality. Since the weights of the edges represent actually the total number of transactions between the source and the target node, the degree computation takes into consideration the weights. We see that, in general, the out-degree is much higher than the in-degree. This means contracts such as WETHGateway, UniswapV2Router and HegicWBTC Options are the source of more transactions that send out Ether than the target of incoming transfers.

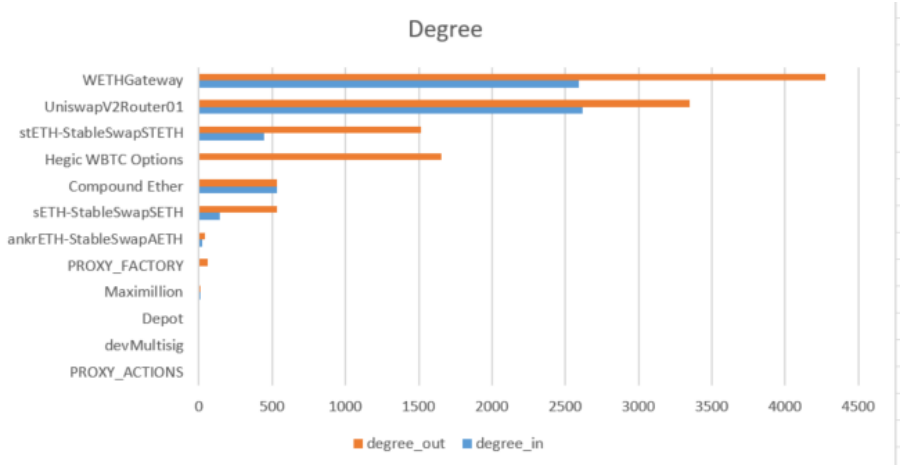


Figure 4.14. Smart contract subnetwork degree

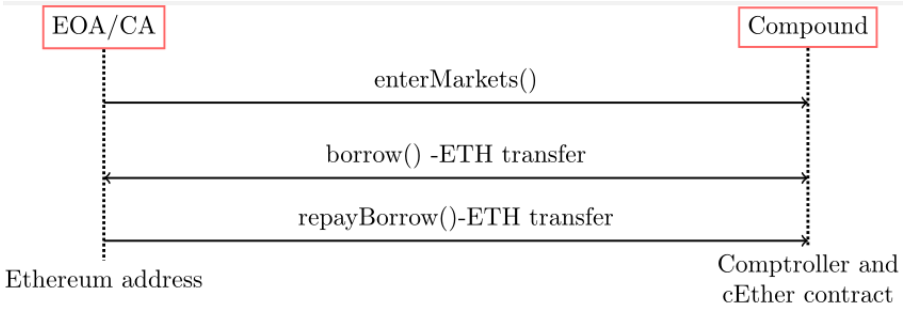


Figure 4.15. Function calls for borrowing ETH from Compound

From the documentation of WETH Gateway[27] we see there are four methods: depositETH(), withdrawETH(), borrowETH() and repayETH(). The first and last methods are responsible for the incoming Ether transfers. The deposit function is used to send Ether to the contract and get back an equivalent amount of WETH tokens. While the repay method allows a borrower to repay his WETH borrow by sending ETH which is then wrapped to WETH and the debt position closed. The second and the third methods are the ones where ETH is transferred out. Withdrawing, in this case, will deduct the amount from the WETH balance of the specified address, unwrap it, and send the Ether to the given address. For borrowing, a debt in WETH is created, the borrowed amount of WETH is unwrapped and sent out. We can conclude that, in the analyzed transactions, there were more borrows/withdrawals than deposits/repays.

The UniswapV2Router01[28] is the first version of the router, in the meantime, a new version became available and the recommended one[29]. Router01 contains methods for adding/removing ETH liquidity and for swapping ETH for tokens or tokens for ETH. From the usage of addLiquidityETH() and swapETHforExactTokens() we observe the incoming transactions to Router01, while the removeLiquidityETH() and swapExactTokensForETH() are performing the reverse operations and result in the outgoing transactions. Again, from the out-degree, the out-flow of ETH is sticking out in this case as well.

