Identifying Key Non-Financial Risks in Decentralised Finance on Ethereum Blockchain

Thesis · October 2020

DOI: 10.13140/RG.2.2.27175.57769

1 author:

Xavier Meegan
Politecnico di Milano
1 PUBLICATION **0** CITATIONS

SEE PROFILE

Identifying Key Non-Financial Risks in Decentralised Finance on Ethereum Blockchain

by

Xavier Meegan

Submitted to MIP Politecnico di Milano
in partial fulfillment of
the requirements for the degree of
Master of Fintech

MIP Politecnico di Milano

October 2020

Identifying Key Non-Financial Risks in Decentralised Finance on Ethereum Blockchain

Xavier Meegan

Fintech, Master's Thesis, 2020

Thesis Supervisor: Massimo Morini

Abstract

This paper identifies twelve key non-financial risks in decentralised finance (DeFi) on *Ethereum* blockchain. Innovation is ubiquitous in DeFi, yet definition of risks of that innovation lags behind. An investor using DeFi needs to be informed about both financial and non-financial risks in order to assume risk for a given return. Financial risks are well known in traditional finance and these same financial risks can be found in DeFi. What separates DeFi from traditional finance is the non-financial risks it is exposed to, as a result of it operating on the underlying blockchain, *Ethereum*. Based on extensive research, attendance at conferences, participation in the DeFi community and conducting interviews with several influential figures in DeFi, I have identified twelve key non-financial risks in decentralised finance on Ethereum blockchain. I consolidated my research into the following non-financial risks existent in DeFi on *Ethereum* blockchain: *scalability risk, smart contract vulnerability risk, oracle risk, design risk, composability risk, centrality risk, economic incentive risk, financial illiteracy risk, regulatory risk, finality risk, disclosure risk and the risk of more risks.* This work aims to build the foundations for future non-financial risk management work in DeFi.

Acknowledgements

A special thank you to my supervisor, Professor Massimo Morini, for accepting to assist me for the duration of this work, especially during such turbulent times through COVID-19.

A thank you to all the DeFi experts that took the time out to answer my questions. In particular to Jordan Lyall and Jack Clancy from Consensys, Felix Lutsch from Chorus One, Paul Salisbury and Donn Krassiyenko from Techemy Capital, Hugh Karp from Nexus Mutual and Marouane Hajji from Unslashed.

A thank you to my incredible parents and in particular to my Mum for being the grammar queen I so desperately needed to publicly release this work.

Finally, a thank you to Mariana Gomez de la Villa for being so incredibly supportive about getting this work public and to Gamze Tillem for her assistance in formatting.

Contents

About

The subject of this Master Thesis is "Identifying Key Non-Financial Risks in Decentralised Finance on Ethereum Blockchain". This Thesis is undertaken as part of Xavier Meegan's Master of Fintech at MIP Politecnico, completed by the student September 2020. This Master thesis was supervised by Prof. Morini, a member of Algorand Foundation's Economic Advisory Committee and Head of Rates and Credit Modelling / Coordinator of Model Research, at IMI bank. Prof. Morini has been Advisor and Trainer at the World Bank, the Monetary Authority of Singapore and several private and public financial institutions. Prof Morini authored the first articles proposing blockchain and smart contract solutions for financial instruments, often quoted by US regulators. Prof. Morini led the development of smart contracts for collateralized derivatives with trusted computation on Ethereum. In the field Prof. Morini researches on decentralized financial market infrastructures, layer-two solutions and stability mechanisms and his work on blockchain has featured on Coindesk, Bitcoin Magazine, Harvard Business Review. Prof. Morini provided supervision to Xavier Meegan by consultation for the extent of the period of writing of this thesis. Professor Morini is employed by multiple financial entities, however Professor Morini's opinions were his own, rather than that of any company he has worked for. Professor Morini's extensive background in risk management, technology transformation and decentralised financial market infrastructure made for a perfect match of expertise in the formulation of this thesis. Professor Morini guided, rather than directed the student on this topic, and he provided supervision and advice in support of Xavier Meegan's original research and ideas. Xavier Meegan is solely responsible for all qualitative research undertaken and for the views expressed in this Thesis on the subject of current risks in DeFi.

Chapter 1

Introduction

Decentralised Finance Applications (herein called "DeFi") has taken the world by storm since its inception in 2017. DeFi is the transformation of traditional financial products into products that operate without an intermediary via smart contracts on a blockchain. The value of DeFi specific to the Ethereum blockchain, has grown from $4 on August 2017, to $7,820,000,000 as of time of writing September 7, 2020. [52] Types of DeFi applications are wide-ranging. Some popular DeFi applications include lending, stablecoins, decentralised exchanges (DEXs), derivatives, synthetic assets, insurance and asset management. DeFi is lucrative compared to its counterpart of traditional finance as it offers yields (returns) unobtainable in traditional finance. In the past year especially, DeFi has enjoyed a meteoric rise, grabbing headlines as it has seen yields obtained by investors in excess of 1000% annualised per year, just by depositing tokens into protocols. This is in comparison to barely above 0% that can be earned by simply depositing cash into a bank. In traditional finance, investors can expect higher returns on financial instruments that are considered to be risky. Cryptocurrency, which lives on a blockchain, has been considered to be an extremely risky asset. Whereas cryptocurrency was once used as a store of value or utility token, it is now being used as something that can help governance of DeFi protocols. For example, a DeFi protocol such as a lending platform generates a fee every time an investor wants to borrow from a liquidity pool, that fee can be distributed directly back to the token holder (governor). For the first time in the short history of cryptocurrencies, we are seeing tokens that can be properly valued based on revenue and future growth, the same valuing mechanism seen in stocks. However,

it can be difficult to value a DeFi protocol, especially because of how inherently risky its usage is. It is not uncommon for a project to receive $500 million in funding in two days, only to have a bug in the code and for investors to lose everything. [27] At present, DeFi is like the wild west, with investments being made without proper risk management being taken into consideration. Based on qualitative research, this paper identifies twelve non-financial key risks that every DeFi investor should be aware of before considering using DeFi protocols or investing in a DeFi token. The purpose of this paper is to comprehensively define Ethereum blockchain systemic risks in decentralised finance as it is an area of risk that directly affects DeFi, which is nascent and not well yet consolidated and described. The twelve non-financial risks in DeFi on Ethereum blockchain are: *scalability risk, smart contract vulnerability risk, oracle risk, design risk, composability risk, centrality risk, economic incentive risk, financial illiteracy risk, regulatory risk, finality risk, disclosure risk and the risk of more risks.*

Chapter 2

Background

On March 12 2020, in an event now known in the blockchain / cryptocurrency community as 'Black Thursday', many cryptocurrencies suffered value losses between 40-50% from investors having a negative outlook on future market conditions, leading to a quick and fast initial sell-off which then caused many leveraged positions to be liquidated from the value of their collateral not meeting requirements for borrowing, causing an even bigger sell-off and even bigger drop in cryptocurrency prices. When a leveraged position is liquidated, the collateral posted by the borrower is sold immediately on the market. Therefore, an initial sell-off from spot investors from fear of future market conditions that could be attributed to their outlook on how COVID-19 could affect future market conditions was then amplified by leveraged investors becoming liquidated from the value of their collateral dropping simultaneously, creating even more sell orders of cryptocurrencies on the market. 'Black Thursday' exposed the infrastructure of the underlying blockchain of cryptocurrencies, as whilst investors were trying to exit positions or add more collateral to ensure they would not be liquidated, they experienced network congestion and therefore could not transact on the blockchain fast enough. The result of dramatic drops in price of popular cryptocurrency assets such as Bitcoin and Ethereum led many investors to attempt to exit positions or rush to add more collateral to leveraged positions, only to find out they could not sell their assets or post more collateral because there was too much activity on the network, leaving investors stuck with failed transactions, holding assets that they failed to sell or having leveraged positions closed (why is this important). A core mechanism of the blockchain is its

consensus, or in other words, how a distributed network of permissionless nodes reach an agreement without the involvement of a third-party intermediary on what transactions are valid, which in turn is then finalised and then published on the blockchain. During 'Black Thursday', Bitcoin and Ethereum both used *'Proof-of-Work'* as their consensus, an algorithm that rewards *'miners'* for asserting that transactions on the blockchain are correct. *Miners* play an important role in *Proof-of-Work* blockchains, as blockchains such as Bitcoin, have a maximum block size that limits the maximum amount of transaction data that can be added to a block. Because there is a limit on how many transactions can be included in a single block, cryptocurrency senders wishing to transact must bid in order for their transactions to be successfully published on the blockchain. *Miners* simply ignore transactions that have been sent with transaction fees that are too low and mine transactions with higher fees. When there is high network activity (e.g. demand due to the native crypto-asset rising or panic-selling due to macroeconomic events as was the case with COVID-19), *miners* require higher transaction fees as there is more demand to use the network. 'Black Thursday' exposed cryptocurrency investors, as investors that were trying to sell off their stake in cryptocurrencies could not sell their assets fast enough and leveraged investors could not add more collateral to their leveraged positions due to the extremely high transaction fees required. By the time investors realised their transactions had not gone through (due to being ignored by *miners* from sending fees that were too low and therefore disregarded in favour of transactions that were sent with higher fees), the price of almost all cryptocurrencies had already dropped 40-50%. It is important to note that cryptocurrency markets were not the only markets to suffer heavy losses on March 12, 2020, as other commodity and stock markets also posted significant losses. One example of a market that posted significant losses on the same day that cryptocurrency prices dropped 40-50% was the US stock index, *S&P500*, which measures the stock performance of 500 large companies listed on stock exchanges in the United States, plunging 9.5% in the biggest single-day drop since 1987. [16] The huge losses across global markets on March 12, 2020 can be attributed to investors becoming aware that the novel 'COVID-19' was more than just a flu. 'Black Thursday' revealed rather brutally that, at least for the time being, those in the crypto space must acknowledge and

begin to address a correlation with traditional markets.

Within a blockchain, there can be several verticals that exist on that specific blockchain. *Ethereum* is an open-ended, decentralized, blockchain-based, public software platform that facilitates peer-to-peer contracts, known as Smart Contracts, as well as Decentralized Applications, known as DApps. [11] A core difference in *Ethereums* blockchain as opposed to Bitcoin's blockchain is the introduction of smart contracts within the native blockchain ecosystem allowing developers to build decentralised applications. Both *Bitcoin* and *Ethereum* are based on blockchains, but *Ethereums* blockchain extends the concept of a distributed ledger to enable further advanced commands and hence verticals can be created within *Ethereum* that are separated from the traditional currency 'store-of-value' property of *Bitcoin*. Ethereum has several DApps that exist on its blockchain. A *DApp* can be created and used in any industry vertical – e.g. health, energy, supply chain, gaming or social networks. More recently, there has been a surge in the creation of DApps in the financial vertical on *Ethereum*. As mentioned previously, a *DApp* is a decentralised application and therefore a *DApp* that lives in the financial vertical is itself DeFi. As cryptocurrency prices dramatically fell across the market on 'Black Thursday', the worst vertical of DApps to suffer from the collapse in cryptocurrency prices was DeFi.

In essence, DeFi financial applications run on *Ethereum* blockchain infrastructure that investors can connect to in a permissionless manner. This paper will focus on DeFi applications that solely run on the infrastructure of *Ethereum* blockchain. While most of the DeFi applications are still analogous to existing financial products from the established financial world, one can expect entirely new DeFi use cases to emerge in the future separate from traditional finance. In theory, everything that is programmable is imaginable. [29] DeFi is permissionless, composable, transparent, censorship resistant, decentralised, accessible and flexible. Anyone can participate in DeFi, there are no permissions needed to access it and it is accessible to anyone with an internet connection. No global entity can take DeFi down as long as blockchain exists, as data on the *Ethereum* blockchain is decentralised across nodes so that anyone in the world can download a copy of the ledger and begin transacting on the network. DeFi transactions are transparent as all transactions on a permissionless

blockchain are recorded publicly. All previous transactions can be downloaded and seen via clients and current transactions on *Ethereum* can be seen in the mempool (pending transactions that are awaiting to be published on the network if valid by *miners*). Another core ingredient of DeFi is its composability. Since the inception of DeFi, many in the community have coined DeFi, 'Money Legos' because of how easy it is for developers to stack existing DeFi protocols on top of each other to create new products. Developers are able to build quickly and easily in DeFi because all DeFi protocols are open-sourced, meaning their smart contract code (often written in Solidity) is available publicly for anyone to use or modify. DeFi is unique in that developers can leverage any combination of DeFi protocols together without requiring any special permissions, opening up a frictionless innovation cycle unlike anything we have seen in traditional finance. [7] As DeFi smart contract code is made available publicly, this also contributes to the extremely flexible nature of DeFi. Developers can take back-end code (e.g. smart contracts) and create their own front-end (e.g. a webpage) that can enhance user experience of using DeFi protocols. The open-source nature of DeFi enhances innovation and community and is what sets it apart from its traditional counterparts in banking and fintech.

