Decentralized Finance: On Blockchain- and Smart Contract-based Financial Markets

Fabian Schär
University of Basel

Abstract – This paper explores the Decentralized Finance (DeFi) ecosystem. We examine how DeFi is emerging on top of the public Ethereum smart contract platform, compare it to the centralized architecture of traditional financial markets and highlight opportunities and potential risks of this ecosystem. We propose a multi-layered framework to analyze the implicit architecture and the various DeFi building blocks, including token standards, decentralized exchanges, decentralized debt markets, blockchain derivatives and on-chain asset management protocols. We conclude that DeFi still is a niche market with certain risks, but also has interesting properties in terms of efficiency, transparency, accessibility and interoperability. As such, it may potentially contribute to a more robust and transparent financial infrastructure. (JEL G15, G23, E59)

1 INTRODUCTION

Decentralized Finance (DeFi) is a movement in the Blockchain space that has recently gained a lot of traction. This term generally refers to open financial infrastructures built upon public smart contract platforms, such as the Ethereum blockchain (see Buterin 2013).

In contrast to the traditional financial sector, DeFi does not rely on intermediaries and centralized institutions. Instead, it is based on open protocols and decentralized applications (DApps). Agreements are enforced with smart contracts, transactions executed in a secure and deterministic way and legitimate state changes persisted on a public Blockchain. Thus, this architecture is capable of creating an immutable and highly interoperable financial system with unprecedented transparency, equal access rights and little need for custodians, central clearing houses or escrow services, as most of these roles can be assumed by smart contracts.

DeFi already offers a broad variety of applications. One can, for example, buy USD-pegged stablecoins on decentralized exchanges, move these tokens to an equally decentralized lending platform to earn interest and subsequently add the tokenized interest-bearing instruments to a decentralized liquidity pool or an on-chain investment fund.

The backbone of all DeFi protocols and applications are so-called smart contracts, a term that generally refers to small applications stored on a Blockchain and executed by a large network of many computers. Smart contracts are relatively inefficient

compared to traditional centralized computing. Their advantage is a high level of security, in the sense that smart contracts guarantee deterministic execution and allow anyone to verify the resulting state changes. When implemented in a secure manner, smart contracts are highly transparent and minimize the risk of manipulation and arbitrary intervention.

To understand the novelty of smart contracts, we first have to look at regular server-based web applications. When someone interacts with such an application, this person is unable to observe the application's internal logic. Moreover, this person is not in control of the execution environment. Either one (or both) could be manipulated. As a result, the user has to trust the application service provider. Smart contracts mitigate both of these problems and ensure that an application runs exactly as expected. The contract code is stored on the underlying blockchain and can therefore be publicly scrutinized. The behavior of the contract is deterministic and function calls (in the form of transactions) are processed by hundreds of network participants in parallel, ensuring the legitimacy of the execution. When the execution leads to state changes, e.g. the change of account balances, these changes are subject to the Blockchain network's consensus rules and will be reflected in and protected by the state tree of the blockchain.

Smart contracts have access to a Turing-complete instruction set and are therefore quite flexible. Additionally, they are able to store cryptoassets and thereby assume the role of a custodian, with completely customizable criteria for how, when and to whom these assets can be released. This allows for a large variety of interesting applications and flourishing ecosystems.

The original concept of a smart contract was coined by Szabo (1994). Szabo (1997) used the example of a vending machine to further describe the idea and argued that many agreements could be "embedded in the hardware and software we deal with, in such a way as to make breach of contract expensive for the breacher." Buterin (2013) proposed a blockchain-based smart contract platform to solve any trust issues regarding the execution environment and to enable secure global states. Additionally, this platform allows the contracts to interact with each other. The concept was further formalized by Wood (2015) and implemented under the name Ethereum. Although there are many alternatives, Ethereum is by far the largest smart contract platform, in terms of market cap, available applications and development activity.

DeFi still is a niche market with relatively low volumes – however, these numbers aregrowing rapidly. The value of funds that are locked in DeFi related smart contracts recently reached USD 1 billion. It is important to understand that these are not transaction volume or market cap numbers; the value refers to reserves that are locked in smart contracts for use in a variety of ways that will be explained in the course of this paper. Figure 1 shows the Ether (ETH) and USD values of the assets that are locked in DeFi applications.

Figure 1

Total Value Locked in DeFi Contracts (USD and ETH).

Data Source: DeFi Pulse

2 DEFI BUILDING BLOCKS

DeFi uses a multi-layered architecture. Every layer has its own distinct purpose. In this section we propose a framework for the analysis of these layers and study the token and the protocol layer in greater detail.[1]

We differentiate between 5 layers: the *settlement*, *asset*, *protocol*, *application* and *aggregation* layers.

The *settlement layer* (1) consists of the Blockchain and its native protocol asset. It allows the network to securely store ownership information and ensures that any of the state changes adhere to the network's rule set. As such, the Blockchain can be seen as the foundation for trustless execution and serves as a settlement and dispute resolution layer.

The *asset layer* (2) consists of all tokens that are issued on top of the settlement layer. This includes the native protocol asset as well as any additional tokens that are based on token standards supported by the Blockchain.

The *protocol layer* (3) provides standards for specific use-cases such as decentralized exchanges, debt markets, derivatives and on-chain asset management. These standards are usually implemented as a set of smart contracts and can be accessed by any user (or DeFi application). As such, these protocols are highly interoperable.

The *application layer* (4) creates user-oriented applications that connect to individual protocols. The smart contract interaction is usually abstracted by a web browser-based front end, making the protocols easier to use.