An interesting observation is the almost equal in- and out-degrees for Compound Ether. Minting cEther tokens requires the transfer of ETH to the contract and is done by calling the mint() method of the cEther contract (see 4.12), while redeem() and redeemUnderlying() methods are used to retrieve the supplied Ether and, therefore, responsible for some of the outgoing flow of Ether. There is also the possibility to borrow Ether from the Compound Ether contract, however before being able to take a loan, users have to supply other form of collateral to Compound and mark the supplied assets as collateral by calling a method called enterMarkets() of the Comptroller[30] smart contract. Then, they can call the borrow() method of the cEther contract to receive the loan. The borrow() calls

[27] https://docs.aave.com/developers/the-core-protocol/weth-gateway

[28] https://docs.uniswap.org/protocol/V2/reference/smart-contracts/router-01

[29] https://docs.uniswap.org/protocol/V2/reference/smart-contracts/router-02

[30] https://compound.finance/docs/comptroller

contribute to the outgoing edges and the complementary method, repayBorrow(), brings the Ether back to the cEther smart contract. Figure 4.15 displays the borrowing process.

Betweenness centrality is a measure that shows the incidence of a node being on the shortest paths between other nodes. In a social network, high betweenness centrality shows the importance of a node in the information flow [16]. If we adapt this concept to our smart contract network, we can think about the flow of ETH, transferred from one smart contract to another. Compound Ether has the highest betweenness centrality in this case and we can interpret it as it being a major hub in the flow of ETH in the network.

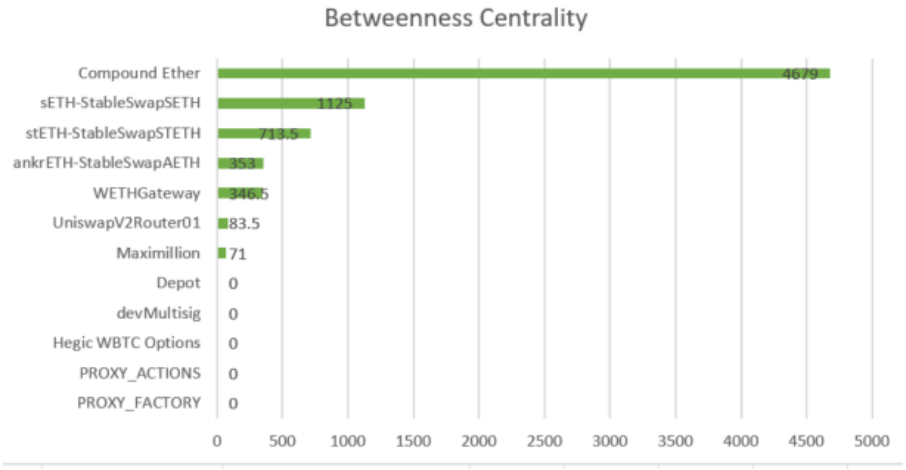


Figure 4.16. Smart contract subnetwork betweenness centrality

Two other metrics, which are computed for the overall subnetwork (figure 4.11), are reciprocity and assortativity. Reciprocity characterizes the likelihood of nodes to be mutually linked and assortativity measures whether, in general, high-degree nodes are linked to other high-degree nodes, while low-degree ones are connected to other low-degree nodes. In this case, we say the network is assortative - the metric tends to 1. If the opposite situation is present, high-degree nodes connected to low-degree ones, and vice-versa, the network is said to be disassortative - the value tend to -1 [23]. In our case, the two metrics were computed with the same python package and the values are the following:

- reciprocity: 0.4552
- assortativity: -0.6118

In their comprehensive study of the entire Ethereum blockchain[23], Lee et al. explored these two metrics as well and determined that the reciprocity is the highest for the smart contract network as it shows the contracts' high reliance on one another. In terms of assortativity, the result is consistent as well with Lee's observations that disassortativity is usually present in smart contract networks since there are some smart contracts which are used by a large majority of other contracts, regardless of the latters degree. This is especially true in DeFi, where there are smart contracts from established protocols with which anyone can interact.

Chapter 5

Discussion

5.1 Summary of key findings

Exploring DeFi protocols, even though a limited number, offered great insights into what kind of financial products are available on decentralized applications at this stage. Understanding this complex and intricate ecosystem is the first and most important step of any analysis. Each protocol's use case was explained, with a focus on the ones from the lending and decentralized exchanges categories, to build a consistent background for the analysis.

Even at this first phase of the study, we were able to identify some of the key smart contracts, such as Compound Ether, Aave's WETh Gateway and Uniswap's Router01, and confidently say they play a major role in the subsystem of analyzed protocols, at least in the flow of ETH. From the analysis of the in- and out- degrees for the smart contracts present in the seed data, we could map some of the edges to blockchain interactions. The fact that the networks only showed the internal transactions with Ether value was, on one side a limitation, however, it allowed for a more precise interaction determination.