Even though DeFi offers many benefits, it also introduces many inherent non-financial risks. DeFi faces the same financial risks that have been well covered in financial theory. Financial risks DeFi faces include market risk, credit risk, liquidity risk and operational risk. All financial risks have been well defined in the past and it is still common to find a financial risk manager within a company who deals with these types of risks. What separates DeFi from traditional finance however, is the inherent non-financial risks it is exposed to as a direct result of its reliance on the infrastructure layer, *Ethereum*.

With a better understanding of non-financial risks apparent within DeFi, an investor can make more informed decisions. Within traditional finance, an investor can measure risk on expected return (e.g. reward / yield). DeFi is an attractive alternative to traditional finance as the returns found in DeFi are much higher than those found in traditional finance. In traditional finance however, an investor is aware of all potential risks that might come from their investment into something and is prepared to take on a specific amount of risk for that return. In DeFi, risks

are unknown and returns of DeFi protocols are far greater than their traditional counterparts. Investors are prepared to take on unknown risks in order to obtain high rewards – but at what cost and for how long?

This paper seeks to clearly identify key non-financial risks within DeFi on *Ethereum* as a first step to create better standards for proper risk management within DeFi in general.

Chapter 3

Project Objective

The objective of the project is to identify a standardised and complete list of defined non-financial risks that can be used by investors when examining whether or not they should invest in a DeFi protocol, with particular reference to a DeFi protocol that is built on Ethereum. Currently in DeFi, non-financial risk is apparent within the ecosystem, but there is no standardised identification or explanation of risks for a potential investor to access prior to investing. All work done so far in DeFi risk management has been based on opinion, rather than research and evidence. As a result of thorough research, twelve non-financial risks are identified in this paper, so the potential DeFi investor can make better informed decisions regarding investment and risk management strategies. Investors will be able to better hedge their exposure to non-financial risks in DeFi and better risk management tools will be able to be created. In essence, this project which identifies non-financial DeFi risks will help the ecosystem to grow, as risks can be more easily identified and investments can be made with more confidence.

Chapter 4

Methodology

The methodology I chose to complete my project work was based on qualitative research. I collected 30 academic papers related to blockchain risks and DeFi. From these papers, I examined initial ideas about what potential risks already existed in DeFi. I managed to attend four conferences during the period of my project work and I had speakers on panels address my questions about risks in De-Fi directly to refine my original ideas about DeFi risks. I researched various blockchain / cryptocurrency websites, as well as reading articles from influential figures in DeFi about current and future risks. I contacted many journalists, bloggers and innovators in De-Fi and asked for their personal opinions about what they thought the biggest risks in DeFi were now and in the future. I wanted my explanations of risk to be as unbiased as possible and so I interviewed eight experts that work in various areas of DeFi: from insurance, to enterprise, to institutional investors, to a company that secures blockchain networks, to derivatives. I conducted eight semi-structured interviews with experts, having a sample size of 8. I discussed with eight experts about the twelve risks I thought existed in DeFi and ensured they understood and agreed upon the definitions outlined in this paper. All of the eight experts assisted in the refinement of my original DeFi risks definitions. I got involved with Consensys' (largest enterprise Ethereum company) community DeFi Score on telegram, which was the first attempt at creating a community focused on DeFi risks. From researching academic papers, contacting numerous experts working within DeFi, asking questions related to non-financial DeFi risks at conferences, being active in discussions on social media and consistently scanning websites of previous work done

on DeFi risks, I was able to collate, refine and define current risks that can be found in DeFi, in particular non-financial risks that exist on Ethereum blockchain.

Chapter 5

Results - List of Twelve Key Non-Financial Risks in Decentralised Finance on Ethereum

5.1 Scalability Risk

Scalability risk is the risk that *Ethereum* could experience network congestion unpredictably. The risk originates from the core of *Ethereum*, whereby transaction fees are higher the more demand there is to use the blockchain. When there is less demand from less transactions being made on *Ethereum*, the risk is low. When there is high demand from more transactions being made on *Ethereum*, the risk is high. The biggest component of scalability risk is how unpredictable it is to know when *Ethereum* blockchain network will be congested from users sending more transactions than usual. A DeFi application might not work as intended when network congestion is high, in particular if it has a reliance on oracles (more about this in the definition of oracle risk). A recent example of a protocol being exposed to scalability risk was *MakerDAO* during Black Thursday. *Key External Actors* in the *MakerDAO* ecosystem were able to commit arbitrage of the protocol as other economic actors within the protocol could not access their usual operations due to increased network congestion.

MakerDAO is a cryptocurrency that is focused on running as a *'decentralised autonomous* organisation' [41]. *MakerDAO* is an ERC-20 token running on *Ethereum* blockchain. Buying the token gives token holders governance rights within the protocol [41]. MakerDAO runs purely through smart contracts as a type of credit facility / commercial bank within the DeFi ecosystem, issuing loans with interest rates. To understand the issues with MakerDAO protocol, a reader needs to have a basic understanding of Vaults within the protocol and the liquidation process of credit.

As a brief overview, users can deposit *Ethereum* as collateral into *MakerDAO* smart contracts to generate the stablecoin *'DAI'* through a 'collaterised debt position'(CDP) [41]. *DAI* is a decentralized, unbiased, collateral-backed cryptocurrency soft-pegged to the US Dollar [41]. Users generate *DAI* by depositing collateral assets into Maker Vaults within the Maker Protocol. To reduce credit risk, *MakerDAO* requires overcollaterisation to open up a *CDP*, this ensures that the value of *ETH* inside the smart contract is always worth more than the amount of *DAI* it is supposed to be backing. The amount of *DAI* created is relative to the ETH deposited (known as the collateralisation ratio). If *ETH* is worth $100 at the time of CDP creation and the collateralization ratio is 150%, that a user sends 1 ETH ($100) into the *CDP* smart s price of collateral assets in Maker Vaults in order to know when to trigger liquidations. The protocol derives internal collateral prices from a decentralised oracle infrastructure consisting of a broad set of individual nodes called Oracle Feeds [41]. Liquidators (known as *Keepers*in MakerDAO) are automated scripts (bots) or individuals (i.e a person manually doing the various processes), who initiates the liquidation process of a *CDP* [17]. *Keepers* bid on the underlying collateral *(ETH)* in *DAI*, the winning bid is sent the collateral *(ETH)* and *DAI* received from the bidder is burnt, this "closes" the Vault, and makes the system "whole" (all the debt has been paid off, *DAI* has been burnt and the Vault owner has been given the remaining collateral). The durations of auctions are set by *MakerDAO* token holders. These are the relevant fees, credit and protocols within *MakerDAO* that are necessary to analyse events of Black Thursday.

On 'Black Thursday', the price of *ETH* dropped 43% in a few hours. The effect this had on the *MakerDAO* protocol was that of many *CDPs* being liquidated, due to the value of *ETH* dropping below the collaterisation ratio threshold required. As

the price of *ETH* was falling, network congestion was increasing as *ETH* holders began moving their tokens to mitigate the falling value of *ETH* leading to higher gas prices (transaction costs) [73]. In *MakerDAO*, *CDP* owners have the option to add more collateral to their positions should it get too close to the collaterisation ratio. In the case of 'Black Thursday', *CDP* owners could not access their vault fast enough to add more ETH as collateral, resulting in *CDPs* being liquidated in auctions [73]. High gas prices also caused a significant lag in the accurate price of *ETH* collateral from oracles, resulting in many Vaults suddenly simultaneously needing to be liquidated at once when the price oracle eventually updated. As many *CDPs* were being liquidated at once, Keepers within *MakerDAO* could not access the auctions as many of them were using the same script provided by *MakerDAO*, which did not take high gas prices into consideration. There were not enough Keepers accessing the bidding process. Even individual liquidators might have been deterred from purchasing *ETH* in auctions for fear of slippage between the price of *ETH* at the start time of the auction and the end time. Some arbitragers noticed that *ETH* could be purchased by bidding \$0 for any amount of *ETH* being auctioned, as other manual keepers and bots were not paying enough gas for their transactions to be able to participate in liquidation auctions. The arbitragers proceeded to purchase \$8.32million USD of *ETH* by bidding \$0 *DAI* (essentially gathering *ETH* for free). Auction durations were set to last only 10 minutes during Black Thursday. According to the whiterabbit research team, out of 3,994 liquidation transactions, 1,462 (36.6 percent) were realized with a 100 percent discount [73]. The biggest vault lost 35,000 *ETH* [74]. *MakerDAO* did not have proper mechanisms in place to mitigate scalability risk. The fact the auction process was set at 10 minutes is an indication that *MakerDAO* did not consider how high gas prices and increased network usage could potentially cause users of the protocol trouble. *MakerDAO* keepers did not have fair access to bidding due to design flaws misreading scalability issues of the protocol. The CDP liquidation process had critics before Black Thursday, with some vocal about a Black Swan event triggering many *CDPs* to be liquidated [73]. However, this is a normal function of the protocol. If *CDPs* were liquidated on 'Black Thursday' and the system operated as it was intended with fair access to bidding, *MakerDAO* could have been given a pass. However, \$8.32 million

was taken from *MakerDAO* on Black Thursday, simply because the protocol did not properly consider the risks of how more network usage and higher gas prices could affect the functionality of their platform.

5.2 Smart Contract Vulnerability Risk

Smart contract vulnerability risk is the risk that an attacker could find a way to drain funds from a smart contract due to code being written incorrectly, or an attacker uses well-known attack vectors to exploit the functionality of a smart contract. In April 2020 alone within the DeFi ecosystem, there were 5 security incidents related to smart contract vulnerabilities [13]. Since the inception of *Ethereum*, there have been many instances of smart contracts being exploited and funds drained where this should not have been allowed. The most famous exploitation of an *Ethereum* smart contract has been the exploit of *'The DAO'* smart contract, occurring when an attacker utilized a *'re-entrancy'* vulnerability to drain US$60million of *ETH* [39]. Smart contracts are generally designed to manipulate and hold funds denominated in *ETH*. This makes them very tempting attack targets, as a successful attack may allow the attacker to directly steal funds from the contract. [50].

Ethereum 'smart contracts' are pieces of executable code that run on the blockchain to autonomously facilitate, execute, and enforce the pre-defined terms of an agreement without the involvement of a trusted third party [72]. Smart contracts possess the functionality to hold a state, exchange digital assets, take input, store data, obtain information from external services, and express business logic [43]. These programs define a set of rules for the governing of associated funds, typically written in a Turing-complete programming language called *Solidity* [19]. Solidity programming language is very similar to *JavaScript*, however it executes features differently. In fact, some smart contract vulnerabilities are said to come just from the disconnection between the semantics of *Solidity* and the intuition of programmers [1]. *Solidity* syntax resembles a mixture of *C* and *JavaScript*, but it comes with a variety of unique concepts that are specific to smart contract development that might be unfamiliar for new developers, such as visibility modifiers or the function-wide scoping of variables [68]. Developers usually write smart contract code in a high-level

language which compiles into EVM bytecode [68].

In this paper, I will focus on six recurring smart contract vulnerabilities that arise from developers not considering risk and security as part of programming smart contracts, which makes them targets for frequent exploitation. The vulnerabilities outlined are listed in order of incorrect contract vulnerabilities *(re-entrancy, unhandled exceptions, integer overflow)* and validator influenced vulnerabilities *(transaction ordering dependency, timestamp dependency)*. I will also introduce my own subtype of smart contract vulnerability risk not previously considered, *upgradeable key risk*.

5.2.1 Smart Contract Vulnerability Risk – Re-Entrancy Vulnerability

Re-entrancy is exploitation of an incorrect contract vulnerability when a contract tries to send *ETH* before having updated its internal state. If the destination address is another contract, it will be executed and can therefore call the function to request *ETH* again and again. The vulnerability can be explained as follows: "Any interaction from a contract (A) with another contract (B) and any transfer of *ETH* hands over control to that contract (B). This makes it possible for B to call back into A before this interaction is completed." [23]. This exploitation has been used to extract the highest amount of funds from a smart contract in *Ethereum* so far, when an exploiter managed to drain all funds from *'The DAO'*. This type of smart contract vulnerability is still targeted as an exploitation of smart contracts. As recently as April 2020, four years after the first major exploitation of a re-entrancy exploitation within a smart contract, an exploiter managed to drain US $25million from an ERC-777 token (imBTC) on dForce, a DeFi lending application [2]. The re-entrancy vulnerability allowed the hacker to repeatedly increase their ability to borrow all other assets on the dForce's lending platform – ultimately leading to the attacker with the ability to exit with all of the assets deposited in the lending application. The DeFi application (dForce) did not properly assess the risk of a re-entrancy attack occurring on the new ERC-777 token standard when they added it to their application (this could also be associated with design risk). The irony of this attack was that the token dForce added (imBTC) was exploited in another DeFi application

(Uniswap) for $300,000 two days before it was exploited in dForce [56]. Had dForce checked what happened to Uniswap's application, they could have mitigated the smart contract vulnerability risk. Luckily for users of dForce, white hat hackers found out who the exploiter was and therefore the exploiter returned all funds back to dForce so overall all funds were properly returned (as opposed to *The DAO* hack where no funds were returned, which resulted in a hard fork of the protocol into *Ethereum* and *Ethereum Classic*, more on this in finality risk) [2].