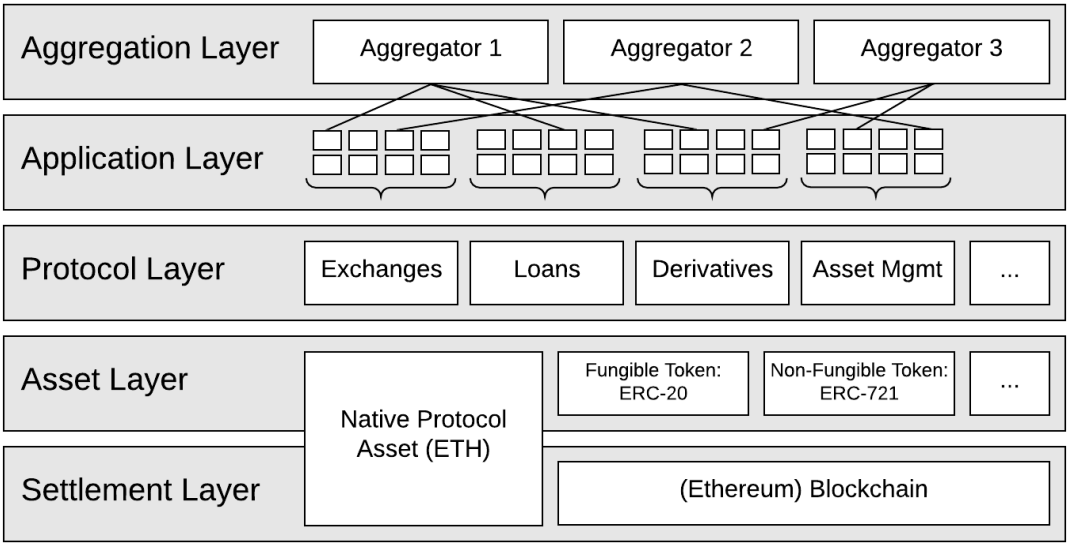
The *aggregation layer* (5) is an extension of the application layer. Aggregators create user-centric platforms that connect to several applications and protocols. They usually provide tools to compare and rate and services, allow users to easily perform

otherwise complex tasks by connecting to several protocols simultaneously, and finally combine relevant information in a clear and concise manner.

It is important to understand that these layers are hierarchical, in the sense that they are only as secure as the layers below. If, for example, the Blockchain in the settlement layer is compromised, all of the subsequent layers would be unsecure. Similarly, if we would use a permissioned ledger as the foundation, any decentralization efforts on subsequent layers would be ineffective.

Figure 2 visualizes the Decentralized Finance stack and its layers.

Figure 2
The Decentralized Finance Stack



Now that we have an understanding of the conceptual model, let us take a closer look at the tokenization and protocol layer. After a short introduction to asset tokenization, we will look into decentralized exchange protocols, decentralized lending platforms, decentralized derivatives and on-chain asset management. This allows us to establish the foundation needed for our analysis of the potential and risks.[2]

2.1 Asset Tokenization

Public blockchains are databases that allow participants to establish a shared and immutable record of ownership. Usually this is used to track the native protocol asset of the respective blockchain, e.g. Bitcoin (BTC) on the Bitcoin blockchain and Ether (ETH) on the Ethereum blockchain. But when this technology became more popular, so was the idea of making additional assets available on these ledgers. The process of adding new assets to a blockchain is called tokenization, and the blockchain representation of the asset is referred to as a token.

The general idea of tokenization is to make assets more accessible and transactions more efficient. In particular, tokenized assets can be transferred easily and within seconds from and to anyone in the world. They can be used in many decentralized

applications and stored within smart contracts. As such, these tokens are an essential part of the DeFi ecosystem.

From a technological perspective, there are various ways in which public Blockchain tokens can be created (see Roth et al. 2019). However, most of these options can be disregarded, as the vast majority of tokens are issued on the Ethereum blockchain through a smart contract template referred to as the ERC-20 token standard (Vogelsteller and Buterin 2015). These tokens are interoperable and can be used in almost all DeFi applications. As of September 2019, there are over 200,000 ERC-20 token contracts deployed on Ethereum.[3] Table 1 shows the number of tokens that are listed on exchanges and the aggregated token market cap in USD per Blockchain. Almost 90% of all listed tokens are issued on the Ethereum Blockchain. The slight deviation in terms of market cap originates from the fact that a relatively large portion of the stablecoin USDT is issued on Omni.

Table 1
Listed Tokens and Total Token Market Cap by Blockchain Platform

	Number		Market Capitalization (USD)	
Platform	**Absolute**	**Relative**	**Absolute**	**Relative**
Ethereum	1'311	89.24%	14'879'102'605	82.07%
Neo	28	1.91%	115'982'772	0.64%
Waves	24	1.63%	33'387'555	0.18%
Stellar	23	1.57%	354'051'529	1.95%
EOS	14	0.95%	44'979'951	0.25%
BitShares	12	0.82%	18'357'351	0.10%
Binance Coin	11	0.75%	46'637'426	0.26%
Qtum	9	0.61%	14'157'113	0.08%
Nem	6	0.41%	18'265'758	0.10%
VeChain	5	0.34%	3'818'174	0.02%
Tron	5	0.34%	281'139'575	1.55%
Omni	4	0.27%	2'185'678'477	12.06%
Others	17	1.16%	134'296'983	0.74%

Data sources: coinmarketcap.com and tether.to per September 17, 2019. Data preparation in the style of Roth et. al (2019).

From an economic perspective, we are more interested in the nature of the asset than in the underlying technical standard that is used to implement the asset's digital representation. One of the main issues in terms of economics is counterparty risk. Native digital tokens, including the aforementioned protocol assets (BTC, ETH), are unproblematic in this regard. In contrast, when someone introduces tokens with a promise, e.g. interest payments, dividends or the delivery of a good or service, the

corresponding token's value will depend on the credibility of this claim. If an issuer is unwilling or unable to deliver, the token will be worthless.

Generally speaking, there are three backing models for promise-based tokens: backed by *off-chain collateral*, *on-chain collateral* and *no collateral*. Off-chain collateral means that the underlying assets are stored with an escrow service, e.g. a commercial bank. On-chain collateral means that the assets are locked on the Blockchain, usually within a smart contract.[4] When there is no collateral, counterparty risk is at its highest. In this case, the promise is entirely trust-based. Berentsen and Schär (2019) have analyzed the three categories in the context of stablecoins, i.e. low-volatility cryptocurrencies that are pegged to the USD.