The reciprocity and assortativity values for the network were consistent with results from a much larger scale study. Moreover, the disassortative character of the network is a strong argument for decentralization, where any smart contract can be be called by any other Ethereum address.

5.2 Limitations

It was already mentioned that the available internal transactions from the network represent only a small part of the interactions between contracts, due to the Etherscan API limitations for internal transactions. Therefore, the provided overview is incomplete and does not show the true complexity protocol interactions, showing only transactions where Ether was transferred. A more comprehensive study would also have to consider more protocols. Moreover, the data collection procedure for the seed data was not as rigorous as it should have been. A better approach would be to directly contact the protocol developers and get a list of the deployed contract directly from them.

For the analysis, the obtained weighted subnetwork does not exactly model the transactions themselves. There is no way of establishing whether three nodes being on the same

path have been involved in the same transaction. It can only be stated that, at some point, there was an interaction between the first and the second SC, as well as between the second and the third.

5.3 Possible Future Work

This small-scale analysis represented the first phase of a more comprehensive study. One of the goals of this phase was to establish a methodology and provide a more streamlined process for the next phase. Of course, the available data represented the highest drawback here. Therefore, for the next phase, trying to find a more reliable data source is paramount. Using AIT's own Ethereum node, set on archive mode, proved to give inconclusive results when queried for the internal transactions triggered by a given set of external or parent transactions, identified by their transaction hash. This fact was due to how the node was synchronizing on the blockchain network, as the archive mode was only recently set, and there were still missing blocks.

As it turned out, one fo the team's collaborators had all the blockchain data from January to May 2021 and we can use it for the next phase of the study. Just to get a glimpse of the huge amount of data, the archive containing the internal transactions has 38.67 GB. Of course, this brings more obstacles to overcome in terms of infrastructure, parsing algorithms, storage capacity, etc.

Chapter 6

Conclusion

This work started from the premise of exploring the intricate space of decentralized finance, a new and exciting area of blockchain applications. This task proved to be more challenging than initially thought. With only the Etherscan data, we could barely scratch the surface of the network interactions, especially between smart contracts. However, even this limited amount of data provided a coherent and insightful initial image of what can be achieved with more available data. We could clearly determine specific paths and interactions in the network, while also gaining an understanding of some of the most important services offered by these protocols.

One should not venture to say that DeFi can replace centralized finance at this point, but it surely is on the right path. The barrier to entry for DeFi is much lower than what it can be seen in traditional financial markets and this led to more than 235 DeFi projects, with 219 being build on Ethereum, listed on DeFi Prime[1], in a span of just a few years.

[1] https://defiprime.com/

List of Figures

List of Tables

Bibliography

[1] Hayden Adams, Noah Zinsmeister, and Dan Robinson. "Uniswap v2 Core. 2020". In: (Mar. 2020). URL: https://uniswap.org/whitepaper.pdf.

[2] Ian Allison. "DeFi Major Aave Working With Polygon to Bypass Ethereum Congestion". In: *Nasdaq* (Mar. 2021). URL: https://www.nasdaq.com/articles/defi-major-aave-working-with-polygon-to-bypass-ethereum-congestion-2021-03-31.

[3] Hendrik Amler et al. "DeFi-ning DeFi: Challenges & Pathway". In: *arXiv preprint arXiv:2101.05589* (2021).

[4] Andreas M Antonopoulos and Gavin Wood. *Mastering ethereum: building smart contracts and dapps*. O'reilly Media, 2018.

[5] Adam Back. "Hashcash. 1997". In: (1997). URL: http://www.cypherspace.org/hashcash.

[6] Dave Bayer, Stuart Haber, and W Scott Stornetta. "Improving the efficiency and reliability of digital time-stamping". In: *Sequences Ii*. Springer, 1993, pp. 329–334.

[7] Lucas Campbell. "DeFi Wallets - Best Crypto Wallets for Decentralized Finance". In: *DeFi Rate* (Jan. 2021). URL: https://defirate.com/wallet/.

[8] David Chaum. "Blind signatures for untraceable payments". In: *Advances in cryptology*. Springer. 1983, pp. 199–203.

[9] Richard Chen. "A Comparison of Decentralized Exchange Designs". In: *Medium* (Apr. 2019). URL: https://thecontrol.co/a-comparison-of-decentralized-exchange-designs-1deef249f56a.

[10] Simon Cousaert, Jiahua Xu, and Toshiko Matsui. "SoK: Yield Aggregators in DeFi". In: *arXiv preprint arXiv:2105.13891* (2021).

[11] Wei Dai. "B-Money-an anonymous, distributed electronic cash system". In: (1998).