5.2.2 Smart Contract Vulnerability Risk – Unhandled Exceptions Vulnerability

Programmers also need to be aware of the incorrect contract vulnerability unhandled exceptions in *Solidity*. In *Ethereum*, a smart contract often needs to call another to fulfil required functionalities, calls are made by either sending instructions or calling a contract's method directly with reference to the contract's name [21]. When a contract is being called, exceptions may be raised raised resulting in a contract terminating and reverting back to its original state whilst simultaneously returning a false value to the user calling the contract [39]. Some examples of exceptions occurring in *Solidity* include when there is not enough gas to execute an operation, the call stack limit has been exceeded or some unexpected system error occurs due to the node of the user performing the call [39]. Some low-level operations in *Solidity* such as send, which is used to send *ETH*, do not throw an exception on failure, but rather report the status by returning a boolean (a true or false output of whether *ETH* has been sent) [39]. It is for this reason, that the *Ethereum* foundation, recommends against using the send function when writing smart contracts, as it is dangerous and causes many problems [23]. Out of 19.366 *Ethereum* smart contracts 27.9% had mishandled exceptions, as of May 5, 2016 [39] A classic example of unhandled exception issues can be seen with the *King of the Ether* contract. A user public address may call a transfer function to become the new owner of a smart contract, yet be unaware that it has had *ETH* returned to their original wallet (for example not having enough gas for a transaction) and a contract that has had *ETH* being sent to it (e.g. KoET) by the same user address might not recognise the transaction has failed, yet proceed to process and update the contract

owner anyway. The risk here is that funds are refunded to a user address and another contract address updates some functionality of it thinking funds have been transferred correctly. This is because no formal exception is thrown resulting in an operation failing from the start. Another example is with the call stack limit of the *Ethereum Virtual Machine (EVM)*. A transfer fails if the call-stack depth is over 1,024 frames, which can be deliberately forced by a malicious caller. When a contract invokes a call or send function to call another contract, the call stack depth increases by one. A malicious caller can invoke a contract 1,023 times and cause a contract's send function to fail purposefully. If the 1024-frames call stack limit is exceeded, EVM will throw an error. Where this could be an issue in DeFi is if a financial instrument is created that relies on many addresses being involved in sharing the value of a specific contract.

5.2.3 Smart Contract Vulnerability Risk – Integer Underflow/Overflow Vulnerability

The incorrect smart contract integer underflow/overflow vulnerability occurs when a computed value is too large for the type attached to the value. The integer underflow/overflow vulnerability is the most common security issue in smart contracts [61]. A study by Min and Cai used Mythril10 to detect vulnerabilities in 1,311 smart contracts and found the most common high severity vulnerability in smart contracts to be "Integer Overflow" (finding the vulnerability in 12.87% of smart contracts examined) [65]. The *Solidity* compiler does not trigger any error flag to resolve the code with integer overflow/underflow problems, so developers must carefully examine the smart contracts they are deploying before release [51]. The *Ethereum Virtual Machine (EVM)* has integer data types that are designated with bit level specification; e.g. "uint8" for an 8-bit unsigned integer, or "uint256," for a 256-bit unsigned integer. The bit level specification of integers causes value storage limitations [51]. Each uint256 is limited to 256 bits in size translating to any integers between 0 and 4 294 967 295 ($2^{256} - 1$), or in the case of uint8 it would be limited to integer size between 0 and 255 ($2^8 - 1$). If an integer variable is assigned to a value larger than this range, it resets to 0; if the variable assigned to a value less than the range, it would be reset to the top value of the range [23]. The best

way of thinking about this vulnerability is using the example of an odometer in a car. A car odometer has 6 slots that can have a number range between 0-9, meaning the largest amount of kilometres an odometer can show is 999999. If the car does one more kilometre than 999999, the odometer will reset to 000000. This example is how underflows/overflows in *Solidity* work if a value is outside the range of the value type attached to it. Generally, it is more likely for an underflow attack to occur, where a number reaches a negative, which in turn changes the integer to the maximum value of the integer data type. The biggest exploitation of this vulnerability was the Proof of Weak Hands smart contract which was drained for US $2 million worth of *ETH* [65]. If a DeFi developer has been careless and accidentally given an opportunity for an integer overflow/underflow in their smart contract, an exploiter could manage to send all tokens inside the smart contract to themselves, putting any DeFi investor at risk who had previously deposited tokens into the smart contract. This could be particularly risky in lending pools. There has been significant progress of this risk being mitigated in future updates of *Solidity* [65].

5.2.4 Smart Contract Vulnerability Risk – Transaction Ordering Dependency Vulnerability

Transaction ordering dependency is a type of validator-influenced smart contract vulnerability. Transaction-ordering dependency occurs when two dependent transactions invoke the same contract and are part of the same block [43]. Transaction ordering dependency is also known as *front-running*. This type of vulnerability is reliant on how miners in *Ethereum* determine contract states between blocks (ie they will add a transaction first to the blockchain if a sender pays a higher amount of gas). In blockchain, two transactions can be sent to the mempool/tx-pool within a block and the order in which they arrive in is irrelevant. The only thing that matters in a block, is how high the fee is that is attached to the transactions in the mempool/tx-pool. If two transactions are identical, the one that will be published is the one that has a higher fee attached. This is a vulnerability for arbitragers. Miners (and other traders) can scan *Ethereum* blockchain for addresses making arbitrage trades and simply copy and paste the same trades with a higher transaction fee attached in the same block to get the reward. The reward deserves to be the arbitrager who

made the initial trade but instead goes to whoever can copy the trade and send it with a higher transaction fee attached. This vulnerability is an area of serious on decentralised exchanges. In traditional finance, a change in the state of a ledger or database is made immediately when a trade is made. On decentralised exchanges, an address might find an opportunity for arbitrage and attempt to arbitrage trades (trading ERC-20 tokens) between decentralised exchanges. The risk of the front-running vulnerability being exposed in DeFi is becoming more prevalent with the introduction of flash loans. Flash loans are a new type of financial innovation in DeFi. A flash loan is a loan that is only valid within one blockchain transaction [53]. A flash loan is essentially non-collateralised, risk free debt [53]. Flash loans mitigate default and illiquidity risk [54]. Flash loans are atomic, either a loan is made with the principal and interest being paid back to the creditor at the end of a block, or it reverts back to its original state if the borrower fails to pay back the principal and interest required by the protocol within the same block [54]. In this sense, flash loans are risk-free because the loan only ever exists on the blockchain if it succeeds and not if it fails. It is impossible for the creditor or the debtor to lose money when the transaction is made through a flash loan. Programmable conditions must be met in order for a flash loan to be successful and therefore published on a blockchain. In flash loans, any amount of an asset can be borrowed, up to the maximum amount of an asset that is provided within a liquidity pool for it, without the borrower needing to put up any collateral. Aave is a protocol that offers flash loans, which has liquidity pools available for arbitragers wanting to use them. Aave sets interest rates on flash loans at around 0.09%. There have been flash loans taken out of Aave in excess of $20 million (with no collateral deposited to take out the loan) [53]. The most common use of a flash loan so far in DeFi has been for arbitrage [53]. Flash loans are becoming easier to use in DeFi as developers continue to build better front-end applications for non-technical people to use them [53]. If flash loan popularity continues to rise, especially from arbitragers, miners could have large incentives to front-run transactions on the blockchain, receiving the reward of intelligent arbitrages instead of the creators of the transaction. At equilibrium, all arbitrages using flash loans should ultimately be extracted by miners [54]. Theoretically, this could serve as a deterrent against the use of the flash loan innovation, as it will leave flash loanees

unable to monetize their arbitrage discoveries. Flash loans are a novel innovation in DeFi that are at risk of never scaling due to the vulnerability of transaction ordering dependency. Flash loans are particularly interesting in DeFi because whilst DeFi is in its nascent stages, there are lots of opportunities to trial its best use case (arbitrage) on different DeFi protocols. Arbitrage in essence exploits vulnerabilities in DeFi, yet is exposed to vulnerabilities itself in the process of assuming it. An example of this can be seen in the recent attacks on *bZx*. *bZx* lost $950,000 in two attacks within a week of each other from an arbitrager using flash loans [54]. Flash loans can be dangerous when combined with buggy code, improper price feeds or both [27]. An anonymous arbitrager quickly found out they could use a flash loan on dYdX, putting up no collateral, at no risk to them, to borrow a large sum of money to make trades on different DeFi applications to exploit different price feeds and liquidity pools. Basically, an arbitrager knew that they could exploit illiquidity of exchanges by using trading strategies to inflate prices of assets, then shorted and longed the same asset on separate exchanges to earn a profit. Flash loans amplify DeFi's underlying *composability risk, oracle risk, design risk and smart contract vulnerability risk.* A flash loanee is exposed to being front-runned by miners due to transaction ordering dependency vulnerability whilst simultaneously generating other vulnerabilities on other DeFi platforms putting other DeFi user funds at risk.

5.2.5 Smart Contract Vulnerability Risk – Time Stamp Dependence Vulnerability

The other validator-influenced vulnerability is time stamp dependence, which occurs in contracts using the block timestamp as a condition to trigger and execute a transaction [43]. Time is dependent on asynchronous block times in *Ethereum*, not on a synchronous global clock. If a contract uses the block.timestamp (or now) global variable as a triggering condition for executing a critical operation (e.g. a money transfer) or as a source of randomness, it can be manipulated by a malicious miner [8]. This can be done because the timestamp is set to the system time of the validator's local computer or server [39]. When a block is mined, the miner has to generate the timestamp for the block. The timestamp of a block can vary by approximately 900 seconds comparing with other blocks' timestamps in order for it to be

published on a blockchain [39]. Because of this flexibility in setting the timestamp of a block by miners, it is possible for an adversary or malicious validator to choose different block timestamps to manipulate the outcome of timestamp dependent smart contracts. If a miner holds a stake on a contract, he/she could gain an advantage by choosing a suitable timestamp for a block he/she is mining [39]. A validator can set the block timestamp to be a specific value which influences the value of the timestamp-dependent condition (such as randomness) to favour the miner [39]. New types of instruments (especially hedging ones) being created in DeFi that generate randomness using block.timestamp or financial contracts that have a payout based on blockchain events within a 15-minute period should be tested vigorously in the case a validator could manipulate a block timestamp to increase their probability of a smart contract payout.

5.2.6 Smart Contract Vulnerability Risk – Upgradeable Smart Contract Vulnerability

Upgradeable smart contract risk is a subtype of smart contract vulnerability that is a new type of risk introduced in this paper. Upgradeable key risk is the risk that an administrator can upgrade a smart contract and completely change the behaviour of it that a user expects. This is also a type of design risk. Currently, the majority of popular DeFi protocols have some form of centralized control that enables specific 'administrator' addresses to intervene in powerful ways (e.g. pausing the system, modifying balances, blacklisting addresses or upgrading parts of the system). The greater access administrators have to upgrading smart contracts and controlling protocol procedures should equal greater amounts administrator risk (where a centralised controller can change a part of a protocol to negatively affect DeFi user funds e.g. by blacklisting their address, stealing their tokens etc). No DeFi project can prove the operational security of their upgradeable smart contracts is strong. There are two things that a user should be wary of if an administrator can upgrade smart contracts. A DeFi user should make sure DeFi protocols publicly disclose when contracts are being upgraded (mitigating disclosure risk) and that they have been audited thoroughly, otherwise the upgraded contract could be exposed to new smart contract vulnerabilities. As well as DeFi users making sure they are

always informed of smart contract changes and they should determine what level of trust they have in the administrators that have access to upgrade smart contracts of a DeFi protocol. Right now, existing techniques to upgrade smart contracts have flaws that increase the complexity of the smart contract significantly, and ultimately introduce more smart contract vulnerabilities. [47]. Trail of Bits, one of the biggest auditors of *Solidity* smart contracts recommends developers strive for initial deployment of simple, immutable, and secure smart contracts, rather than opting for postponing importing code in the future to address security or feature issues [47]. In DeFi, security is paramount and malleability of smart contracts adds complexity and potential attack surfaces, making them more vulnerable than non-upgradeable non-complex smart contracts [15]. Upgradeable smart contract risk is perhaps one of the biggest risks in DeFi as it juxtaposes against what DeFi strives to become, financial products with no intermediary. Right now in DeFi most applications have an administrator that can upgrade smart contracts, which technically can be considered as the controller being an intermediary as they have real control over user fund balances. Upgradeable smart contracts lead DeFi users to be exposed to greater amounts of *smart contract vulnerability risk, design risk, composability risk, administrator risk and disclosure risk.* DeFi watch lists how much access administrators have to chance a DeFi protocol, which DeFi users should use to be aware of their exposure to the above risks [71].