On-chain collateral has several advantages. It is highly transparent and claims can be secured by smart contracts, allowing processes to be executed in a semi-automatic way. A main disadvantage of on-chain collateral is that this collateral is usually held in a native protocol asset (or a derivative thereof) and therefore will experience price fluctuations. Take the example of the DAI stablecoin, which mainly uses ETH as its on-chain collateral to create a decentralized and trustless DAI-token that is pegged to the value of 1 USD. Whenever anyone wants to issue new DAI-tokens, this person first needs to lock enough ETH as underlying collateral in a smart contract provided by the Maker Protocol. Since the USDETH exchange rate is not fixed, there is a need for overcollateralization. If the value of the underlying ETH collateral at any point falls below the minimum threshold of 150% of the outstanding DAI value, the smart contract will auction off the collateral to cancel the debt in DAI.

Figure 3 shows some key metrics of the DAI stablecoin including price, total DAI in circulation and the stability fee, i.e. the interest rate that has to be paid by anyone who is creating new DAI (see Section 2.3).

There are also several examples for off-chain collateralized stablecoins. The most popular ones are USDT and USDC. They are both available as ERC-20 tokens on the Ethereum blockchain. DGX is an ERC-20 based stablecoin backed by gold and WBTC is a tokenized version of Bitcoin, making Bitcoin available on the Ethereum blockchain. The problem of tokens that use off-chain collateral is that they require regular audits and precautionary measures to make sure that the underlying collateral is available at all times. This process is costly and, in many cases, not completely transparent to the token holders.

While we are unaware of any working designs for unbacked stablecoins, i.e. stablecoins that do not use any form of collateral to maintain the peg, there are several organizations working on that idea.

Although stablecoins serve an important role in the DeFi ecosystem it would not do justice to the subject of tokenization to limit the discussion to these assets. There are all kinds of tokens that serve a variety of purposes, including governance tokens for decentralized autonomous organizations (DAO), tokens that allow the holder to perform certain actions in a smart contract, tokens that resemble shares, bonds and even synthetic tokens that can track the price of any real-world asset.

Figure 3
DAI Stablecoin Key Metrics

Data Sources: DeFi Pulse, Coinmarketcap

Another very interesting category are so-called non-fungible tokens (NFTs). NFTs are tokens that represent unique asset, i.e. collectibles. They can either be the digital representation of a physical object like a piece of art, making them subject to the usual counterparty risk or a digitally-native unit of value with unique characteristics. In any case the non-fungibility characteristics of the token ensures that the ownership of each asset can be individually tracked and the asset precisely identified. NFTs usually are built upon the ERC-721 token standard (Entriken et. al 2018).

In the following sections we will discuss the protocol layer and examine how tokens can be traded using decentralized exchanges (Section 2.2), how they can be used as collateral for loans (Section 2.3), to create decentralized derivatives (Section 2.4) and how they can be included in on-chain investment funds (Section 2.5).

2.2 Decentralized Exchange Protocols

As of September 2019, there are over 2,800 cryptoassets[5] listed on exchanges. While most of them are economically irrelevant and have a negligible market cap and trading volume, there surely is a need for marketplaces where people can trade the more popular ones. This allows them to rebalance their exposure in accordance with their preferences and risk profiles and to adjust portfolio allocations.

In most cases, these trades are conducted through centralized exchanges. Centralized exchanges are relatively efficient but they have one severe problem. In order to be able to trade on a centralized exchange, traders must first deposit assets with the exchange. They thereby forfeit direct access to their assets and have to trust the exchange operator. Dishonest or unprofessional exchange operators may confiscate or lose the assets. Moreover, centralized exchanges create a single point of attack and therefore face the constant threat of becoming the target of malicious third parties. Both problems are intensified by the relatively low regulatory scrutiny and the immense scaling efforts many of these exchanges had to go through within a short time period. Accordingly, it is rather unsurprising that we have witnessed many cases in which centralized cryptoasset exchanges have lost customer funds.

Decentralized exchange protocols try to mitigate these issues by removing the trust requirement. Users no longer have to deposit their funds with a centralized exchange. Instead, they remain in exclusive control of their assets until the trade is executed. Trade execution happens atomically through a smart contract, meaning that both sides of the trade are performed in one indivisible transaction, mitigating the counterparty credit risk. Depending on the exact implementation, the smart contract may assume additional roles, effectively making many intermediaries such as escrow services and central counterparty clearing houses (ccp) obsolete.

Early decentralized exchanges such as *EtherDelta* have been set up as walled gardens with no interaction between the various implementation. In particular, there was no shared liquidity, leading to relatively low transaction volumes and large bid/ask spreads. High network fees as well as cumbersome and slow processes to move funds between these decentralized exchanges have rendered supposed arbitrage opportunities useless.

More recently, there has been a move towards open exchange protocols. These projects try to streamline the architecture of decentralized exchanges by providing standards on how asset exchange can be conducted, and allowing any exchange that is built on top of the protocol to use shared liquidity pools and other protocol features. However, most importantly, other DeFi protocols can make use of these marketplaces and exchange or liquidate tokens when needed.

In the following subsections, we compare various types of decentralized exchange protocols, some of which are not exchanges in the narrow sense, but have been included in our analysis, as they serve the same purpose. The results are summarized in table 2.

Table 2
Most Popular Decentralized Exchange Protocols

Protocol Name	Protocol Type	Price Discovery
0x	Exchange	Off-Chain Order Books
(Air)Swap	P2P / OTC	P2P Negotiation
Bancor	Liquidity Pool	Smart Contract (CRR)
Kyber Network	Reserve Aggregator	Proposal by Maker
UniSwap	Liquidity Pool	Smart Contract (CPM)

Decentralized Order Book Exchanges

Decentralized order book exchanges can be implemented in a variety of ways. They all use smart contracts for transaction settlement but they differ significantly in the way the order books are hosted. In particular, one has to distinguish between on-chain and off-chain order books.

On-chain order books have the advantage of being completely decentralized. Every order is stored within the smart contract. As such there is no need for additional infrastructure or third party hosts. The disadvantage of this approach is that every action requires a blockchain transaction. It therefore is a very expensive and slow process, for which even the declaration of the intent to trade results in network fees. Considering that volatile markets will require frequent order cancellations, the problem becomes even more severe.

For this reason, many decentralized exchange protocols rely on off-chain order books and only use the blockchain as a settlement layer. Off-chain order books are hosted and updated by centralized third parties, usually referred to as relayers. They provide takers with the information they need to select an order they would like to match. While this approach certainly introduces some centralized components and dependencies to the system, the relayers' role is limited. In particular, relayers are never in control of the funds and neither match nor execute the orders. They simply provide ordered lists with quotes and may charge a fee for that service. The openness of the protocol ensures that there is competition between the relayers and mitigates potential dependencies.