[12] Philip Daian et al. "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability". In: *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2020, pp. 910–927.

[13] Glyn Davies. *History of money*. University of Wales Press, 2010.

[14] Asli Demirguc-Kunt et al. *The Global Findex Database 2017: Measuring financial inclusion and the fintech revolution*. World Bank Publications, 2018.

[15] Michael Egorov. "StableSwap-efficient mechanism for Stablecoin liquidity". In: *Retrieved Feb* 24 (2019), p. 2021.

[16] Jennifer Golbeck. "Chapter 21 - Analyzing networks". In: *Introduction to Social Media Investigation*. Ed. by Jennifer Golbeck. Boston: Syngress, 2015, pp. 221–235. ISBN: 978-0-12-801656-5. DOI: https://doi.org/10.1016/B978-0-12-801656-5.00021-4. URL: https://www.sciencedirect.com/science/article/pii/B9780128016565000214.

[17] L. Gudgeon et al. "The Decentralized Financial Crisis". In: *2020 Crypto Valley Conference on Blockchain Technology (CVCBT)*. 2020, pp. 1–15. DOI: 10.1109/CVCBT50464.2020.00005.

[18] Stuart Haber and W Scott Stornetta. "How to time-stamp a digital document". In: *Conference on the Theory and Application of Cryptography*. Springer. 1990, pp. 437–455.

[19] Campbell R Harvey, Ashwin Ramachandran, and Joey Santoro. "DeFi and the Future of Finance". In: *Available at SSRN 3711777* (2020).

[20] Johannes Rude Jensen, Victor von Wachter, and Omri Ross. "How Decentralized is the Governance of Blockchain-based Finance: Empirical Evidence from four Governance Token Distributions". In: *arXiv preprint arXiv:2102.10096* (2021).

[21] Leslie Lamport, Robert Shostak, and Marshall Pease. "The Byzantine generals problem". In: *Concurrency: the Works of Leslie Lamport*. 2019, pp. 203–226.

[22] Patrick Laurent et al. "The tokenization of assets is disrupting the financial industry. Are you ready". In: *Inside magazine* 19 (2018), pp. 62–67.

[23] Xi Tong Lee et al. "Measurements, analyses, and insights on the entire ethereum blockchain network". In: *Proceedings of The Web Conference 2020*. 2020, pp. 155–166.

[24] Robert Leshner and Geoffrey Hayes. "Compound: The money market protocol". In: *White Paper* (2019).

[25] Alex Lielacher. "Hot Wallets vs Cold Wallets: What's the Difference?: CoinMarketCap". In: *RSS* (Mar. 2021). URL: https://coinmarketcap.com/alexandria/article/hot-wallets-vs-cold-wallets-whats-the-difference.

[26] Bowen Liu, Pawel Szalachowski, and Jianying Zhou. "A first look into defi oracles". In: *arXiv preprint arXiv:2005.04377* (2020).

[27] Peter Ludlow. "A Cypherpunk's Manifesto". In: *Crypto Anarchy, Cyberstates, and Pirate Utopias*. 2001, pp. 81–83.

[28] MakerDAO. "The Maker Protocol White Paper: Feb 2020". In: *The Maker Protocol White Paper | Feb 2020* (Feb. 2020). URL: https://makerdao.com/en/whitepaper/.

[29] Fernando Martinelli and Nikolai Mushegian. "A non-custodial portfolio manager, liquidity provider, and price sensor". In: (2019). URL: https://balancer.finance/whitepaper.

[30] Du Mingxiao et al. "A review on consensus algorithm of blockchain". In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2017, pp. 2567–2572. DOI: 10.1109/SMC.2017.8123011.

[31] Satoshi Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". In: *Decentralized Business Review* (2008), p. 21260.

[32] Mark Newman. "Measures and metrics". In: *Networks*. Oxford University Press, 2010.

[33] Cong T Nguyen et al. "Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities". In: *IEEE Access* 7 (2019), pp. 85727–85745.

[34] Kasey Panetta. "The 4 phases of the Gartner blockchain Spectrum". In: *Smarter With Gartner* (Oct. 2019). URL: https://www.gartner.com/smarterwithgartner/the-4-phases-of-the-gartner-blockchain-spectrum/.

[35] Daniel Perez et al. "Liquidations: DeFi on a Knife-edge". In: *arXiv preprint arXiv:2009.13235* (2020).

[36] Kaihua Qin et al. "CeFi vs. DeFi–Comparing Centralized to Decentralized Finance". In: *arXiv preprint arXiv:2106.08157* (2021).