5.3 Oracle Risk

Oracle risk is the risk a smart contract could receive dishonest input about offchain values due to manipulation of information from the provider, or an oracle does not update a smart contract with offchain information as fast as an application expects it to (related to scalability risk). DeFi protocols are extremely reliant on oracles, which are third parties reporting information from real-world (off-chain) sources. Oracles are necessary because distributed ledgers are deterministic, single data systems, inherently incapable of recording or considering any information other than transactions within the blockchain: a concept referred to as the 'oracle problem' [57]. Oracles act as a middle layer between the blockchain and an API by translating in-

formation for the blockchain to read. It is important to note that a blockchain oracle is not the data source itself, but rather the layer that queries, verifies, and authenticates external data sources and then relays that information [57]. An oracle taxonomy can be categorised into data source, trust model, design pattern and interaction [**?**]. This paper focuses on the trust model of oracles. Many DeFi protocols are dependent on oracles because many of their smart contracts are constructed in a way that relies heavily on offchain financial information (e.g. prices of specific real-world assets or currencies). An example of this is *MakerDAO* relying on price oracles to maintain the soft-peg of their stablecoin *DAI* to USD [41]. The goal of the pricing oracle is to approximate the market value of the underlying collateral assets as accurately as possible in real-time [30]. Oracles can be a cause for concern as their trust models can differ from the trust model of the infrastructure they are feeding information into (eg the design of *Ethereum* is trustless and decentralized, whereas some oracle designs are trust-dependent and centralised). Since an oracle controls the input data into a smart contract, it controls the operation of the smart contract as it responds to the input data [59]. Some critics of DeFi question how decentralised DeFi protocols really are, as most incorporate centralised oracles as input into their decentralised smart contracts [59]. An important design consideration for DeFi protocols is whether they choose to use a trust model that is centralised or decentralised from an oracle provider (bearing in mind each use of an oracle to update a smart contracts costs money denoted in *ETH* by the network). In a centralized trust model oracle service (such as Provable), a third party private company fetches and feeds data into the smart contract. Should a DeFi protocol rely on a centralised oracle provider, they need to have full trust in that provider relaying correct information and consider the manipulation risk that the provider could be compromised (from leaking sensitive information, getting hacked, experiencing downed servers, feeding incorrect information, etc) [6]. Not only is there a risk that one oracle provider is easier to manipulate, but it can also be said that centralised oracles lose some key features of smart contracts being *deterministic, tamper-proof,* and *reliable* in their end-to-end execution [6]. A blockchain can have trustworthy transactions but if a centralised oracle provider is dishonest, the information from oracle inputs within these transactions will be incorrect. Even if a blockchain reaches

consensus on how a transaction (e.g. a smart contract) is passed between addresses by an agreeance from nodes, the nodes that accept the transaction as valid have no way of identifying if a transaction includes valid metadata (e.g. information about the outside world like the temperature). Many DeFi protocols rely on oracles which determine pay-outs. An oracle could have reason to manipulate data to receive an unfair payout by feeding inaccurate data into the blockchain (e.g. through smart contracts) if logic within a smart contract relies on data being a certain value.

It is currently common practice for DeFi protocols to select a set of centralised trust model oracles for price reporting feeds, which may be crypto exchanges, over-the-counter market makers, or traders, and task them with providing price update of the corresponding collateral asset at regular intervals [30]. However, a poorly designed price-reporting system can be exploited by malicious attackers (that can be made easier with centralised trust model oracle providers One example of such exploitation was the oracle attack on June 25 2019 by an arbitager on the DeFi protocol Synthetix (a synthetic asset issuance platform) [66]. At the time, Synthetix only had two commercial APIs serving as price feeds for its forex product sKRW (synthetic Korean Won), when one of the APIs started to intermittently report a corrupted price that was inflated by 1000x [62]. The one correct and one inflated price feed were aggregated into the sKRW smart contract, when a trading bot noticed the arbitrage opportunity and proceeded to net $37 million from the incorrect pricing from a corrupted oracle [30]. An interesting thing to note about this attack is that Synthetix halted all transfers and trading within the system after the attack [63]. A key principle of DeFi is that it runs without an intermediary, yet in this situation an intermediary had the capacity to shut down the system, meaning DeFi users of Synthetix are trusting people, not smart contracts with their funds (see *design risk, admin key risk* and *upgradeable contract vulnerability risk).* Synthetix has a high reliance on price oracles as synthetic assets only follows the prices of assets. Investors have no right to the underlying asset when they purchase a synthetic [25]. Because of the centralized point of failure of the centralised price oracle attacks, Synthetix decided a better option for oracles would be to use a decentralised service known as *Chainlink*, in order to decentralize both the smart contract execution and the data oracle layer [24]. A decentralised trust model for oracles is an appealing alternative

to centralised oracles as decentralised oracle providers have economic incentives to provide reliable input data to smart contracts, which eliminates the single point of failure risk of centralised oracles. Decentralizing the oracle trust model gives developers the flexibility of choosing number of oracle (nodes) they want to service their smart contracts or the minimum amount of reputation a node can have to provide information to an application. Having multiple oracles not only protects against a single oracle going offline; it protects against an oracle being a single point of attack for hacks or bribes which is a key risk in centralised trust model oracles [6]. *Chainlink* decentralises not only nodes that are providing information but also the data sources the nodes are accessing themselves, as nodes gather data from multiple sources (instead of just one like in centralised) and then aggregate that data into a single deterministic data point to trigger a smart contract [6]. Decentralised trust models can be created in customised ways to make sure the data being input into smart contracts has the maximum chance of being correct. For example, *Chainlink* uses node reputation. Node reputation is based on things such as previous histories of a nodes up-time and correctness of information a node has provided in the past. In addition, *Chainlink* is an ERC-20 token that can be traded which has an economic incentive model determining the price of the token. The token ensures nodes are held accountable as they have to lock up this token (which has monetary value) to have the right to relay offchain information to a smart contract. If they acquire and distribute dishonest information to the network (which can be distinguished by *Chainlink* network by comparing their values against other nodes), the node loses the ERC-20 tokens and therefore loses monetary value. Whilst decentralised oracles might provide a greater chance of trustworthy data with added features such as reputations, decentralised data sources and economic incentives, it also comes with added risks of decentralisation itself such as the chance of *sybil attacks, mirroring or freeloading* [64]. All of which are opportunities for oracle (nodes) to collude to manipulate information, replicate other node information without checking authenticity, or share information offchain to other nodes they control to reduce operating costs (from refreshing APIs).

Liu and Szalachowski found that although oracles play a critical role in the DeFi ecosystem, the underlying mechanics of oracles are vague and unexplored [38].

This study concluded that currently there is large potential for the development of malicious oracles due to a lack of transparency of oracle information, lack of accountability of oracle providers and inadequate operational robustness [38]. At present, it is unclear where centralised and decentralised oracle protocols retrieve date/information for use by their oracles. Oracles can and have caused huge problems in DeFi, yet when an oracle does not work as intended, there is no accountability from the provider that relayed the improper information. In practice, a DeFi application should be more accountable to users by fully informing users how oracles are used and where information provided by the oracle is retrieved from. In future we could see solutions such as utility tokens, derivatives or improper information insurance that forces DeFi oracle providers to become more accountable for the information they are relaying to DeFi protocols. Finally, DeFi applications must ensure they always send enough gas for operational robustness by paying a high enough gas price and sending adequate amounts of maximum gas to mitigate scalability risk and oracle risk simultaneously in the case the network becomes congested, as higher gas prices will allow for offchain information updates as expected. A recommendation for DeFi applications would be to publicly display all oracle providers they use and disclose how what data sources these oracle providers are using for information, as well as how much gas they spend per oracle call and maximum amount of gas they are willing to pay for an oracle call, to make sure during network congestion the platform will work as intended (also mitigating design risk). The oracle problem is a big issue in DeFi and both centralised and decentralised trust models have their strengths and weaknesses for reliability of information. At the end of the day, blockchain oracles are always at risk of centralization, collusion, and sybil attacks [3]. A trend we could see in the future to mitigate this is to use numerous decentralised oracle providers to add another level of trust against manipulation. DeFi users should have great knowledge on what determines the prices of the financial products they are trading and whether it relies on the use of oracles, as oracles can be manipulated and result in DeFi users losing funds. Oracles, like DeFi, are an innovative field. It is expected that as oracle solutions continue to mature, performance of them will improve through lower-cost scalability, better privacy, enhanced reliability and greater connectivity. For now, oracle risk should

be considered one of the biggest risks in DeFi that users need to be aware of.

5.4 Design Risk

Design risk is the risk that one small flaw will lead to the demise of a protocol not working as intended. A DeFi protocol can have great amounts of security and high levels of risk mitigation yet if it adds a new smart contract or token to its protocol that has less or different levels of security or risk mitigation without thorough examination of what it is adding, it can potentially lead to the undoing of the whole protocol. This is particularly relevant in DeFi, as composability of DeFi is a strong selling point, meaning that often DeFi platforms integrate code made by others with their own code. The risk of this is that third parties might start using underlying code made by another DeFi platforms that do not work well within their own platform as the platform is not designed properly for integrations or new standards. An example of this integration problem occurred in May 2020, when a *Tokenlon* used the ERC-777 token standard to create a token, rather than using the standard ERC-20 token which is designed to suit most De-Fi platforms. [67] ERC-777 has different programing standards than ERC-20 and therefore security of the token standard is completely different and hence there is different vulnerabilities of the token. Uniswap and dForce integrated the imBTC token made by Tokenlon (an ERC-777 token that is backed 1:1 with BTC) whilst not properly managing risk of how the new token standard could be exploited within their own smart contracts. Uniswap was attacked first, even though twelve months prior Open Zeppelin had already publicly disclosed an easy way to attack Uniswap liquidity pools using a re-entrancy attack if the token confined to ERC-777 standard [76]. Uniswap does not choose for itself what tokens can be traded on its platform [70]. A user unknowingly supplied an ERC-777 token to be traded on Uniswap without knowing that it could be exploited. An attacker took advantage of a re-entrancy vulnerability by calling the tokensToSend method of the imBTC token within the Uniswap smart contract. The tokensToSend method (attacking contract) was called after receiving *ETH* but before the imBTC tokens were swapped. Because of its re-entrancy vulnerability, the exchange was trying to swap imBTC tokens by sending over *ETH* thinking it

would receive imBTC tokens but never did as the function never properly instigated. Every *ETH* that was lost from the imBTC liquidity pool inflated the true price of imBTC until the exchange lost all *ETH* and liquidity of the pool was drained to the attacker without the attacker losing any of his imBTC tokens [76]. A few days later the attack happened again on dForce, this time draining $25 million from their liquidity pools (this is the same attack mentioned in re-entrancy of smart contract vulnerability risk). A key consideration here is the composability that led to the design in dForce's protocol. dForce essentially copy and pasted Uniswap's smart contracts (because smart contracts are open source for anyone to see on the blockchain) with a different user interface. When an attacker drained liquidity on Uniswap, dForce should have known that their user funds were at risk as they used the exact same code as Uniswap. Due to oversight, the attacker was able to perform the exact same attack on dForce days later. Design risk could apply to any DeFi platform as designs of token standards and best practices are constantly evolving, therefore if DeFi protocols decide to integrate third party code within their platform, they should be aware of how the integration could affect how their existing smart contracts operate. After all, a DeFi platform is only as strong as the weakest security platform that they list. A final point about design risk is that it can also relate to complexity of the DeFi platform. As mentioned previously, in *MakerDAO* there were not enough keepers to properly liquidate CDPs. To be a keeper in *MakerDAO* takes a lot of time and effort to understand and most keepers are run by bots (therefore someone needs a high-degree of programming knowledge). If it were simpler to be a keeper at the time of CDPs being liquidated, there may have been more keepers on the system at the time to mitigate the $8.2 million they ended up losing.