The dominant protocol that uses this approach is called 0x (Warren and Bandeali, 2017). The protocol uses a three-step approach for trades. First, the maker sends a pre-signed order to the relayer for inclusion in the order book. Second, a potential taker queries the relayer and selects one of the orders. Third, the taker signs and submits the order to the contract, triggering the atomic exchange of the cryptoassets.

Smart Contract-Based Liquidity Pools

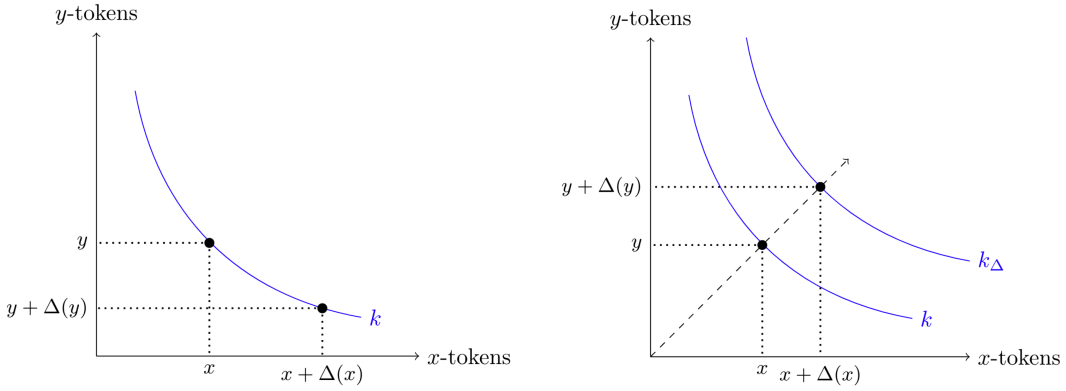
A liquidity pool is a smart contract that holds (at least) two cryptoassets in reserve and allows anyone to deposit tokens of one type and thereby withdrawing tokens of the other type. To determine the exchange rate, smart contract-based liquidity pools use

variations of the constant product model, where the relative price is a function of the smart contract's token reserve ratio. The earliest implementation we are aware of was proposed by Hertzog et. al (2017). Adams (2018) has simplified the model and Zhang et. al (2018) provide a formal proof of the concept.

In its simplest form, the constant product model can be expressed as $xy = k$, where x and y correspond to the smart contract's token reserves and k is a constant. Considering that this equation must hold, when someone executes a trade, we get $(x + \Delta x) \cdot (y + \Delta y) = k$. It can then be easily shown that $\Delta y = \frac{k}{x+\Delta x} - y$. Consequently, Δy will assume negative values for any $\Delta x > 0$. In fact, any exchange corresponds to a move on a convex token reserve curve. This is shown in Figure 3a. A liquidity pool using this model cannot be depleted, as tokens will get more expensive with lower reserves. When the token supply of either one of the two tokens approaches zero, its relative price rises infinitely as a result.

Figure 3

Visualization of Liquidity Pool Token Reserves with Constant Product Model



It is important to point out that smart contract-based liquidity pools are not reliant on external price feeds (so-called oracles). Whenever the market price of an asset shifts, anyone can use the arbitrage opportunity and trade tokens with the smart contract until the liquidity pool price converges to the current market price. The implicit bid/ask spread of the constant product model (plus a small trading fee) may lead to the accumulation of additional funds. Anyone who provides liquidity to the pool receives pool share tokens that allow them to participate in this accumulation and to redeem these tokens for their share of a potentially growing liquidity pool. Liquidity provision results in a growing k and is visualized in Figure 3b.

Two examples of smart contract-based liquidity pool protocols are Bancor and UniSwap. Due to its simpler design, UniSwap is somewhat more optimized in terms of network fees.

Smart Contract-Based Reserve Aggregation

Another approach is to consolidate liquidity reserves through a smart contract that allows large liquidity providers to connect and advertise prices for specific trade pairs.

A user who wants to exchange token x for token y may send a trade request to the smart contract. The smart contract will compare prices from all liquidity providers, accept the best offer on behalf of the user and execute the trade. As such, the smart contract acts as a gateway between users and liquidity providers, ensuring best execution and atomic settlement.

In contrast to smart contract-based liquidity pools, prices are not determined within the smart contract. Instead, prices are set by the liquidity providers. This works fine as long as there is a relatively broad base of liquidity providers. However, if there is limited or no competition for a given trade pair, the approach may result in collusion risks or even monopolistic price setting. As a countermeasure, reserve aggregation protocols usually have some (centralized) control mechanisms, like maximum prices or minimum number of liquidity providers. In some cases, liquidity providers may only participate after a background check, including KYC.

The best-known implementation of this concept is the Kyber Network (Luu and Velner, 2017), which serves as a backbone protocol for a large variety of decentralized finance applications.

Peer-to-Peer (P2P) / Over-the-Counter (OTC) Protocols

An alternative to classic exchange or liquidity pool models are peer-to-peer (p2p) protocols. They mostly rely on a two-step approach, where participants can query the network for counterparties who would like to trade a given pair of cryptoassets and then negotiate the exchange rate bilaterally. Once the two parties agree on a price, the trade is executed on-chain via a smart contract. In contrast to other protocols, offers can be accepted exclusively by the parties who have been involved in the negotiation. In particular, it is not possible for a third party to front run someone accepting an offer, by observing the pool of unconfirmed transactions (mempool).

To make things more efficient, the process is usually automated. Additionally, one can use off-chain indexers for peer discovery. These indexers basically assume the role of a directory, in which people can advertise their intent to trade a pair. Note that these indexers only serve to establish a connection. Prices are still negotiated P2P.

(Air)Swap is the most popular implementation of a decentralized P2P/OTC protocol. It has been proposed by Oved and Mosites (2017).

2.3 Decentralized Lending Platforms

Loans are an essential part of the DeFi ecosystem. There is a large variety of protocols that allow people to lend and borrow cryptoassets. Decentralized loan platforms are special in the sense that they require neither the borrower nor the lender to identify themselves. Everyone has access to the platform and can potentially borrow money or provide liquidity to earn interest. As such, DeFi loans are completely permissionless and not reliant on trusted relationships.

