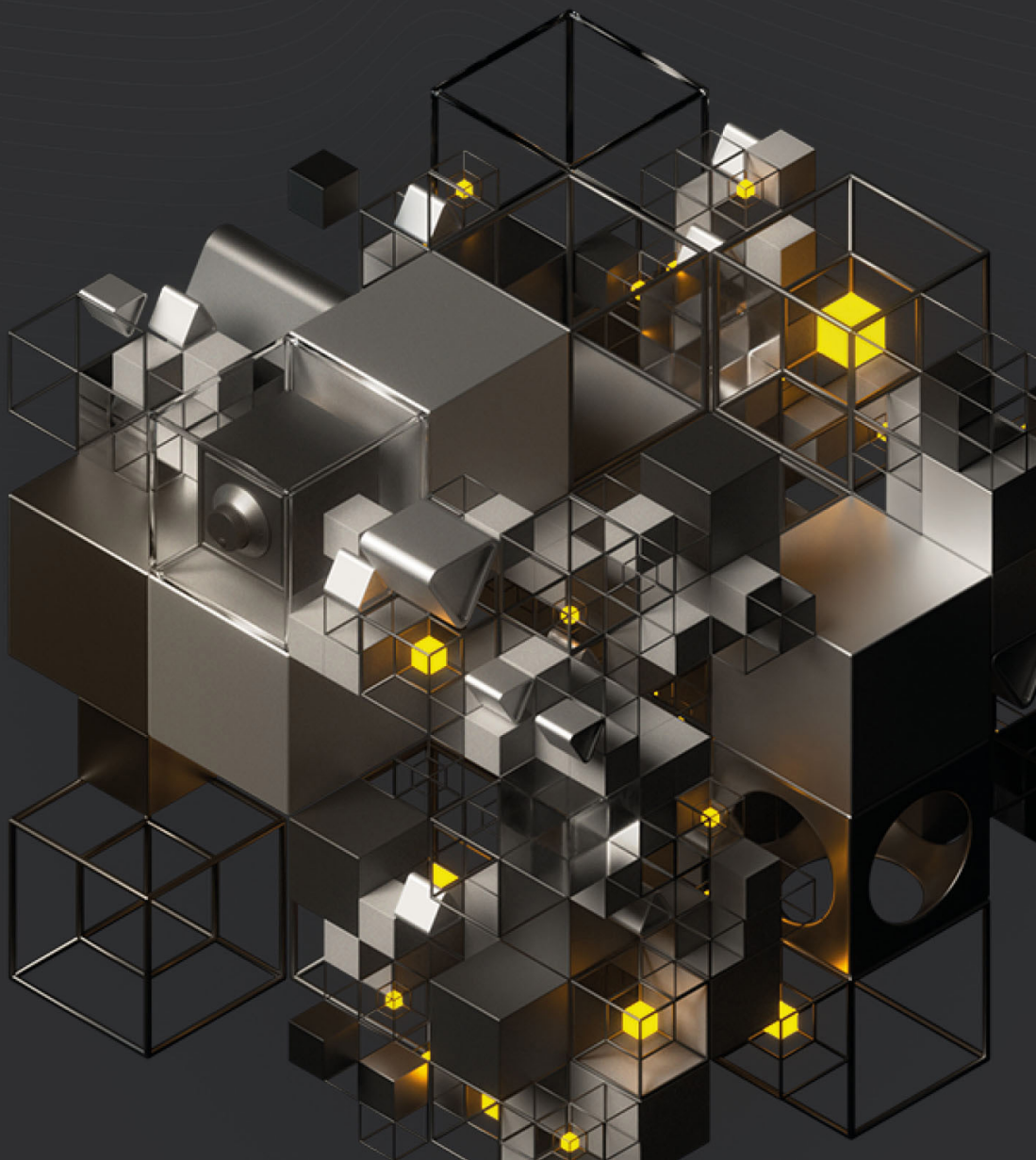
Redefine 2020: A Primer

Where decentralization meets finance*



* There be dragons

About

Lead authors:

Alexander Bokhenek | Byzantine Solutions | ab@byzantine.solutions

Ivan Kamakin | Byzantine Solutions | ik@byzantine.solutions

Demelza Hays | Cointelegraph Consulting | demelza@cointelegraph.com

Contributors:

Alexandra Vachnadze | Byzantine Solutions | av@byzantine.solutions

Andrew Durgee | Republic Advisory Services | andrew@republic.co

Aleksandra Nikiforova | Byzantine Solutions | an@byzantine.solutions

Jason Choi | Cointelegraph Consulting | jason.choi@cointelegraph.com

Graham Friedman | Republic Advisory Services | graham@republic.co

Arsenii Dain | Cointelegraph Consulting | arsenii.dain@cointelegraph.com

Ermin Sharich | Cointelegraph Consulting | e.sharich@cointelegraph.com

September 2020

COINTELEGRAPH
Consulting

Cointelegraph Consulting offers bespoke research on digital assets and distributed ledger technology. Our services range from phone calls with clients when they have a question, educational seminars for companies via online conferencing, and in-depth written reports on a wide range of topics. Our team consists of management consultants, professional researchers and seasoned blockchain technologists that have a passion for providing unbiased buy-side research.

BYZANTINE SOLUTIONS

Byzantine Solutions is a fintech innovation lab specialized in distributed ledger technology and blockchains, decentralized finance, cryptographic applications, tokenomics research, and mechanism design. The company offers R&D services, product ideation and architecture design, technical due diligence, and tokenomics development.

Abstract

Emergence of distributed ledger technologies (DLT) and smart contracts, and their popularization in the second half of 2010s is spearheading technological shifts in a number of fields. Most notable current applications lay in fintech, digital assets and financial instruments. The concept of automatic, guaranteed execution of code that is able to manage value and is resistant to modifications by any one party turned out to be an immensely powerful driver of thought in economics, game theory, mechanism design, and business, also bringing forward rapid advances in cryptography.

Perhaps the most notable outcome of DLT to date is the emergence of complex financial instruments that do not generally rely on any central party—be it a company or a government—to neither operate, nor merely regulate itself. Decentralized finance (or DeFi) is equipped to withstand societal, economical, and political shocks, by running immutable code incentivizing market forces on decentralized networks.

The present report sets out to explain in depth what that means, how that works, what is the landscape of the industry, and what makes it, in our view, a promising, if not groundbreaking, direction of thought and economic interaction.

Table of Contents

Disclaimer

This publication is for information purposes only and represents neither investment advice, nor an investment analysis or an invitation to buy or sell financial instruments. The statements contained in this publication are based on the knowledge as of the time of preparation and are subject to change at any time without further notice. The authors have exercised the greatest possible care in the selection of the information sources employed, however, they do not accept any responsibility (and neither does Cointelegraph) for the correctness, completeness, or timeliness of the information, respectively the information sources made available, as well as any liabilities or damages, irrespective of their nature, that may result there from (including consequential or indirect damages, loss of prospective profits or the accuracy of prepared forecasts).

1 General Introduction

1.1 Foreword

Choosing accurate terminology around DLT and DeFi is in itself a non-trivial matter. The industry is still very young and not sufficiently studied; concepts evolve every year, and the terms setting them apart come and go. Bitcoin, for instance, was initially described as a "peer-to-peer electronic cash system"[2], defining its contribution as a solution to the double-spending problem that allowed to remove a trusted third-party from the duty of keeping the ledger ('record'). The terms 'trustless', 'permissionless' and 'decentralized' (to name a few), commonly used now to convey the fundamental characteristics of Bitcoin, were only coined later, and on separate occasions.