5.5 Composability Risk

Composability risk is the risk that a DeFi platform is reliant on another DeFi platform operating properly for its own platform to function correctly. Composability risk is related to design risk. Composability is a system design principle that enables applications to be created from component parts [77]. Composability is often referred to as money legos in the DeFi ecosystem as it code can be selected and

assembled in multiple combinations [46]. DeFi developers can easily use and build on top of existing protocols because most DeFi protocols are open-source for anyone to use. The adoption of DeFi can be attributed to its composability as composability helps to create 'network effects,' a powerful phenomenon where the value of goods or services grows as the number of users increases [42]. Due to its open-source nature, a relatively large number of DeFi protocols integrate components made by third parties in the making of their own protocol [77]. DeFi protocols benefit from composability as it contributes to faster innovation in the space (ie network effect and open-source code), but it results in higher levels of interdependency between platforms (causing great amounts of composability risk). Kyle Kistner of dYdX (a popular DeFi derivative platform) stated that risk flows one way with composability, as a protocol is more at risk the more existing protocols it builds on top of [35]. Stan Kulechov (creator of Aave, the platform that created flash loans) recommended that to improve risk assessment of DeFi, to take into account the composability within any scoring system, as a protocol might include inheritable risk from other protocols it is building on top of [36]. One might suggest that protocols that are more exposed to composability risk are ones that build products on top of other existing smart contracts, rather than their own unique products. The interdependency of all DeFi platforms makes it highly exposed to systemic risk as the fall of one could lead to the fall of many. Herz and Gervais labelled this financial contagion within DeFi [31]. Herz and Gervais stated that DeFi protocols do not exist in isolation but rather intertwine with one another creating a system of assets and debt obligations that are hard to follow [31]. Therefore, the failure in one core protocol that many others have built on top of would cause a cascading shock throughout all other protocols that use any of its components, spreading like a contagion to the rest of DeFi. This example could be foreseeable in DeFi. If DeFi continues to grow and one core protocol such as *MakerDAO* that creates the stablecoin *DAI* fails (e.g. *DAI* loses its peg to the USD and is no longer \$1), this could cause the failure of another DeFi platforms that uses *DAI* in their protocol (e.g. *Compounds* issuance of *cDAI* from a collaterisation of *DAI*) resulting in a DeFi user losing funds. This can be known as evaporating collateral. In this scenario, holders of *cDAI* would have to purchase more DAI to make sure their positions remain adequately collateralised,

which would put stress on the liquidity of *DAI* and also leave the door open for scalability risk if the network becomes congested from many users doing so.

The result of this would be limited liquidity of *DAI*, leaving many positions on *Compound* undercollaterised and therefore liquidated, due to a fault in *MakerDAO*, not *Compound*. This is an example of how composability risk multiplies the more protocols use existing platforms to build their products. There are currently 600 DeFi platforms that have integrated *DAI*, [42]. meaning 600 DeFi platforms are exposed to the same composability risk. The financial innovation of DeFi from composability, makes the environment somewhat similar to that of the bank environment around the time of the GFC twelve years ago that was experiencing increased amounts of financial innovation from financial engineering. Similarities can be made in DeFi of types of financial innovation going on in 2008, such as pooling risk in different products (from platforms building on top of each other e.g. *Compound* using *MakerDAOs* code), re-hypothecation of collateral (using collateral of one protocol in another protocol whilst being exposed to credit risk) and fractional ownership (limitless opportunities to creaste wrapped tokens accruing interest that represents claims on another ERC-20 tokens e.g. CHAI made from *DAI*).). Essentially, the global financial system broke down as a relatively risk-free product (mortgage) was repackaged into many different securities (e.g mortgage-backed securities and collaterised-debt obligations) with different risk levels. Many mortgages ended up defaulting and therefore all securities that relied on cash flows from the underlying mortgage lost value. Banks were intertwined and got caught holding risky securities they did not foresee [33].Banks were misled by ratings agencies rating debt higher than what it actually deserved, whilst simultaneously becoming complacent with checking that the payoffs of securities they were buying would continue into the foreseeable future (i.e. checking for credit risk) [33]. Much of this complacency was a result of the securities deriving payoffs that were so complex, not even financial engineers understood them [33]. Without understanding risk of the underlying securities and complexities of payoffs, banks took excessive risks anyway by creating and trading the mortgage-backed securities to obtain high returns and premiums for each security that could be created and sold. In DeFi right now, products are becoming more complex and often tokens are created that have payoffs from other tokens,

not only this but products are intertwined with one another as DeFi platforms are regularly built using existing components and therefore reliant on the whole system working in order for their platform to work. DeFi is at risk of the very same financial crisis it's underlying technology was created to prevent (from excessive financial risk which led to the GFC). DeFi platforms do not mitigate composability risk at all currently and if anything they enhance it. DeFi users should know that if a protocol they are using uses existing components and products that are not original to the protocol, they are exposed to higher amounts of composability risk. Composability risk could lead users to undercollaterisation, liquidation and complete loss of funds.

5.6 Centrality Risk

Centrality risk is the risk a central point of failure in DeFi could be the undoing of the whole DeFi ecosystem. One type of Centrality risk in DeFi is related to upgradeable smart contracts, as this type of risk has an administrator controlling a platforms functions. If someone can control a platforms functionality, DeFi users put trust in the administrator to not change how they think the platform will operate when they invest. Most DeFi protocols have varying levels of centralisation, as there is usually functionality for a product owner to update or change the protocol at any time. Many DeFi protocols plan to decentralise as their platforms mature but whether or not this can be done successfully remains to be seen. This centrality risk to DeFi users is a double-edged sword. On the one hand if a protocol decentralises further in the future it might be more prone to smart contract vulnerabilities that can't be changed, whereas if a DeFi protocol always has administrators, it can always be stopped or taken down at any time. If a DeFi protocol endures the latter, a user always puts a lot of trust in administrators and currently there is almost no accountability on DeFi protocols to behave properly. Recently Spankchain and notorious security penetrator Samzscun criticised DeFi protocol *Compound*, for not properly disclosing that they had a centralised administrator. *Compound* is a smart contract that allows users to borrow and lend tokens. *Compound* is a protocol on the *Ethereum* blockchain that establishes money markets, which are pools of assets with algorithmically derived interest rates, based on the supply and demand for the

asset [69]. Spankchain's founder Soleimani, publicly announced he was looking to invest $500,000 *DAI* ($500,000 USD) and would have liked to invest in *Compound* to earn an attractive rate on interest, however he noted there was 'no free lunch' and concluded that his investment would be too risky in *Compound* with a centralised controller of platform functionality [44]. This was an opportunity cost to *Compound* for them not being fully decentralised and an example of how DeFi users have less faith in DeFi protocols the more centralised they are [44].

Another type of *centrality risk* in DeFi is related to centralised stablecoins. There are 3 main stablecoins that are used the most in DeFi: *DAI, USDC and USDT (Tether)*, all of which have some central points of failure (hence centrality risk). All stablecoins mentioned are supposed to emulate the US dollar and be pegged 1:1 with it. We have already learnt in this paper how *DAI* is minted by a decentralised minter *(MakerDAO)* with *ETH* as collateral. *USDC* and *USDT* issue stablecoins differently because they are centralised, meaning a third party mints the tokens that are supposedly backed 1:1 with USD. An investor using centralised stablecoins *(e.g. USDT and USDC)* has to trust the value of their token is truly worth 1 USD. In reality, a centralised third party can control who owns their token or the token itself might not have true value if there no accountability for the issuer of the token to properly back it. This is a centrality risk as a DeFi investor has to trust a centralised issuer will act honestly. Currently, *USDT* is the biggest stablecoin by market capitalisation. At the time or writing, there has been $9 billion *USDT* issued (denoted in USD) that is now held in cryptocurrency wallets [10]. 1 *USDT* is supposed to be redeemable for 1 USD, meaning if an investor sends 1 *USDT* to the third party who issued it, it should be fully redeemable for the sender to receive 1 USD back. In DeFi, stablecoins are frequently used in lending protocols, where investors can earn returns for depositing the token into a pool that other investors can borrow (all done without an intermediary). A problem here can arise if DeFi investors are borrowing and lending *USDT*, expecting that 1 *USDT* token is actually backed by 1 USD and can be redeemed for as much. In April 2019, *USDT's* attorney revealed that *USDT* tokens were only backed 74% by reserves [20]. In other words, the digital tokens, which are meant to be worth $1 apiece, would have only $0.74 of redeemable value if all were converted at once. In DeFi a complete failure of

the ecosystem due to its composable nature is not completely out of the question, therefore in future investors should be wary when using *USDT* in case they are left with an amount of the token they cannot properly redeem for the true amount of USD they think they owned. Despite this, as much as $US 200 million *USDT* is being borrowed and lent on *Compound*, a popular DeFi lending protocol [26]. *USDC* is the second most popular stablecoin in DeFi which is also completely centralised, as users have to trust Circle (the third party that issues it) will redeem 1:1 their *USDC* for USD. Whilst there is more reliability that *USDC* is backed 1:1 with USD as compared to *USDT* (being backed only 74%), the centrality risk here lies in the fact that circle can blacklist certain addresses and take *USDC* away from addresses holding it if they suspect suspicious activity [9]. This is highly risky to a DeFi user, as if any time Circle deems an address that holds *USDC* as suspicious, they have a right to withdraw all *USDC* from a DeFi users wallet having a DeFi user lose all *USDC* funds. Finally, there is now some centrality risk in the most decentralised stablecoin, *DAI*, as *MakerDAO* (the decentralised party that mints *DAI*) recently proposed to accept *USDC* stablecoin as collateral to mint *DAI* to address price instability (losing peg) and liquidity issues. However, there were many critics of this as many noted that using *USDC* as collateral for a decentralised stablecoin introduces centrality risk [60]. An example of this is 100,000 *DAI* being minted using 150,000 *USDC* as collateral, if the third party that issues *USDC* withdraws 150,000 from the address that minted *DAI* (as they are suspicious of an address), the *DAI* would be backed by nothing (collateral is worthless) and lose its peg to USD (being risky to DeFi traders as they always expect their stablecoin to equal 1 USD). Therefore, the most popular stablecoins used in DeFi *(USDT, USDC and DAI)* all have different levels of centrality and each have unique risks for holding them which could lead to the value of the stablecoin losing its peg to the asset it is targeting (USD). Even if DeFi users are using a decentralised platform, the value of the tokens they hold are dependent on the third parties staying true to redemption rates and correct exchange rate of tokens for fiat currency, meaning trust is not solely put into smart contracts, but also centralised third parties.

The final type of *centrality risk* this paper will discuss is the reliance on *Infura* as a node infrastructure operator. The *Infura* IaaS (infrastructure-as-a-service) by

ConsenSys, provides *Ethereum* clients running in the cloud, so users do not have to run a node themselves to work with *Ethereum* (which is expensive) [32]. *Infura* provides enormous value to the development community by removing the cost and time investment that is necessary to sync and run an *Ethereum* node that would otherwise put DeFi development out of reach for many. An estimated 63% of the *Ethereum* community use *Infura* as their preferred method of interacting with the blockchain [22]. *Infura* provides the necessary tools for any application to start developing on *Ethereum* immediately, without the need to run the infrastructure themselves. Not only does it help developers but the node cluster lets users run applications without requiring them to set up their own *Ethereum* node or wallet (a good use-case for non-blockchain educated users). The *Infura* API suite provides instant access over HTTPS and WebSockets to the *Ethereum* and IPFS networks [22]. Another reason for *Infuras* popularity is because of its ability to help applications scale (reducing scalability risk at the cost of greater centrality risk). Even if a DeFi protocol decides to start developing their application running their own *Ethereum* node, if it gets adoption they will need to accommodate more traffic as users make more requests. To scale on *Ethereum*, a DeFi protocol needs to run more nodes to handle transactions to make sure user experience does not break. *Infura* is a convenient solution to scaling a DeFi protocol as a DeFi platform can connect to the free *Infura* API as a substitute to running more nodes and scale, rather than spend time and money setting up additional *Ethereum* nodes of their own. *Infura* ensures network stability and uptime in the operations of their nodes, yet this is actually a potential centralised point of failure for any decentralised application using *Infura*. A public blockchain strives to be decentralised. In an ideal blockchain ecosystem, service providers, dApps, and decentralized systems would operate their own nodes to verify information and data in a fully peer-to-peer and distributed manner. If node infrastructure operators like *Infura* are tasked by popular DeFi protocols to handle data requests on their behalf (as they want to save money from not running a full node), then the risk of centralizing the *Ethereum* network could increase [75]. *Infura* is operated by a single provider – the *Ethereum* development studio *ConsenSys* – and relies on cloud servers hosted by Amazon [48]. As such, concerns exist that the service represents a single point of failure for the entire network. If a DeFi

protocol relies on *Infura* to communicate with the blockchain then it creates a single point of failure as *Infura's* service could introduce bugs or become unavailable for whatever reason, crippling the ability for DeFi protocols relying on *Infura* to function properly [22]. In addition, any DeFi protocol using *Infura* removes the core benefits of a decentralised application (e.g. being unstoppable, censorship resistant and trustless). *Infura* has full control of data it is providing and as such DeFi protocols need to trust this data is correct with no way of verifying, trust *Infura* is not censoring transactions and also trust that *Infura* will always run a node for them (which is also reliant on Amazon servicing *Infura*). Metamask the most popular wallet on *ETH* makes use of *Infura*. MetaMask makes use of *Infura* for communicating with the *Ethereum* blockchain to determine a user's account balances and to submit transactions [14]. If *Infura's* service was somehow compromised by an attacker, the attacker could send false information to an honest user's cryptocurrency wallet that could cause them to think they have received a payment when in reality they have received nothing. It's easy to see how this could lead to real-life consequences including the loss of funds [22]. One such example is how *Uniswap*, a DeFi *decentralised exchange (DEX)* uses *Infura*. *Uniswap* uses *Infura* as the go-to provider for connecting to *MetaMask* and for querying information when a user has not connected a wallet to the *Uniswap* exchange. *Uniswap* pulls an array of information from smart contracts on the *Ethereum* blockchain to feed into the interface and populates data such as pricing between pairs, user balances, and swap rates through *Infura's* API. *Uniswap* relies on *Infura* providing information to a user when they are not running their own node, *Infura* or an attacker could manipulate the data that is supposedly on the blockchain and send wrong information to a user's DeFi application (e.g. manipulate prices). Because of DeFi's composable nature, any type of attack, censorship or modification of Infura on a DeFi protocol could lead to financial contagion of the DeFi ecosystem. Not only this but many DeFi protocols rely on *Infura* to feed them accurate information about the *Ethereum* blockchain, so an attacker might find incentive to attack *Infura* and then disperse corrupt information to DeFi protocols for profit in the future. If *Infura* was to be attacked, this could be the demise of the DeFi ecosystem.