In order to protect the lender and stop the borrower from running away with the funds, there are two distinct approaches:

First, credit can be provided under the condition that the loan must be repaid atomically, meaning that the borrower receives the funds, uses and repays them – all within the same Blockchain transaction. If the borrower has not returned the funds (plus interest) at the end of the transaction's execution cycle, the transaction will be invalid and any of its results (including the loan itself) reverted. These so-called flash loans (Wolff, 2018; Boado, 2020) are a very interesting, but still highly experimental application. Although there are not too many known use-cases besides arbitrage, flash loans could potentially mature to become an important part of DeFi lending.

Second, loans can be fully secured with collateral. The collateral is locked in a smart contract and only released once the debt is repaid. Collateralized loan platforms exist in three variations: *Collateralized debt positions*, *pooled collateralized debt markets* and *P2P collateralized debt markets*. Collateralized debt positions are loans that use newly created tokens while debt markets use existing tokens and require a match between a borrowing and a lending party. The three variations are discussed below.

Collateralized Debt Positions

Some DeFi applications allow users to create collateralized debt positions and thereby issue new tokens which are backed by the collateral. To be able to create these tokens, the person has to lock cryptoassets in a smart contract. The number of tokens that can be created depends on the target price of the tokens which are being generated, the value of the cryptoassets that are being used as collateral and the target collateralization ratio. The newly created tokens are essentially fully collateralized loans that do not require a counterparty and allow the user to get a liquid asset, while maintaining market exposure through the collateral. The loan can be used for consumption, allowing the person to overcome a temporary liquidity squeeze or to acquire additional cryptoassets for leveraged exposure.

To illustrate the concept let us use the example of Maker DAO, a decentralized protocol that is used to issue the USD-pegged DAI stablecoin. First, the user deposits ETH in a smart contract, i.e. the CDP (or vault). Subsequently, he or she calls a contract function to create and withdraw a certain number of DAI and thereby locks the collateral. This process currently requires a minimum collateralization ratio of 150%, meaning that for any USD 100 worth of ETH that are locked up in the contract, the user can create at most 66.66 DAI.[6]

Any outstanding DAI are subject to a stability fee, which in theory should correspond to the maximum interest rate of the DAI debt market. This rate is set by the community, namely the MKR token holders. MKR is the governance token for the Maker DAO project. As shown in Figure 2, the stability fee has been fluctuating wildly between 0% and 20%.

To close a CDP, the owner has to send the outstanding DAI plus the accumulated interest to the contract. The smart contract will allow the owner to withdraw their collateral, once the debt is repaid. If the borrower fails to repay the debt, or if the collateral's value falls below the 150% threshold, where the full collateralization of the loan is at risk, the smart contract will start to liquidate the collateral at a potentially discounted rate.

Interest payments and liquidation fees are partially used to burn MKR, thereby decreasing total MKR supply. In exchange, MKR holders assume the residual risk of extreme negative ETH price shocks, which may lead to a situation in which the collateral is insufficient to maintain the USD-peg. In this case, new MKR will be created and sold at a discounted rate. As such, MKR holders have skin in the game and it should be in their best interest to maintain a healthy system.

It is important to mention that the MakerDAO system is much more complex than what is described here. Although the system is mostly decentralized, it is reliant on price oracles. This introduces some dependencies, which will be discussed in Section 3.2.

MakerDAO has recently switched to a multi-collateral system, with the goal to make the protocol more scalable by allowing a variety of cryptoassetes to be used as collateral.

Collateralized Debt Markets

Instead of creating new tokens, it is also possible to borrow existing cryptoassets from someone else. For obvious reasons, this approach requires a counterparty with opposing preferences. In other words: For someone to be able to borrow ETH, there must be another person willing to lend ETH. To mitigate counterparty risk and to protect the lender, loans must be fully collateralized and the collateral locked in a smart contract – just as in our previous example.

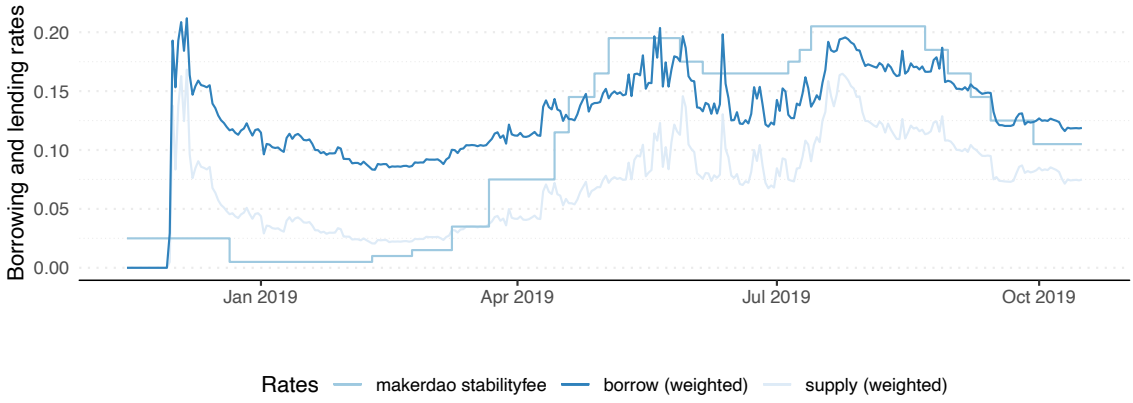
Matching between lender and borrower can be done in a variety of ways. The broad categories are P2P and pooled matching. P2P matching means that the person who is providing the liquidity lends the cryptoassets to specific borrowers. Consequently, the lender will only start to earn interest once there is a match. The advantage of this approach is that the parties could potentially agree on a time period and operate with fixed interest rates.

Pooled loans use variable interested rates that are subject to supply and demand. The funds of all borrowers are aggregated in a single, smart contract-based lending pool and lenders start to earn interest right when they deposit their funds to the pool. However, the interest rates are a function of the pool's utilization rate. When liquidity is readily available, loans will be cheap. When it is in great demand, loans will become more expensive. Lending pools have the additional advantage that they can easily perform maturity and size transformation while maintaining a relatively high liquidity for the individual lender.