In general terms, we informally define decentralized finance as an ecosystem of financial instruments[3] that do not rely on any particular third party for its governance and/or operation, nor where any particular third party has the power to interfere with the usage of the service by any user. The team or the company that has actually produced the instrument is also considered a third party that shouldn't have privileged access, so in this sense, most projects in the industry (and in the present report) are only DeFi in the making, having at the time strong reliance on central agents, with roadmaps to eliminate it in the coming years.

The approach to eliminate centralization lies in combining immutable core logic (often embodied in smart contracts) with economic incentives for rational market agents. Every component of a particular instrument, or activity required for its operation, is replaced with financially motivated actions of arbitrary agents in market competition. Coincidentally, this usually entails running on or in conjunction with a public permissionless blockchain capable of executing smart contract logic, as otherwise an ability of the system's maintainer to interfere with the transactional or custodial medium would contaminate the instrument.

An edge case is committee-based components that take actions as they are agreed upon by a pre-defined group of participants (usually between 5 and 30). Since the committee in itself is fixed and can be targeted, if not as a legal entity, than as a small group of individuals, it does not completely fill the role of a decentralized component. Meanwhile, if the function of the committee is well defined, and no decentralized mechanisms are known to fill that role and guarantee the required combination of properties, a committee may be a reasonable medium-term solution. The most frequent component implemented with committees is price feeds (called "price oracles").

Other grey areas, in this view, are about compromises that pitch the permissionless structure against the ability to operate and iteratively upgrade, and the social consensus on whether a particular practice or concession is justified for its gain. It usually boils down to four topics.

Emergency management. No code is perfect, even after rigorous testing, and when it handles financial assets, the cost of failures can be almost arbitrary, given that unwinding of history is almost always impossible for blockchain networks. Some form of failsafes is often installed, giving their administrators excessive power. Failsafes usually act on the entire instrument rather than particular transactions or assets, and are perceived as such, with both factors somewhat mitigating the effect of centralization.

Upgradability and development cycles. In a similar manner, no code is ever feature-perfect or ideally optimized. Development teams work on products after their launch, and with smart contracts, upgrades often involve re-deploying the new version alongside (e.g. Uniswap) or instead of (e.g. Maker) the operational one. Upgradability can be done in several ways. One of them entails the ability to switch the code managing assets in the system, another is to just deploy new code and then use an emergency switch to make the 'obsolete' system unusable. The access to either of these instruments can be held by an individual, a committee, or a decentralized governance system, representing different positions on the gauge between upgradability and centralization.

Underlying technology. The chain is only as strong as its single point of failure. A project with great decentralization mechanisms

[2] The Bitcoin original paper (Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System, link)

[3] By financial instruments, among other things, we mean digital currencies, trading, lending, prediction markets, and derivative instruments.

cannot really benefit from them if it runs on a permissioned blockchain, or only uses assets that can be arbitrarily revoked or frozen by their emitents.

Legal compliance. The topic of legality and jurisdictions is a hot one in the DLT crowd. On the one hand, the general notion of building accessible, democratic, trustless, and incorruptible instruments is often perceived to go against actively seeking legal compliance and selective transparency to the regulatory bodies. On the other hand, risk of legal prosecution and the desire to seek venture funding set two bars for compliance that are often sought by projects. An extreme example of the latter is the "corporate" archetype of projects that ensure compliance with prejudice, personally identify every customer, introduce anti money-laundering (AML) systems before developing a product, and then sometimes claim decentralization "on top" of that layer that is viewed as baseline. One issue with their approach is that such checks are nigh impossible to introduce without creating centralization bottlenecks, i.e. concrete mechanisms for disabling user accounts, freezing funds, etc., by a single API call by the admin entity. These mechanisms generally revert the overall level of censorship resistance back to that of pre-DLT financial services.