5.7 Economic Incentive Risk

Economic incentive risk is the risk economic incentives that encourage network participants to perform certain actions could fail to encourage the right behaviour or not be sufficient enough, leading to other users being adversely impacted [34]. Defined by Hugh Karp from Nexus Mutual (DeFi's leading insurance protocol), economic incentives risks are specific to particular protocols. Hugh Karp explains the risk as *'DeFi platforms having a reliance on economic actors acting rationally to perform certain actions within a protocol. This reliance can break down when circumstances change and an action is no longer rational or when there is an opportunity for changing the pay-off diagram e.g. with bribe attacks which would change pay-off structure and break incentives (hence increase economic incentive risk).'* For example, the incentives in the *MakerDAO* smart contracts could be too aggressive and the *DAI/USD* peg could break if the *ETH* price drops too far, too quickly. The top 4 DeFi protocols (accounting for about $900 million USD in *ETH* locked up) are *MakerDAO, Synthetix, Compound* and *Aave* – all of which have their own token that is used for governance of the DeFi protocol. Token holders of these DeFi protocols determine how the platform functions. *Compound* is the second biggest DeFi protocol in existence right now (having *ETH* totalling $145.4 million USD locked up in its platform) and is the most recent out of the top four DeFi protocols to introduce a native token [52]. Whilst *MakerDAO, Synthetix* and *Aave* have always had a native token operating within their platform (as they bootstrapped funding selling their native token), *Compound* first started without a native token (bootstrapping funding from venture capitalists), as a lending platform formed by a collation of smart contracts and a front-end. *Compound* recently introduced a governance token on June 15, 2020 – COMP – which allows tokenholders and delegates to vote on important protocol decisions like new collateral types, borrowing power, and interest rate models. Currently, COMP holds no economic benefits and is solely used to vote on protocol proposals. Because COMP is purely a governance token, COMP holders might vote for economic benefits of the tokens e.g. in the form of platform fees in the future. The risk here is that COMP holders could vote for anything in the future, with their votes being dependent on the amount of COMP token they hold – like in traditional votes with stocks. The founder of *Compound* recently wrote *"The*

governance right gives the community complete control to evolve the economics of the protocol and COMP in entirely new ways – so I have no idea what COMP looks like in two years." [18]. This is potentially worrying, as although a native token mitigates centrality risk, the token holders are not accountable to regulation like in traditional financial systems and therefore can start proposing malicious upgrades for their benefit, rather than the benefit of users that have already interacted with the platform. If there are millions of dollars deposited in *ETH* on the platform and a small number of addresses accumulate large amounts of the native token COMP, these users would have a large say in how the platform functions including if they charge lenders/borrowers high amounts for using their platform / the type of interest they can earn on assets they interact with. For example, the protocol could propose a new upgrade which gives them a type of liquidation function on funds that have been locked up for a certain amount of time, which could introduce risks a DeFi user was not aware of at the time first interacting with a protocol. Any number of risks could be thought of if a native token creates proposals that contribute to network participants behaving maliciously for financial gain.

5.8 Financial Illiteracy Risk

Financial illiteracy risk is the risk that a platform has been developed by someone with no financial background. DeFi is the transformation of traditional financial products into products that operate without an intermediary via smart contracts on *Ethereum* blockchain. The programmers transforming traditional financial products into code in smart contracts often have no financial background whatsoever. This is in contrast to traditional finance, where traditional financial products are traded by institutions and created by financial engineers with certification. DeFi strives to be more accessible to the global community and is open for anyone to build products. A serious risk of this is that developer might create a DeFi product having no financial knowledge about financial implications of the product they are creating and have users investing in the product without considering the risks. Currently the DeFi environment has no certification to prove someone knows exactly what they are programming or investing into. A lot of risks mentioned in this paper might

be mitigated if a type of certification or course was required for participation in DeFi. Of course, this offsets the goal of DeFi to be accessible to anyone no matter their education. However, as we have already seen, sometimes DeFi platforms do not comply with all decentralised ideals (e.g. great amounts of centralisation from administrators able to control platform functionality), so perhaps in the future we will also see some barriers to enter DeFi, defying the ideal of DeFi accessibility, for the future adoption of DeFi.

5.9 Regulatory Risk

Regulatory risk is the risk that any DeFi protocol can be affected by government with either laws being made that affect how a DeFi protocol operates or laws being made effectively shutting down DeFi protocols. In 2017, during the *'Initial Coin Offering'* phase, many projects that had no value at all were created due to the ease with which ERC-20 tokens could be created on *Ethereum*. After the peak of the bubble at the end of 2017, regulators caught up with ICOs and made it almost impossible by outlawing companies being able to raise funds via ICO from 2018 onwards. In general, the greatest amount of regulatory attention so far has focused on traditional concerns of investor and customer protection, particularly in the case of cryptocurrencies and ICOs, not on DeFi [77]. In many ways, DeFi has some striking similarities with that of the ICO boom, as the growth of the ecosystem is exponential like ICOs were in 2017 and an increasing amount of projects and DeFi related tokens are created every day. DeFi is so innovative that it currently operates in a regulatory grey area, as no regulator in any jurisdiction has attempted to regulate how DeFi is used in any way. In the future, determining jurisdiction will be far from easy, as DeFi projects tend to fall under many different state, federal/national and regional licensing and supervision regimes. Each potential regulator will impose additional conditions reflecting its own perspective, mandate and powers, as we have seen with government response to Facebook's introduction of Libra [49]. Traditional finance normally has rules that creates a hierarchy of liability and accountability, based on contractual rather than technical or financial relationships (e.g. smart contracts in DeFi), where the supervised entity needs to ensure compliance from

all service providers connected to it [77]. In DeFi, there is no accountability or governing body overseeing how protocols are functioning and making sure they are compliant so user funds are not at risk. Regulatory risk is a critical risk of DeFi right now, as at any point if a regulator decides that DeFi investing is becoming out of control or find too many users have funds at risk, the regulator can enforce measures which ban populations of countries from interacting with DeFi platforms. Regulators often referred to ICOs as the wild west during the ICO boom of 2017. In many ways, DeFi is also turning into a type of wild west as financial innovation accelerates with little being done in the form of risk management, governance or regulation of DeFi protocols. One type of DeFi product in particular is drawing greater amounts of regulatory attention as time goes on are stablecoins (e.g. Facebook's Libra, or *MakerDAO's DAI).* Stablecoins are a core component in the DeFi ecosystem as they offer investors a way to mitigate volatility risk (if investors held cryptoassets such as *ETH* they would be exposed to high amounts of volatility, whereas stablecoins are normally pegged 1:1 to a dollar or have algorithmically low fluctuations). Stablecoins are a sort of safety net within the DeFi ecosystem and connect cryptoassets to real-world pricing (e.g. through oracles as we have learnt). Stablecoins are the most popular asset to deposit into borrowing / lending platforms in DeFi which is the most popular use-case for DeFi so far [52]. In fact there are now US $10 billlion of stablecoins that have been issued in the crypto ecosystem as of 16 June 2020 [55]. In April 2020, global financial watchdog G20 (which recommends operational measures to central banks) called for a worldwide consensus on the banning/supervision of all stablecoins, citing 10 risks of how stablecoins could substitute national currencies, which would be disastrous for countries as they lose sovereignty of their own dollar [28]. The FSB has requested public feedback on the recommendations with a deadline being set for July 15, three months before the final recommendations will be published [28]. If recommendations are to ban stablecoins by September 2020, DeFi could be in big trouble as all financial innovation that has been built so far could be halted if users cannot use stablecoins to interact with DeFi platforms. After all, DeFi's biggest platform, *MakerDAO*, which accounts for 50% of all *ETH* locked up in DeFi or $500 million USD, has its main function to issue stablecoins, which are then locked up and used in other DeFi protocols. Regulatory

risk could amplify all risks mentioned so far in this paper if one day a regulatory crackdown was to happen.

5.10 Finality Risk

Finality risk is the risk that *Ethereum* blockchain will fork, resulting in the creation of two different chains (resulting in DeFi assets being available on two or more chains, not one). Currently, the consensus (trust mechanism) of *Ethereum* is Proof-of-Work [23]. Compared to a traditional database, public blockchains using *Proof of Work* as consensus such as *Bitcoin* and *Ethereum* always have a probability of being reversed, a probability that often decays with time but which is never zero. Meaning transaction finality is probabilistic [4]. For any given block, there is always the possibility that someone will create a longer chain by re-ordering previous blocks in their favour and ignoring the true chain [5]. Probabilistic finality occurs when a transaction's finality increases as more blocks are added to the blockchain after the transaction [58]. The creator of *Ethereum*, Vitalik Buterin, addressed the risk of probabilistic finality of *Ethereum* in a blog post dating back from 2016:*if an attacker has less than 25% of network hashpower, then a model can be created where an attempted double spend as a random walk that starts at -6 (meaning "the attacker's double-spend chain is six blocks shorter than the original chain"), and at each step has a 25% chance of adding 1 (ie. the attacker makes a block and inches a step closer) and an 75% chance of subtracting 1 (ie. the original chain makes a block). We can determine the probability that this process will ever reach zero (ie. the attacker's chain overtaking the original) mathematically, via the formula* $(0.25 / 0.75)\hat{6} = 0.00137$. *If you want even greater certainty, you can wait 13 confirmations for a one-in-a-million chance of the attacker succeeding, and 162 confirmations for a chance so small that the attacker is literally more likely to guess your private key in a single attempt. Hence, some notion of de-facto finality even on proof-of-work blockchains does in fact exist.* [5]. It is common practice for enterprises using public *Ethereum* blockchain to wait six blocks before they consider their transactions final. It is a risk for DeFi users that something e.g. a 51% attack of *Ethereum* protocol could result in one chain continuing and one chain forking into a new direction (insert

fork image). The relationship between finality and DeFi is one of particular interest, as the DeFi ecosystem carries a few resemblances with that of *The DAO* (previously mentioned to have failed due to smart contract vulnerability risk). In the wake of *the DAO* hack, two proposals emerged as to how the *Ethereum* community would handle the loss of funds. 89% of the *Ethereum* community voted to hard fork the blockchain, so investors into *the DAO* smart contract would receive a 'refund' of the *ETH* they deposited. The creator of *the DAO* hack explained how the hard fork refunded in an article: *'The hard fork of the* Ethereum *blockchain moved the funds tied to* The DAO *to a new smart contract designed to to one thing: let the original token owners withdraw the funds. The token owners were given the original exchange rate of 1* ETH *to 100* DAO *tokens. The DAO* caused the last *ETH* hard fork in 2016. On 20th July 2016, at a block height of 1.92 million, *Ethereum* introduced an irregular state change via a hard fork in an effort to return approximately 3.6 million *ETH* that had been taken from a smart contract known as *The DAO* (US $50 million). Now *Ethereum* (where DeFi lives) exists on one chain and *ETC (ETH classic* which kept the same chain as the old *ETH* with addresses that had hacked funds) exists on another chain. At the time of *the DAO* hack, only $50m US was drained from the smart contract, yet the result was a hard fork into two chains. This makes the immutable aspect of the blockchain doubtful as if something in the past needs to be changed, it can be done. Keep in mind that this blockchain forked because of the governance of *Ethereum*, not because of an attacker diverting the course of a chain (e.g. with a 51% attack), meaning there is finality risk from both attackers of the protocol and governance of *Ethereum* itself. Finality risk gets its attributes from the consensus of a protocol. Buterin's explanation of risk of transactions being reverted from probabilistic finality addresses *Ethereums* current consensus mechanism – *Proof of Work*. His explanation also touches on how *Proof of Stake* (the consensus *Ethereum* is currently transitioning to in 2020) can be a better use of consensus in achieving probabilistic finality . The upgraded *Ethereum* network will switch from the proof-of-work to the proof-of-stake consensus algorithm, replacing miners with validators who will bet their coins to verify transactions. Once validators verify honest transactions, they will receive the rewards in the form of passive income — this process is called staking [40]. *Ethereum*'s transition to *Proof*

of Stake is a huge uncertainty to *Ethereum* blockchain itself, let alone it's impact on DeFi. The transition to *Proof of Stake* known as *Ethereum* Casper, will be an enormous security test for *Ethereum* and could make it more vulnerable to attacks and manipulation (by validators of the network i.e. stakers), hence being more exposed to complicated forks. In *Proof of Work*, miners cannot afford to validate transactions on different chains as they would lose any chance for reward (would not compute fast enough), whereas in *Proof of Stake* computation is far less, therefore it is much more affordable for validators to validate transactions on multiple chains simultaneously (with higher chances of earning rewards the more they validate on). This is also known as the *nothing at stake* security issue of PoS consensus where *double spending* issues can arise.