There is a large variety of lending protocols. Some of the most popular ones are Compound (Leshner and Hayes, 2019), dYdX (Juliano, 2017) and bZx.[7] Figure 4 shows the asset-weighted borrowing and lending rates for DAI and ETH. In the case of DAI, we have also included the Maker DAO stability fee, which should always be the highest rate in the system. Surprisingly, this is not always the case, meaning that some people have paid a price premium on the secondary market.

Figure 4

Weighted DAI Collateralized Debt Market Rates and MakerDAO Stability Fee.



Data sources: Loanscan.

As of September 2019, DAI accounts for almost 90% of all loans in the DeFi ecosystem.

2.4 Decentralized Derivatives

Decentralized derivatives are tokens that derive their value from the performance of an underlying asset, the outcome of an event or the development of any other observable variable. They usually require an oracle to track these variables and therefore introduce some dependencies and centralized components. The dependencies can be reduced when the derivative contract uses multiple independent data sources.

We differentiate between asset-based and event-based derivative tokens. We call a derivative token asset-based when its price is a function of the performance of an underlying asset. We call a derivative event-based when its price is a function of any observable variable that is not the performance of an asset. Both categories will be discussed in the following sections.

Asset-Based Derivative Tokens

Asset-based derivative tokens are an extension of the CDP model described in Section 2.3. Instead of limiting the issuance to USD-pegged stablecoins, the locked collateral can be used to issue synthetic tokens that follow the price movements of a variety of assets. Examples include tokenized versions of stocks, precious metals and alternative cryptoassets. The higher the volatility of the underlying, the larger the risk of falling below a given collateralization ratio.

A popular derivative token platform is called Synthetix (Brooks et. al., 2018). It is implemented in such a way that the total debt pool of all participants increases or decreases depending on the aggregate price of all outstanding synthetic assets. This ensures that tokens with the same underlying remain fungible, i.e. redemption does not depend on the issuer. The flip side of this design is that users assume additional risk

when they mint assets, as their debt position will also be affected by everyone else's asset allocation. Surprisingly, Synthetix has no liquidation mechanism.

A special case of asset-based derivative tokens are inverse tokens. Here, the price is determined by an inverse function of the underlying's performance within a given price range. These inverse tokens allow users to get short exposure to cryptoassets.

Event-Based Derivative Tokens

Event-based derivative tokens can be based on any objectively observable variable with a known set of potential outcomes, a specified observation time and resolution source. Anyone can buy a full set of sub-tokens for a given event by locking 1 ETH in a smart contract. A full set of sub-tokens consists of one sub-token for each potential outcome. These sub-tokens can be traded individually. When the market resolves, the smart contract's cryptoassets will be split among the sub-token owners of the winning outcome. In the absence of market distortions, the ETH price of each sub-token should therefore correspond to the probability of the underlying outcome.

Under certain circumstances, these prediction markets may serve as decentralized oracles for the likelihood of a future outcome. However, market resolution (and therefore the price) greatly depends on the trustworthiness of the resolution source. As such, event-based derivative tokens introduce external dependencies and may be unilaterally influenced by a malicious reporter. Potential attack vectors include bad or misleading question specification, incomplete outcome sets that may render the event unresolvable and the choice of unreliable or fraudulent resolution sources.

The most popular implementation is called Augur (Peterson et. al, 2019). It uses a multi-stage resolution and disputing process that should minimize the dependency on a single reporting source as much as possible. If the community does not agree with the designated reporter, they may start a dispute, that should eventually lead to the true outcome.

2.5 On-Chain Asset Management

Just like traditional investment funds, on-chain funds are mainly used for portfolio diversification. They allow users to invest in a basket of cryptoassets and employ a variety of strategies without having to handle the tokens individually. In contrast to traditional funds, the on-chain variant does not require a custodian. Instead, the cryptoassets are locked up in a smart contract. The investors never lose control over their funds, can withdraw or liquidate them and are able to observe the smart contracts' token balances at any point in time.

The smart contracts are set up in such a way that they follow a variety of simple strategies, including semi-automatic rebalancing of portfolio weights and trend trading using moving averages. Alternatively, one or multiple fund managers can be selected to actively manage the fund. In this case the smart contract ensures that the asset managers adhere to the predefined strategy and act in the best interest of the investors. In particular, the asset managers are limited to actions in accordance with the fund's rule set and the risk profile stipulated in the smart contract. As such, the smart contract can

mitigate many forms of the principle agent problem and may incorporate regulatory requirements by enforcing them on-chain.

Whenever someone invests in an on-chain fund, the corresponding smart contract issues *fund tokens* and transfers them to the investor's account. These tokens represent partial ownership of the fund and allow token holders to redeem or liquidate their share of the assets. For example, if an investor owns 1% of the fund tokens, this person would be entitled to 1% of the locked cryptoassets. When the investor decides to close out the investment, the fund tokens get burned, the underlying assets sold on a decentralized exchange and the investor compensated with the ETH-equivalent to his or her share of the basket.

There are several implementations of on-chain fund protocols, including the *SetProtocol* (Feng and Weickmann, 2019), *Melon* (Trinkler and El Isa, 2017) and *BeToken* (Liu and Palayer, 2018). All three implementations are limited to ERC-20 tokens and Ether. Moreover, they heavily depend on price oracles and third-party protocols, mainly for lending, trading and the inclusion of low-volatility reference assets such as the DAI or USDC stablecoins. Consequently, there are severe dependencies, which will be discussed in section 3.2.

Both Melon and the SetProtocol allow anyone to create new investment funds. Generally speaking, Melon has a focus on active management, offering a ruleset which ensures that fund managers stick to the funds' strategies. Trading restriction parameters such as maximum concentration, price tolerance and the maximum number of positions as well as user and asset white/black list, are enforced by the smart contracts. The same is true for the fund's fee schedule. The SetProtocol is mainly designed for simple semi-automated strategies with deterministic portfolio rebalancing, triggered by predefined threshold values and timelocks. BeToken operates as a single fund of funds, which is managed by a community of asset managers through a meritocratic system. The more successful an individual fund manager is, the higher their future influence on the allocation of the fund's resources. UniSwap's liquidity pool (see section 2.2) also has some characteristics of an on-chain investment fund. The constant product model creates the incentives for semi-automatic rebalancing of portfolio weights while the trading fees generate a passive income for the investors.