With all that in mind, the present report is limited to instruments that target decentralization as an inherent goal, and only consider having centralization bottlenecks as stepping stones on a reasonably defined path to full decentralization. Within our definition, a project does not fall under the category of DeFi, if (barring programming errors) any entity is able in practice to freeze or revoke assets of any of its users. Similarly, with one notable exception, no entity should be able to deny the user access to the financial service. Therefore we completely exclude projects such as custodially backed stablecoins and centralized exchanges, among some others. The one exception we make is non-custodial layer-2 exchanges utilizing zero-knowledge proofs, as we believe that in this one particular case such power is a temporary technological limitation on the otherwise clear path of technological innovation towards fully decentralized solutions.

From the reader we assume basic familiarity with blockchains and smart contracts to the extent of knowing what they are and what are their most basic properties[4]. In sections on decentralized counterparts for traditional financial instruments, we make passing references to concepts from finance, but essential background about the instruments is usually provided in boxouts nearby.

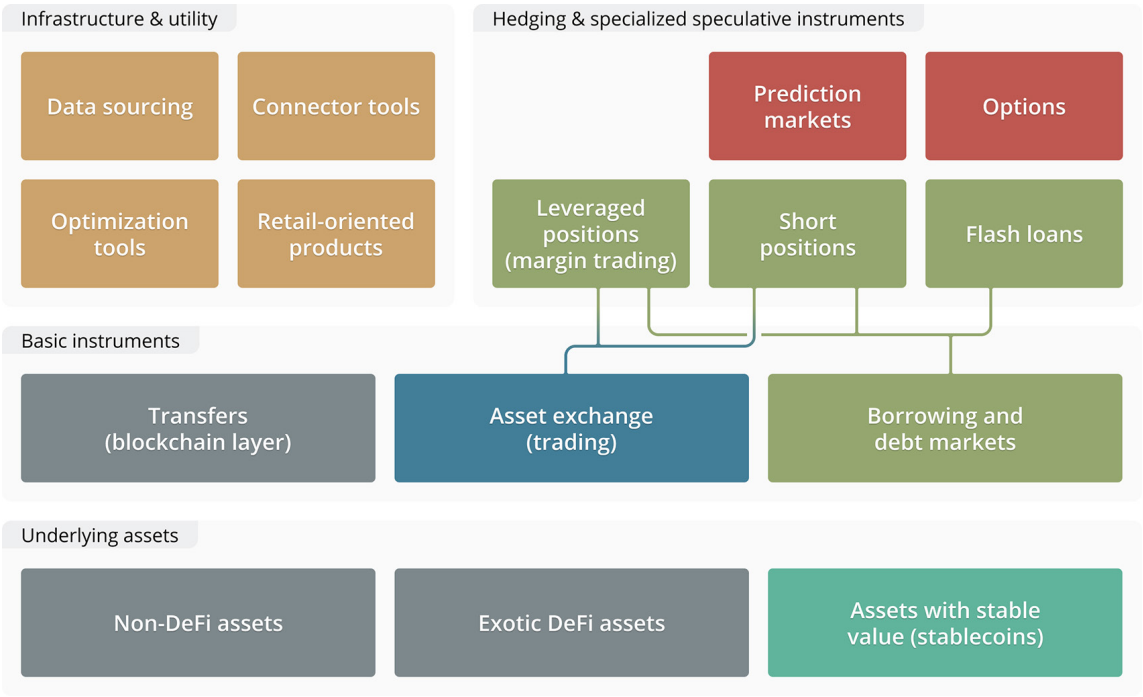
[4] Transactions, blocks, cryptographic signatures, self-custody of digital assets, immutability of history, permissionless transactions, transparency of code and history, tamper-proof execution of smart contract code.

1.2 DeFi at a glance

The base layer of financial services captured by DeFi can be classified functionally, by the action an end-user performs on a unit of value. Individuals and smaller enterprises not specialized in financial markets first and foremost need access to transfers, exchanges, and loans. More complex functionality includes instruments for hedging and advanced speculation: short positions, leveraged positions, prediction markets, flash loans, options, etc. A separate section is made up from infrastructure and utility tools mostly specific to the industry: connections and wrappers over DeFi protocols (both technical and customer-facing), and instruments for data provision.