In July 2020, the first phase of the *Ethereum* 2.0 network is expected to go live. Called Phase 0, this initial evolution of the 2.0 network will launch the beacon chain and enable the *Proof of Stake* consensus mechanism [45]. It is important to note *Ethereum* 2.0 will not be a fork of *Ethereum*, it could just make forks more likely when it transitions to *Ethereum* 2.0. To mitigate against finality risk, *Ethereum* blockchain governance itself (i.e its creator Vitalik Buterin), has put several measures in place to counteract the possibility of validators acting maliciously on multiple chains, including slashing. It is far too early to tell whether these measures are sufficient enough to mitigate finality risk in the future. The risks endured due to *Proof of Stake* are speculative until phase 0 officially commences. Whether forks increase or decrease when *Ethereum* 2.0 is officially live remains to be seen. Any chance of a fork is a risk to DeFi. In late 2019, Dragonfly capital put out an article about how *Ethereum* will never be able to have a meaningful minority fork again, because of DeFi's inherent fragility [37]. Their reasoning was related to how stablecoin creators have great control over how the DeFi ecosystem handles a fork (as due to game theory it pays off better for them to work in unison into one fork together, rather than splitting the DeFi ecosystem). They gave the example of a centralised stablecoin owner e.g. *USDC* not allowing for redemption of their stablecoin into fiat money on certain chains if there was a chance of a fork. Given the composability of DeFi, the removal of *USDC* would have to be coordinated across the entire DeFi ecosystem. If *USDC* acted irrationally with disregard for the ecosystem being

split and chose one fork, on the *Ethereum* main chain derivatives and borrowing of *USDC* would be terminated, illiquidity would cause price plummets of *USDC* and bank runs could occur with lenders not being able to get their *USDC* fast enough from lending platforms. If one element of DeFi comes undone, e.g. a stablecoin only existing on one chain and not the other, composability risk ensures that financial contagion would occur within the DeFi ecosystem. For this reason, if a prominent DeFi player e.g. a stablecoin creator *(USDC, MakerDAO)*, chooses one fork and not the other, the whole DeFi ecosystem will most likely choose the same chain (as composability risk would be too costly to choose another), even if the fork is small and veers away from the major greater community-driven fork of *Ethereum*. This is a classic case of game theory where incentives favour coordination and therefore DeFi would move together [37]. DeFi platforms could be missing out on a lot of value in the minority fork though, e.g. *MakerDAO*, but due to *design risk*, *oracle risk* and *composability risk* of these DeFi platforms, liquidations would occur due to reliance on value of the underlying asset (ETH), reliance on price feeds (oracles) and reliance on external actors keeping the platform in tact (e.g. keepers for liquidations). The new stablecoin existing on the new forked chain would have no value due to the disfunction of *MakerDAO* on the new forked chain – so any other DeFi platforms wanting to use the new fork struggle as functionality is completely different and most operators don't have the infrastructure and deployment processes to manage their system on two chains, so many simply write it off [37]. In this sense perhaps DeFi itself mitigates finality risk, as more than likely major DeFi players will always choose the majority fork of *Ethereum*. Due to DeFi's reliance on one another, it always makes more sense for them to work cohesively rather than separately and it is unlikely DeFi platforms will have enough resources to operate on two forks of *Ethereum* making it unfeasible to have their platform on more than one chain. Whilst this could be a safety net for *Ethereum*, we cannot predict the future and there may be a situation where DeFi platforms choose different sides of a fork (or version of the *Ethereum* truth) for greater payoffs. This could lead to the demise of many platforms due to composability risk and therefore DeFi users should always be aware of any upcoming forks and their implications.

5.11 Disclosure Risk

Disclosure risk is the risk that a DeFi protocol has not disclosed a full list of risks a DeFi user could experience whilst using the platform. There is always a risk to a DeFi user that the user's chosen platform has not adequately disclosed the results of its products auditing reports. Not only that but even if a DeFi platform has been audited, it might have been audited prior to a protocol upgrade, where new vulnerabilities could have been introduced. As we know there is no accountability of DeFi platforms (due to the regulatory grey area), therefore DeFi protocols might have incentive to not disclose all risks of their platform if they think they can fix issues without the public knowing whilst continuing to receive funds. This is a dangerous tactic that has been deployed by DeFi platforms in the past. DeFi Score, a DeFi risk rating branch of *Consensys* entity, made an interesting observation about its risk rating system that failed to properly measure the *bZx* attack (previously mentioned in this paper) [12]. DeFi score previously had a binary measure of yes or no for whether a company had undergone an audit. This measure falsely made *bZx* platform look safe with a higher rating than they deserved, as they had conducted an audit in 2018 before major protocol upgrades. Before the *bZx* exploitation, the DeFi score framework did not take into consideration any other variables for outside of the binary yes or no for an audit conducted of a DeFi platform. After the exploit, DeFi Score upgraded their audit section to better reflect auditing risks of DeFi platforms for their risk rating system, by proposing a more robust and nuanced framework for better and more transparent evaluation of DeFi protocols. DeFi score introduced a new auditing risk framework in response to the *bZx* hack to better rate risk of DeFi platforms by including items such as the date of last audit, whether they received a new audit after a protocol upgrade and whether or not the audit undergone was public. The more improved and granular audit risk system tested better for *bZx*, as it gave them a lower audit scoring than a purely binary yes or no for audit conducted, resulting in *bZx* having a higher risk rating (which one would think would lead to lower funds on the platform) and could have eventuated in less funds being loss before the *bZx* exploit. DeFi users should try and be completely informed about any vulnerabilities found by previous audits into a DeFi platform to mitigate risk. There is a reliance on the DeFi protocol itself to

disclose vulnerabilities found in audits though, so even if a DeFi user is informed about the latest audit, a protocol upgrade could have been done after the audit that could introduce greater *smart contract vulnerability risk* and *design risk*. No accountability in DeFi leads to a greater disclosure risk of vulnerabilities not being properly reported and hence greater risk in DeFi overall. This is definitely an area where a governing body could help to make sure DeFi platforms are disclosing audits and vulnerabilities properly for users of DeFi's sake.

5.12 Risk of More Non-Financial Risks

Risk of more non-financial risks is the risk of more non-financial risks being found-out in the DeFi ecosystem that have not yet been accounted for. As this paper has shown, there are countless numbers of risks and each risk affects individual protocols differently. Innovation in DeFi is rapidly accelerating and we are seeing products being created that defy the very laws of finance itself i.e. with flash loans (no collateral and risk-free, win-win situation to arbitrage). New financial products that are not yet existent in the DeFi ecosystem include asset management, asset issuance and open market platforms, all of which are being created now and likely to exist in the future. These alone will come with their own risks that could be specific to how that platform operates. As we are seeing financial products and mechanisms being constantly created in DeFi, it is very likely that these will introduce new risks that are not mentioned in this paper and impossible to forecast. The DeFi ecosystem itself is highly risky and this paper has introduced twelve key non-financial risks in DeFi on *Ethereum* that are not found in traditional financial products. Using DeFi comes with traditional financial risks, as well as completely new types of non-financial risks. As the DeFi ecosystem grows, The likelihood of different and unforeseen risks will also increase. Blockchain is a nascent technology and there are a lot of attack vectors present in the DeFi ecosystem. Time will tell if DeFi can be truly adopted with its current inherent non-financial risks in the ecosystem.

Chapter 6

Future Work

This paper has highlighted key risks from a qualitative perspective. Future work could be done on finding more quantitative statistics of risks outlined in this paper. Furthermore, future work could revolve around creating risk management tools using these defined risks. An evaluation of current risk mitigation tools could also be an effective thought for future work, comparing risk mitigation tools found currently in the DeFi ecosystem to see if they properly mitigate risks defined in this paper.

Chapter 7

Conclusion

In conclusion, after extensive qualitative research this paper has introduced defined the twelve key non-financial risks in DeFi on Ethereum blockchain: scalability risk, smart contract vulnerability risk, oracle risk, design risk, composability risk, centrality risk, economic incentive risk, financial illiteracy risk, regulatory risk, finality risk, disclosure risk and the risk of more risks. Since 2017, DeFi has experienced a parabolic increase in value being used across all platforms. Not only has there been an exponential increase in value being used within DeFi protocols, but DeFi tokens have been created on top of protocols, which essentially give users of these DeFi protocols access to the revenue it generates, as well as a vote in governance proposals for the future of the protocol. Innovation in DeFi is incredible, but the risks of using DeFi products are greater than ever. Because DeFi is such a new innovation, using traditional financial products in a completely permissionless manner, more work needs to be done on studying the risks illustrated in this paper to create even more concrete definitions of risks. For now, DeFi remains to be stuck in the wild west stage of the lifecycle where investors invest liberally without fully understanding or considering the risks involved. This Thesis provides a framework for understanding the risks inherent in the DeFi ecosystem (particularly as they relate to Ethereum), which can assist the potential DeFi investor to make informed decisions regarding investment and risk management strategies. This also assists investors to make more educated decisions regarding returns relative to risk. Defining risks is the first step to creating better risk management in DeFi. This paper is the first scholarly work that I am aware of in this area to illustrate an exact definition of

risks and can be used as building blocks for future risk management in DeFi. DeFi innovation as outlined in this paper is ever expanding – so too can risk management if the ecosystem is educated about DeFi risks. Definitions provided in this paper are the foundations for future risk management work in DeFi.

References

[1] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. A survey of attacks on ethereum smart contracts (sok). In *POST*, volume 10204 of *Lecture Notes in Computer Science*, pages 164–186. Springer, 2017.

[2] BBC. Dforce hacker returns $25m in 'stolen' crypto-currencies. `www.bbc.com/news/technology-52368511/`, 2020. (Accessed June 10 2020).

[3] Hamda Al Breiki, Muhammad Habib Ur Rehman, Khaled Salah, and Davor Svetinovic. Trustworthy blockchain oracles: Review, comparison, and open research challenges. *IEEE Access*, 8:85675–85685, 2020.

[4] R Brown. When 'final' isn't actually final: Cracking blockchain's consensus conundrum. `www.forbes.com/sites/richardgendalbrown/2019/11/22/when-final-isnt-actually-final-cracking-blockchains-consensus-conundrum/#33a882636040/`, 2019. (Accessed July 20 2020).

[5] V Buterin. On settlement finality. `blog.ethereum.org/2016/05/09/on-settlement-finality/`, 2016. (Accessed July 20 2020).

[6] Chainlink. The key to unlocking smart contracts. `blog.chain.link/oracles-the-key-to-unlocking-smart-contracts/`, 2019. (Accessed June 17 2020).

[7] Chainlink. Defi's permissionless composability is supercharging innovation. `blog.chain.link/defis-permissionless-composability-is-supercharging-innovation`, 2020. (Accessed June 10 2020).

[8] Huashan Chen, Marcus Pendleton, Laurent Njilla, and Shouhuai Xu. A survey on ethereum systems security: Vulnerabilities, attacks and defenses, 2019.