3 OPPORTUNITIES & RISKS

In this section, we analyze the opportunities and risks of the DeFi ecosystem. It lays the foundation for the discussion in section 4.

3.1 Opportunities

DeFi may increase the *efficiency*, *transparency* and *accessibility* of the financial infrastructure. Moreover, the *interoperability* of the system allows anyone to combine multiple applications and protocols and thereby create new and interesting services. We discuss these aspects in the following subsections.

Efficiency

While much of the traditional financial system is trust based, and therefore dependent on centralized institutions, DeFi replaces some of these trust requirements with smart contracts. The contracts can assume the roles of custodians, escrow agents and central counterparty clearing houses (CCP). For example, if two parties want to exchange digital assets in the form of tokens, there is no need for guarantees from a CCP. Instead, the two transactions can be settled atomically, meaning that either both or none of the transfers will be executed. This significantly decreases the counterparty credit risk and makes financial transactions much more efficient. Lower trust requirements may come with the additional benefit of decreasing the regulatory pressure and reducing the need for third party audits. Similar efficiency gains are possible for almost every area of the financial infrastructure.

Additionally, token transfers are much faster than any of the transfers in the traditional financial system. This can be further increased when we consider Blockchain second layer solutions, such as sidechains or state- and payment-channel networks.

Transparency

DeFi applications are completely transparent. All transactions are publicly observable and the smart contract code can be analyzed on-chain. The observability and deterministic execution allow – at least in theory – an unprecedented level of transparency.

Financial data is publicly available and may potentially be used by researchers and users alike. In the case of a crisis, the availability of historic (and current) data is a vast improvement over traditional financial systems, where much of the data is scattered across a large number of proprietary databases or not available at all. As such, the transparency may allow for the mitigation of undesirable events before they arise and help to understand their origin and potential consequences much faster when they emerge.

Accessibility

By default, DeFi protocols can be used by anyone. As such, DeFi may potentially create a truly open and accessible financial system. In particular, the infrastructure requirements are relatively low and the risk of discrimination is almost inexistent due to the lack of identities.

If regulation demands access restrictions, e.g. for security tokens, this can be implemented in the token contracts without compromising the integrity and decentralized properties of the settlement layer.

Interoperability

DeFi protocols are often compared to Lego pieces. The shared settlement layer allows these protocols and applications to interconnect. On-chain fund protocols can make use of decentralized exchange protocols or achieve leveraged positions through lending protocols.

Any two or more pieces can be rehashed to create something entirely new. In particular, anything that has been created before can either be used by a user or by other smart contracts. This leads to an ever-expanding range of possibilities and an unprecedented interest in open financial engineering.

3.2 Risks

DeFi also has certain risks, namely the *smart contract execution* risk, *operational security* and the *dependencies* on other protocols and external data. We discuss these aspects in the following subsections.

Smart Contract Execution

While the deterministic and decentralized execution of smart contracts does have its advantages, there is a risk that something may go wrong. If there are coding errors, these errors may potentially create vulnerabilities that may allow an attacker to drain the smart contract's funds, cause chaos or render the protocol unusable. Users have to be aware that the protocol is only as secure as the smart contracts underlying it. Unfortunately, the average user will not be able to read the contract code, let alone evaluate its security. While audits, insurance services and formal verification are partial solutions to this problem, some degree of uncertainty remains.

Similar risks exist on contract execution. Most users do not understand the data payload they are asked to sign as part of the transactions and may be misled by a compromised front-end.

Operational Security (OpSec)

Many DeFi protocols and applications use admin keys. These keys allow a predefined group of individuals (usually the project's core team) to upgrade the contracts and to perform emergency shutdowns. While it is understandable that some projects want to implement these precautionary measures, the existence of these keys can be a potential problem. If the keyholders do not create and/or store their keys in a secure way, malicious third parties could get their hands on these keys and compromise the smart contract.

Most projects try to mitigate this risk with multisig and timelocks. Multisig requires M-of-N keys to execute any of the smart contract's admin functions and timelocks introduce time delays that could be used to respond accordingly.

Dependencies

As described in Section 3.1, some of the most promising features of the DeFi ecosystem are its openness and interoperability. This allows various smart contracts and decentralized blockchain applications to interact with each other and to offer new services based on a combination of the existing ones. On the flip side, these interactions could introduce severe dependencies. If there is an issue with one smart contract, this may potentially have wide-reaching consequences for multiple applications across the

entire DeFi ecosystem. Moreover, issues with the DAI stablecoin or severe ETH price shocks may cause ripple effects that affect the entire DeFi ecosystem.

The problem becomes apparent when illustrated by an example. Let us assume that a person locks ETH as collateral in the MakerDAO contract to issue DAI stablecoins. Let us further assume, that the DAI stablecoins are then locked in the compound lending smart contract to issue interest-bearing cDAI derivative tokens. The cDai tokens are subsequently moved to the UniSwap ETH/cDAI liquidity pool,[8] allowing the person to withdraw UNI-cDAI tokens that represent a share of the liquidity pool. With every additional smart contract, the potential risk of a bug increases. If any of the contracts in the sequence fail, the UNI-cDAI tokens could potentially become worthless. These "token on top of a token on top of a token" scenarios can entangle the various projects in such a way that the theoretical transparency does not correspond to actual transparency.

Another point worth mentioning is the fact that many smart contracts are reliant on external data. Whenever a smart contract depends on data that is not natively available on-chain, this data must be provided by external data sources. These oracles introduce dependencies and may, in some cases, lead to heavily centralized contract execution. To mitigate this risk, many projects rely on large oracle networks with M-of-N data provision schemes.

4 CONCLUSION

Decentralized Finance offers exciting opportunities and has the potential to create a truly open, transparent and immutable financial infrastructure. Consisting of numerous highly interoperable protocols and applications, all transactions can be verified by every individual and data is readily available for users and researchers to analyze.