The thought process of decentralized finance dates back to at least 2016—when several products were proposed in the community—or even 2014—when Ethereum redefined the scope of possibility with regards to decentralized tamper-proof computation. In embodied form, we view DeFi as hitting ground in late 2017, particularly with the launch of **Maker.DAO**—a non-custodial algorithmically stabilized currency pegged to U.S. Dollar that has, in general, successfully maintained the peg ever since, with limited exposure to central-party influence (more on that later). The general idea was to produce a permissionless credit system, the ability to borrow money after putting up excessive collateral. Naturally, the borrowed currency needed to be stable, and the collateralization of the entire monetary mass needed to support it. 2017 also witnessed the launch of **Bancor** protocol that pioneered in public view the concept of an exchange engine without an order book, running solely on smart contracts in an efficient manner. Go-to protocols for lending (**Compound**) and prediction markets (**Augur**) were launched on Ethereum mainnet in 2018.

In the sections below we will roughly follow the functional breakdown, reviewing ideas and mechanisms that have been considered by live projects to provide the functionality in a permissionless manner, with a design goal to avoid introducing potential single points of failure. The approach entails replacement of the centralized mechanisms for enforcement of desired behaviour and provision of required inputs with programmable rules, algorithmically adjusted incentives, and actions of rational profit-seeking market agents. The class of problems and applicable machinery varies by product group, with several patterns applicable universally and therefore arising in multiple niches.



1.3 Structure of the report

The report has three major content sections, presenting the ecosystem from the ground up: from transfers and assets, to individual financial instruments, to synergies between instruments, to emergent properties and system-wide effects. The general focus is on mechanisms and projects rather than utilitarian value, so some of the instruments (such as margin trading) are covered in the sections corresponding to their underlying DeFi abstraction (in this case, loans) rather than function (in this case, trading).

1 **General introduction.** You are here.

2 **Underlying assets.** The section covers assets that utilize DeFi mechanisms in their design to achieve specific properties. While the financial instruments from the next sections are not limited to using only DeFi assets, we assume that the economic properties inherent to other kinds of assets usable on blockchains are covered by other sources. Of the project groups, it covers stablecoins.

3 **Instruments.** The financial instruments replicated and/or built in decentralized finance are presented in this section. It forms the bulk of the report and illustrates most of the concepts and mechanisms utilized in the industry. It is broken down into parts: trading, borrowing and debt markets (including loan-based trading instruments), risk management and hedging (including options and prediction markets), and infrastructure & utility tools.

4 **Properties of DeFi.** In this section we conceptualize the core underlying properties that are universally applicable to decentralized finance in contrast with traditional finance. We also explore the important interactions between instruments, from the standpoints of both potential synergies and emergent effects, and systemic risks.

5 **Conclusion.** Provides an extended summary of the report and closing thoughts on the possible meeting points between traditional and decentralized finance.

2

Underlying assets

OPENING STATEMENT

Permissionless transfer of value

The cornerstone of decentralized money is the ability to send value without needing permission. It was the selling point of **Bitcoin**, along with money supply with a guaranteed asymptotic upper bound and diminishing emission rate, introducing limits to dilution of value. The engineering of it was brilliant. Four requirements enable this solution:

1 A permissionless network needs a way to maintain open entry and exit for stakeholders. A large limited group is impractical to coordinate, while a small one would be prone to collusion or sequential corruption by a third party, leading to the loss of permissionlessness.

2 The history of the network (namely, movement of assets between stakeholders) has to be unique, immutable, and independently verifiable from the ground-up, in a tamper-proof way. Otherwise an entering stakeholder would depend on their contacts, again, losing permissionlessness. An ability to revert history would also enable parties to spend a same asset more than once, reverting history between iterations, while their purchases made off-chain could not be reverted. Double-spending renders a payment system useless.

3 Updating history should be available to anyone, as long as the core rules are upheld by them (and the previous property is maintained). Limiting the set of those able to update could lead to censorship.