[37] Team RenVM. "A privacy preserving virtual machine powering zero-knowledge financial application". In: (2019). URL: https://renproject.io/litepaper.pdf.

[38] Fabian Schär. "Decentralized finance: On blockchain-and smart contract-based financial markets". In: *FRB of St. Louis Review* (2021).

[39] Martin Schmalz. "One Big Reason There's So Little Competition Among U.S. Banks". In: *Harvard Business Review* (Oct. 2017). URL: https://hbr.org/2016/06/one-big-reason-theres-so-little-competition-among-u-s-banks.

[40] Don Tapscott. "Token Taxonomy: The Need for Open-Source Standards Around Digital Assets". In: *Blockchain Research Institute, 19 Feb* (Feb. 2020).

[41] Don Tapscott and Alex Tapscott. *Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world*. Penguin, 2016.

[42] Synthetix Team. "Litepaper". In: *Synthetix System Documentation* (Mar. 2020). URL: https://docs.synthetix.io/litepaper/.

[43] Dabao Wang et al. "Towards A First Step to Understand Flash Loan and Its Applications in DeFi Ecosystem". In: *Proceedings of the Ninth International Workshop on Security in Blockchain and Cloud Computing*. 2021, pp. 23–28.

[44] Victor Wilfred. "Cryptonomics: What Are Elastic Supply Tokens". In: *RSS* (Jan. 2021). URL: https://www.bsc.news/post/cryptonomics-elastic-supply-tokens-explained.

[45] Molly Wintermute. "Hegic: On-chain Options Trading Protocol on Ethereum Powered by Hedge Contracts and Liquidity Pools". In: (2020). URL: https://github.com/hegic/whitepaper.

[46] Gavin Wood et al. "Ethereum: A secure decentralised generalised transaction ledger". In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32.

[47] Dylan Yaga et al. "Blockchain technology overview". In: *arXiv preprint arXiv:1906.11078* (2019).

[48] Taiyang Zhang. "The REN Crypto Platform: Asset Tokenization". In: *Gemini* (June 2021). URL: https://www.gemini.com/cryptopedia/ren-token-and-renvm-crypto-platform.

Appendix A

Protocol	**Source**
Maker	https://changelog.makerdao.com/ - release 1.2.7
Compound	https://github.com/compound-finance/compound-config/blob/master/networks/mainnet.json
Aave	https://docs.aave.com/developers/v/2.0/deployed-contracts/deployed-contracts
Uniswap	https://uniswap.org/docs/v2/smart-contracts/factory https://docs.google.com/spreadsheets/d/1jKEhOi9gIcM9bKdn7rgJEK0RKpzbE1k6bPy_kJW75Aw/edit#gid=1860072669
Sushiswap	https://reposhub.com/python/miscellaneous/sushiswap-sushiswap.html
Curve Finance	https://curve.readthedocs.io/ref-addresses.html
Balancer	https://docs.balancer.finance/smart-contracts/addresses
Synthetix	https://docs.synthetix.io/addresses/#mainnet-contracts
Nexus Mutual	https://nxm.surge.sh/
Hegic	https://hegic.gitbook.io/start/developers/contracts
RenVM	https://renproject.github.io/contracts-ts/#/mainnet
Harvest Finance	https://github.com/harvest-finance/harvest
Badger	https://badger-finance.gitbook.io/badger-finance/technical/contracts

Table A.1. Seed data sources

Address	Label	Protocol	Deg	Deg in	Deg out	Betweenness cen
0x82ecd135dc...	PROXY_ACTIONS	maker	1	0	1	0
0xe1f64079ad...	Depot	synthetix	2	0	2	0
0x3961245db6...	Hegic WBTC Options	hegic	1657	0	1657	0
0xb65cef03b9...	devMultisig	badger	2	2	0	0
0xf859a1ad94...	Maximillion	compound	20	12	8	71
0xf164fc0ec4...	UniswapV2Router01	uniswap	5961	2615	3346	83.5
0xdcd33426ba...	WETHGateway	aave	6863	2591	4272	346.5
0xa96a65c051...	ankrETH-StableSwapAETH	curvefinance	68	24	44	353
0xc5424b857f...	sETH-StableSwapSETH	curvefinance	675	144	531	1125
0xdc24316b9a...	stETH-StableSwapSTETH	curvefinance	1960	447	1513	713.5
0xa26e15c895...	PROXY_FACTORY	maker	63	0	63	0
0x4ddc2d1939...	Compound Ether	compound	1070	536	534	4679

Table A.2. Network Metrics