DeFi has unleashed a wave of innovation. On the one hand, developers are using smart contracts and the decentralized settlement layer to create trustless versions of traditional financial instruments. On the other hand, they create entirely new financial instruments that could not be realized without the underlying public Blockchain. *Atomic swaps, autonomous liquidity pools, decentralized stablecoins* and *flash loans* are just a few of many examples that show the great potential of this ecosystem.

While this technology has great potential, there are certain risks involved. Smart contracts can have security issues that may allow for unintended usage. Moreover, the term "decentralized" is deceptive in some cases. Many of the protocols and applications use external data sources and special admin keys to manage the system, conduct smart contract upgrades or even perform emergency shutdowns. While this does not necessarily constitute a problem, users should be aware that, in many cases, there is a lot of trust involved. If, however, these issues can be solved, DeFi may lead to a paradigm shift in the financial industry and potentially contribute towards more robust and transparent financial infrastructure.

NOTES

[1] An alternative approach can be found here: https://medium.com/pov-crypto/ethereum-the-digital-finance-stack-4ba988c6c14b

[2] For readers who wish to first get a better understanding the settlement layer and want to read a general introduction to Blockchain and cryptocurrencies, we recommend Berentsen and Schär (2018).

[3] Source: etherscan.io/tokens, accessed September 15th 2019.

[4] UTXO-based Blockchain implementations such as Bitcoin allow sophisticated unlocking conditions through their scripting language. Although most people would not call these locking scripts a smart contract, they achieve similar goals in terms of the custodial capabilities of the Blockchain.

[5] Source: coinmarketcap.com, accessed September 15th 2019.

[6] In practice the collateralization must be much larger, as any credit position with a collateralization below 150% are liquidated.

[7] Information retrieved from https://docs.bzx.network/

[8] Alongside some ETH.

REFERENCES

Adams, Hayden. "UniSwap." 2018. https://hackmd.io/@Uniswap/HJ9jLsfTz

Berentsen, Aleksander and Schär, Fabian. "A Short Introduction to the World of Cryptocurrencies" 2018. Vol. 100, Issue 1, pp. 1-16, 2018.

Berentsen, Aleksander and Schär, Fabian. "Stablecoins: The Quest for a Low-Volatility Cryptocurrency." 2019. In: The Economics of Fintech and Digital Currencies. London, pp. 65-71.

Boado, Ernesto. "Aave Protocol Whitepaper v1.0." 2020. https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf

Brooks, Samuel, Jurisevic, Anton, Spain, Michael and Wawrick, Kain. "Havven: a decentralised payment network and stablecoin." 2018. https://www.synthetix.io/uploads/havven_whitepaper.pdf

Buterin, Vitalik. "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform." 2013. http://ethereum.org/ethereum.html.

Entrinken, William, Shirley, Dieter, Evans, Jacob and Sachs, Nastassia. "ERC-721 Non-Fungible Token Standard." 2018. https://github.com/ethereum/EIPs/blob/master/EIPS/eip-721.md

Feng, Felix and Weickmann, Brian. "Set: A Protocol for Baskets of Tokenized Assets (v1.2)." 2019. https://www.setprotocol.com/pdf/set_protocol_whitepaper.pdf

Hertzog, Eyal, Benartzi, Guy and Benartzi, Galia. "Bancor protocol - continuous liquidity and asynchronous price discovery for tokens through their smart contracts; aka *smart tokens*." 2017. https://whitepaper.io/document/52/bancor-whitepaper

Juliano, Antonio. "dYdX: A Standard for Decentralized Margin Trading and Derivatives." 2017. https://whitepaper.dydx.exchange/

Leshner, Robert and Hayes, Geoffrey. "Compound: The Money Market Protocol." 2019. https://compound.finance/documents/Compound.Whitepaper.pdf

Liu, Zebang and Palayer, Guillaume. "Betoken: A Meritocratic Hedge Fund Built on Ethereum." 2018. https://github.com/Betoken/Whitepaper/blob/master/BetokenWhitepaper.pdf

Luu, Loi and Velner, Yaron. "KyberNetwork: A trustless decentralized exchange and payment service." 2017. https://whitepaper.io/document/43/kyber-network-whitepaper

Oved, Michael and Mosites, Don. "Swap: A Peer-to-Peer Protocol for Trading Ethereum Tokens." 2017. https://swap.tech/pdfs/SwapWhitepaper.pdf

Peterson, Jack, Krug, Joseph, Zoltu, Micah, Williams, Austin K. and Alexander, Stephanie. "Augur: A Decentralized Oracle and Prediction Market Platform (v2.0)." 2019. https://www.augur.net/whitepaper.pdf

Roth, Jakob and Schär, Fabian and Schöpfer, Aljoscha. "The Tokenization of Assets: Using Blockchains for Equity Crowdfunding." 2019. http://dx.doi.org/10.2139/ssrn.3443382

Szabo, Nick. "Smart Contracts." 1994. http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

Szabo, Nick. "The idea of smart contracts." 1997. https://nakamotoinstitute.org/the-idea-of-smart-contracts/

Trinkler, Reto and El Isa, Mona. "Melon protocol: a blockchain protocol for digital asset management." 2017. https://github.com/melonproject/paper/blob/master/melonprotocol.pdf

Vogelsteller, Fabian and Buterin, Vitalik. "ERC-20 Token Standard." 2015. https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md

Warren, Will and Bandeali, Amir. "0x: An open protocol for decentralized exchange on the Ethereum blockchain." 2017. https://0x.org/pdfs/0x_white_paper.pdf

Wolff, Max. "Introducing Marble: A Smart Contract Bank". 2018. https://medium.com/marbleorg/introducing-marble-a-smart-contract-bank-c9c438a12890

Wood, Gavin. "Ethereum: A Secure Decentralised Generalised Transaction Ledger." 2015. https://ethereum.github.io/yellowpaper/paper.pdf.

Zhang, Yi, Chen, Xiaohong and Park, Daejun. "Formal specification of constant product (x times y = k) market maker model and implementation." 2018. https://github.com/runtimeverification/verified-smart-contracts/blob/uniswap/uniswap/x-y-k.pdf

ACKNOWLEDGEMENTS

The author would like to thank Florian Bitterli and Raphael Knechtli for their support with data collection and visualization, as well as Emma Littlejohn for proof-reading the manuscript.