4 The system should provide strong incentives to participate in updating the network. If it doesn't, there is a tragedy of commons: no one would be willing to bear the costs of providing a public good by updating the network, so it would stop (or boil down to just a small set of stakeholders, again, enabling collusion or sequential corruption).

To meet these requirements, Bitcoin brings forward three defining mechanisms:

1 History as a chain of blocks with cryptographic links covers immutability and verifiability of history.

2 Mining difficulty and the consensus over the chain with the most work spent (to be referenced as Nakamoto consensus since) provide open entry and exit for stakeholders and uniqueness of history.

3 Coinbase rewards for producing new blocks eliminate the tragedy of commons with regards to updates[5].

While an important advancement, Bitcoin is not suited to cover the entire range of what one might want of a financial system. Simply replacing wire transfers with Bitcoin transactions while leaving the other financial machinery as-is does not really work. Rather, the question is, can the key properties of Bitcoin be broadened for other types of interactions. The next step taken by the industry is generalization of value transactions (transfers of value, under some basic conditions afforded by Bitcoin script) to execution of arbitrary asset managing code[6] within a similar set of requirements. It was first envisioned and implemented in **Ethereum**—which to date remains the central platform for deployment of DeFi,—followed by other projects focusing on different constraints of Ethereum network.

---

[5] There is a separate concern about keeping history, as a miner does not actually need to remember every block, just the current state of the ledger. Bitcoin leaves this one unsolved, falling back to self-interest of those heavily invested in the Bitcoin ecosystem to keep the network alive and decentralized,—which is a weaker construction. It has, nonetheless, never been endangered in practice.

[6] The halting problem (the fact that an arbitrary program might not terminate in finite time, and there is no way to check whether it would) is usually sidestepped by limiting resource allocation for processing any particular execution, and taxing the user to cover the costs of that processing. This is a step forward from Satoshi's approach to just limit the programming capabilities.

2.1 Stablecoins

The term "stablecoin" is traditionally used in the context of cryptocurrencies to represent a token with stable value (as opposed to a volatile token), defined in terms of a particular non-blockchain asset, most often a fiat currency. The stablecoin is said to be pegged to the target asset, meaning that one unit of that stablecoin is always worth one unit of that asset. Within a decentralized setting, there is a further requirement that the mechanisms guaranteeing the peg should also be decentralized. Stablecoins are the DeFi equivalent of money: while tokens of projects and protocols can be used to facilitate payments, their volatility overcomplicates pretty much every use case to be of practical value for complex applications.

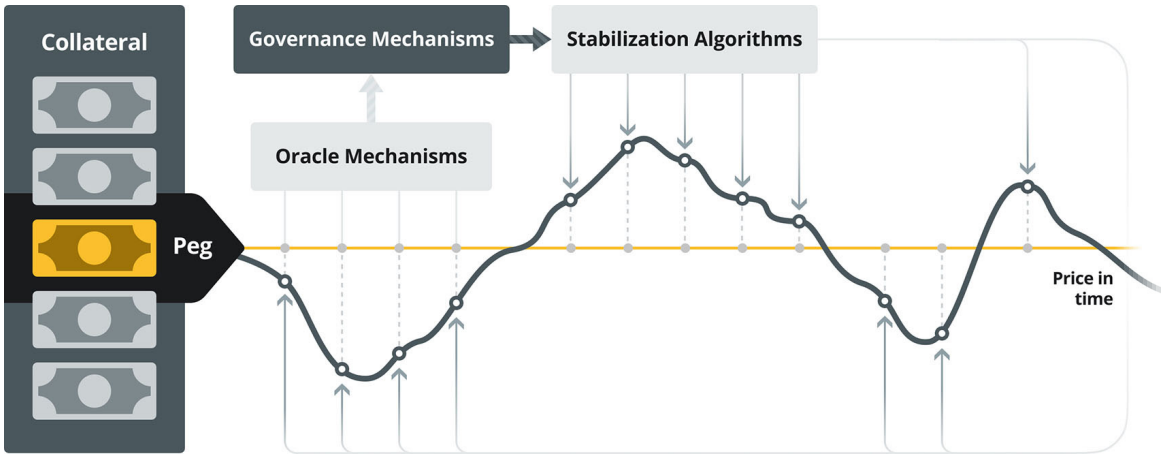
Stablecoins are an asset class arguably unique to DeFi. While they seemingly represent stable (i.e. dollar) values owned by blockchain accounts, they bear no similarity with accounts with payment providers and other money institutions. The usual digital money is in essence an IOU from the respective institution to send the corresponding amount of a fiat currency on withdrawal request. This bears two caveats for a decentralized setting, as there are centralization bottlenecks:

1. Even if we set aside the ability of the company to freeze assets on (even false positive) suspicions of money laundering or other illegal activity, there are also risks of pooling together money and personal information within the organization, lighting it up as a target for hackers. Stablecoins approach that by moving the IOU part in its entirety to on-chain entities (permissionless smart contracts), often using the pattern of overcollateralization and market liquidation to guarantee solvency[7].
2. Regulation of monetary mass and buying power is maintained for fiat money by sovereign states that create them. Both these properties and, in case of money in the banking system, even the day-to-day operation, is controlled by one decision-making body and its representatives.

In DeFi, such centralized entities have no place: considering how important the role of stablecoins is in the ecosystem, a centralized issuer would be able to exploit their position in numerous ways, likely to the detriment of the ecosystem as a whole. This is why stablecoins employ complex economic mechanisms with carefully tailored incentive structures to achieve stabilization without any centralized controller, but still resistant to economic attacks.

Stablecoins not only have an important practical purpose within the DeFi ecosystem, but also serve as a great source of inspiration, as they present some of the most creative applications of game theory and mechanism design in the modern world.

By far the most important characteristic when outlining the stablecoin landscape is the mechanism that ensures the peg. While all stablecoins vary in the details of how their stabilization works, their mechanisms can be categorized into three classes, excluding a few particularly exotic cases: collateralized debt models, seigniorage share models, and profit/loss consumption approach.



A stablecoin is supported with three key components: oracle mechanisms that feed the market prices into the system, governance mechanisms that adjust the system's parameters in response to market events, and stabilization algorithms that adjust incentives for market participants to affect the stablecoin price and push it towards the peg.

[7] As was mentioned in the introduction, for the purpose of the present report, we do not classify tokens custodially backed with fiat currencies as DeFi stablecoins.

2.1.1 Collateralized Debt Model

Stablecoins with a CDP (collateralized debt position) model work by collateralizing the pegged asset with another, more volatile asset. As the collateral asset is volatile, it risks depreciating to the point of not covering the issued stablecoins anymore. Due to this the stablecoins are overcollateralized—this creates a safety margin that helps liquidate stablecoins which are close to becoming undercollateralized.

The principle of the CDP model is best demonstrated through its initial and most well-known implementation—**Maker.DAO**. Another prominent project that evolves the concept further is **Synthetix**.

CASE STUDY

MAKER

Maker.DAO is the project that introduced the CDP model, as well as the first successful stablecoin project. DAI can be considered the "pure" form of a CDP stablecoin—all other stablecoin projects with a CDP model utilize the same base structure, only adding new functionality on top.

To create DAI, the user has to lock collateral into a CDP[8]. CDPs can be created by anyone, and only the owner of the CDP can borrow stablecoins from it, or redeem them to get their collateral back. However, when a CDP goes below a set collateralization ratio (150% in Maker), it must be liquidated to keep the system collateralized—to ensure that this happens quickly, the premium is given for liquidation, which any market agent can claim by acquiring the stablecoin and returning it into the CDP. In Maker ecosystem, these agents are called keepers: they keep the system solvent by liquidating positions dangerously close to insolvency. Keepers compete for liquidating CDPs, ensuring that liquidation is quick.