[9] Circle. Can a customer send usdc tokens to any address? can addresses be blacklisted? support.usdc.circle.com/hc/en-us/articles/360016060352-Can-a-customer-send-USDC-tokens-to-any-address-Can-addresses-be-blacklisted-, 2020. (Accessed June 29 2020).

[10] CMC. Tether. coinmarketcap.com/currencies/tether/, 2020. (Accessed June 29 2020).

[11] CME. Defining ether and ethereum. www.cmegroup.com/education/courses/introduction-to-ether/defining-ether-and-ethereum.html, 2020. (Accessed 28 August 2020).

[12] Consensys CodeFi. The defi score in action: bzx report. codefi.consensys.net/blog/the-defi-score-in-action-bzx-report/, 2020. (Accessed August 08 2020).

[13] Consensys CodeFi. Security risks in ethereum defi. codefi.consensys.net/blog/security-risks-in-ethereum-defi/, 2020. (Accessed June 06 2020).

[14] Consensys. Why infura is the secret weapon of ethereum infrastructure. media.consensys.net/why-infura-is-the-secret-weapon-of-ethereum-infrastructure-af6fc7c77052/, 2018. (Accessed June 20 2020).

[15] Consensys. Ethereum smart contract best practices. consensys.github.io/smart-contract-best-practices/general_philosophy/, 2020. (Accessed June 07 2020).

[16] Consensys. The q1 2020 ethereum defi report. consensys.net/blog/news/the-q1-2020-ethereum-defi-report/, 2020. (Accessed 05 September 2020).

[17] V. Coutts. Black thursday — makerdao's multi collateral dai exploitation (and the plan to recover). medium.com/linum-labs/black-thursday-makerdaos-multi-collateral-dai-exploitation-and-the-plan-to-recover-c083c0b81875/, 2020. (Accessed 01 June 2020).

[18] B. Dale. Compound's approach to defi governance starts with giving away comp tokens. www.coindesk.com/compound-defi-governance-token-comp/, 2020. (Accessed 08 July 2020).

[19] Chris Dannen. *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Apress, USA, 1st edition, 2017.

[20] M. De Silva. Tether, not libra, may be the cryptocurrency behind bitcoin's surge. qz.com/1654449/is-tether-a-factor-in-bitcoins-price-surge/, 2019. (Accessed June 24 2020).

[21] Kevin Delmolino, Mitchell Arnett, Ahmed Kosba, Andrew Miller, and Elaine Shi. Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. volume 9604, pages 79–94, 02 2016.

[22] DigitalX. Why you shouldn't be using infura for your dapp. www.digitalx.com/post/why-you-shouldnt-be-using-infura-for-your-dapp/, 2019. (Accessed June 24 2020).

[23] Solidity Docs. Security considerations. solidity.readthedocs.io/en/latest/security-considerations.html. (Accessed June 09 2020).

[24] Synthetix Docs. Chainlink oracles. synthetix.community/docs/chainlink-oracles/, 2020. (Accessed 08 June 2020).

[25] Synthetix Docs. Synthetix docs. docs.synthetix.io/, 2020. (Accessed 08 June 2020).

[26] Compound Finance. Markets. compound.finance/markets/, 2020. (Accessed July 10 2020).

[27] W. Foxley. Everything you ever wanted to know about the defi 'flash loan' attack. www.coindesk.com/everything-you-ever-wanted-to-know-about-the-defi-flash-loan-attack/, 2020. (Accessed June 18 2020).

[28] FSB. Addressing the regulatory, supervisory and oversight challenges raised by "global stablecoin" arrangements. www.fsb.org/wp-content/uploads/P140420-1.pdf/, 2020. (Accessed July 14 2020).

[29] J. Hansen Grigo, A. H. Patz, and V. Von Wachter. Bitkom - decentralized finance (defi) - a new fintech revolution? www.bitkom.org/sites/default/

files/2020-07/200729_whitepaper_decentralized-finance.pdf/, 2020. (Accessed September 1 2020).

[30] Wanyun Catherine Gu, Anika Raghuvanshi, and Dan Boneh. Empirical measurements on pricing oracles and decentralized governance for stablecoins. *Available at SSRN 3611231*, 2020.

[31] Lewis Gudgeon, Daniel Perez, Dominik Harz, Benjamin Livshits, and Arthur Gervais. The decentralized financial crisis, 2020.

[32] Infura. Docs. infura.io/docs/, 2020. (Accessed July 04 2020).

[33] Robert A Jarrow. The role of abs, cds and cdos in the credit crisis and the economy. *Rethinking the Financial Crisis*, pages 210–234, 2011.

[34] Hugh Karp. Understanding risks in defi: Eth cc presentation. medium.com/nexus-mutual/understanding-risks-in-defi-eth-cc-presentation-4db9c7aedbb1/, 2020. (Accessed July 10 2020).

[35] Kyle Kistner. Composability must be included in risk assessment. github.com/ConsenSys/defi-score/issues/20/, 2019. (Accessed 20 June 2020).

[36] Stanley Kulechov. Composability must be included in risk assessment. github.com/ConsenSys/defi-score/issues/20/, 2019. (Accessed 20 June 2020).

[37] H. Leland, L & Qureshi. Ethereum is now unforkable, thanks to defi. medium.com/dragonfly-research/ethereum-is-now-unforkable-thanks-to-defi-9818b967738f/, 2019. (Accessed July 03 2020).

[38] Bowen Liu and Pawel Szalachowski. A first look into defi oracles. *arXiv preprint arXiv:2005.04377*, 2020.

[39] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. Making smart contracts smarter. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 254–269, 2016.

[40] J. Magas. Ethereum 2.0: The choice between one's own node and a staking service. cointelegraph.com/news/ethereum-20-the-choice-between-

ones-own-node-and-a-staking-service/, 2020. (Accessed on July 01 2020).

[41] MakerDAO. Maker - whitepaper. makerdao.com/en/whitepaper/\#abstract/, 2017. (01 June 2020).

[42] MakerDAO. Decentralized finance (defi) trends. blog.makerdao.com/decentralized-finance-defi-trends/, 2020. (June 19 2020).

[43] Alexander Mense and Markus Flatscher. Security vulnerabilities in ethereum smart contracts. In *Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services*, pages 375–380, 2018.

[44] B Munster. Compound's smart contracts could pose risk to ether, dai holders. decrypt.co/8932/compounds-smart-contracts-could-pose-risk-to-ether-dai-holders/, 2019. (Accessed June 19 2020).

[45] E. Muzzy. What happens to my eth on ethereum 2.0? consensys.net/blog/blockchain-explained/what-happens-to-my-eth-on-ethereum-2/, 2020. (Accessed on July 18 2020).

[46] M. Nystrom. 2019 was the year of defi (and why 2020 will be too). consensys.net/blog/news/2019-was-the-year-of-defi-and-why-2020-will-be-too/, 2019. (Accessed July 20 2020).

[47] Trail of Bits. Contract upgrade anti-patterns. blog.trailofbits.com/2018/09/05/contract-upgrade-anti-patterns/, 2018. (Accessed June 06 2020).

[48] R O'Leary. The race is on to replace ethereum's most centralized layer. finance.yahoo.com/news/dev-ethereum-may-fail-relies-231527712.html/, 2018. (Accessed July 09 2020).

[49] S. O'Neal. Facebook libra regulatory overview: Major countries' stances on crypto. cointelegraph.com/news/libra-vs-us-congress-all-there-is-to-know-ahead-of-hearings/, 2019. (Accessed July 10 2020).

[50] Daniel Perez and Benjamin Livshits. Smart contract vulnerabilities: Does anyone care? *arXiv preprint arXiv:1902.06710*, 2019.

[51] Purathani Praitheeshan, Lei Pan, Jiangshan Yu, Joseph Liu, and Robin Doss. Security analysis methods on ethereum smart contract vulnerabilities: a survey. *arXiv preprint arXiv:1908.08605*, 2019.

[52] DeFi Pulse. Total value locked. Available at: `defipulse.com/`, 2020. (Accessed July 27 2020).

[53] Kaihua Qin, Liyi Zhou, Benjamin Livshits, and Arthur Gervais. Attacking the defi ecosystem with flash loans for fun and profit. *arXiv preprint arXiv:2003.03810*, 2020.

[54] H. Qureshi. The defi 'flash loan' attack that changed everything. `finance.yahoo.com/news/defi-flash-loan-attack-changed-172255685.html/`, 2020. (Accessed July 10 2020).

[55] H Reanudin. Driven by financial institutions, stablecoin acceptance turns a corner. `cointelegraph.com/news/driven-by-financial-institutions-stablecoin-acceptance-turns-a-corner/`, 2020. (Accessed July 03 2020).

[56] A. Shevchenko. Dforce hacker returns stolen money as criticism of the project continues. `cointelegraph.com/news/dforce-hacker-returns-stolen-money-as-criticism-of-the-project-continues/`, 2020. (Accessed June 15 2020).

[57] Smith and Crown. The oracle problem and mixicles. `sci.smithandcrown.com/research/oracle-problem-and-mixicles/`, 2019. (June 04 2020).

[58] Smith and Crown. Transaction finality (probabilistic/deterministic. `sci.smithandcrown.com/glossary/transaction-finality-probabilisticdeterministic/`, 2019. (Accessed July 11 2020).

[59] S. Srinawakoon. Smart contract oracles and price feed centralization. `medium.com/bandprotocol/smart-contract-oracles-and-price-feed-centralization-e74dfa8695af/`, 2019. (Accessed 16 June 2020).

[60] R. Stevens. Why dai stablecoin can now be backed by...usdc stablecoin. decrypt.co/22866/usdc-dai-makerdao-crypto/, 2020. (Accessed 03 July 2020).

[61] Tianyu Sun and Wensheng Yu. A formal verification framework for security issues of blockchain smart contracts. *Electronics*, 9(2):255, 2020.

[62] Synthetix. Chainlink decentralizes first wave of synthetix price feeds! blog.synthetix.io/chainlink-decentralizes-first-wave-of-synthetix-price-feeds/, 2019. (Accessed 12 June 2020).

[63] Synthetix. Response to oracle incident. blog.synthetix.io/response-to-oracle-incident, 2019. (Accessed 06 June 2020).

[64] J. Thevenard. Decentralised oracles: a comprehensive overview. medium.com/fabric-ventures/decentralised-oracles-a-comprehensive-overview-d3168b9a8841/, 2019. (Accessed June 01 2020).

[65] Min Tian and Cai Wei. A security case study for blockchain games. In *2019 IEEE Games, Entertainment, Media Conference (GEM)*, pages 1–8. IEEE, 2019.

[66] R. Todd. Synthetix suffers oracle attack, more than 37 million synthetic ether exposed. finance.yahoo.com/news/synthetix-suffers-oracle-attack-potentially-224737187.html/, 2019. (Accessed June 02 2020).

[67] Tokenlon. imbtc fully restored (transfer, trading, mint, and redeem). https://tokenlon.zendesk.com/hc/en-us/articles/360042172392-imBTC-fully-restored-transfer-trading-mint-and-redeem/, 2020. (Accessed June 11 2020).

[68] Christof Ferreira Torres, Julian Schütte, and Radu State. Osiris: Hunting for integer bugs in ethereum smart contracts. In *Proceedings of the 34th Annual Computer Security Applications Conference*, pages 664–676, 2018.

[69] Cooper Turley. Compound finance review. defirate.com/compound-finance/, 2020. (Accessed June 28 2020).

[70] Uniswap. Uniswap documentation. uniswap.org/docs/v2/, 2020. (Accessed July 06 2020).

[71] DeFi Watch. What is admin key risk? project reviews. defiwatch.net/admin-key-config-and-opsec/project-reviews/, 2020. (Accessed June 20 2020).

[72] Lee Wei-Meng. *Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript*. 01 2019.

[73] Whiterabbit. Black thursday for makerdao: $8.32 million was liquidated for 0 dai. medium.com/@whiterabbit_hq/black-thursday-for-makerdao-8-32-million-was-liquidated-for-0-dai-36b83cac56b6/, 2020. (Accessed June 01 2020).

[74] A. Xie. For defi's sake, maker should take blame for black thursday losses. www.coindesk.com/for-defis-sake-maker-should-take-blame-for-black-thursday-losses/, 2020. (Accessed June 01 2020).

[75] J. Young. Dev: Ethereum may fail if it relies on infura to run nodes, potential solution. finance.yahoo.com/news/dev-ethereum-may-fail-relies-231527712.html/, 2018. (Accessed July 09 2020).

[76] Open Zeppelin. Exploiting an erc777-token uniswap exchange. github.com/OpenZeppelin/exploit-uniswap\#why-it-works/, 2019. (Accessed June 11 2020).

[77] Dirk A Zetzsche, Douglas W Arner, and Ross P Buckley. Decentralized finance (defi). *IIEL Issue Brief*, 2, 2020.