The stablecoins issued are considered debt to the protocol and accrue interest rate (often called "stability fee")—this revenue is distributed to platform stakeholders and insurance pools that are meant to keep the system stable even during black swan events.

> ***Example.*** *Alison has some spare ETH and wants to produce some stablecoins. Using Maker interface, she creates a CDP, placing $3,000 worth of ETH in collateral, and chooses to draw 800 DAI from it. The resulting collateralization ratio is 375%, way above the minimum of 150%. This means that Alison's position is very safe: for her CDP to be liquidated, the price of ETH needs to fall by 60% (so that the collateral value falls from $3,000 to $1,200), which is unlikely.*

Stability in Maker is achieved through arbitrage and adjustment of interest rates on borrowed stablecoins.

When the stablecoin depreciates relative to the peg, the market agents can free their collateral or liquidate positions for cheaper, thus earning a dollar premium. E.g., if the CDP owner exchanges their DAI to, e.g, NVM[9], and then DAI depreciates, they can buy DAI back and unlock the collateral, while retaining some leftover NVM. The additional measure to reduce supply and bring the prices back up is increasing the interest rate—this incentivizes CDP holders to close their positions early to avoid accruing higher fees.

Conversely, when the stablecoin appreciates, arbitrageurs can create new stablecoins (which are always minted at parity), sell them for a higher-than-parity price, and then unlock their collateral when the stablecoin returns to parity. The interest rates can also be decreased, to incentivize new users to create stable assets for a cheaper price.

[8] The term CDP was first used in production by Maker, but in Maker v.2 it was renamed as Vault. We keep using the term CDP as it is more functionally descriptive.

[9] An imaginary token Nevermore (NVM) that we will use as an example of an arbitrary volatile token. (Disclaimer: to the best of our knowledge at the time of writing, there are no known tokens with that ticker and/or project name.)

Maker grappled with a lot of issues over the course of its existence, prompting iterative updates of the protocol and its supporting infrastructure. The most recent example was a series of auction liquidations ending up with 100% discount that led to losses of up to $5.4 million across multiple CDP holders. It was possible because of a rare sequence of events related to the market uncertainty regarding the coronavirus, which led to a sharp 50% drop in ETH price combined with extreme network congestion in Ethereum. Many CDPs became undercollateralized, oracle price updates were delayed, and multiple 0-value bids were made on collateral and made it to resolution. Multiple keepers experienced liquidity shortages, being unable to cover every auction, nor to recycle their liquidity fast enough to cover many auctions in sequence. The keeper bot run by Maker Foundation experienced technical issues due to network congestion[10]. The system maintained the price stability of DAI by auctioning off a large supply of MKR tokens, which is a standard crisis recovery mechanism in the system. Several steps were taken and discussed to prevent similar events in the future, including increasing the length for collateral auctions.

Additionally, in some periods DAI experienced consistent higher than parity prices, even with near-zero stability fees. This prompted a lot of users to demand implementing negative interest rates, to allow the protocol to properly deal with periods of extremely high demand. This, however, is a controversial initiative which meets a lot of resistance from holders of MKR—the Maker governance token. MKR holders are the main beneficiaries of the stability fee, which means that they would have to pay for stability during increased demand. This demonstrates that even systems as widely used as Maker can suffer from non-obvious cases of incentive misalignment, which speaks to complexity of DeFi mechanism design.

Overall, despite issues that arise and are solved iteratively, DAI remains one the most well-used assets in DeFi, with new products being introduced on top of it. Just recently, Maker launched DAI Savings Rate, which not only provides a savings solution in DeFi, but also helps ensure stability.

CASE STUDY

SYNTHETIX

The protocol takes the CDP model introduced by Maker further, allowing CDP owners to issue not a single asset, but a multitude of synthetic assets from the same collateralized position. Synthetics follow the price movements of the assets they mirror (via oracle price feeds), but can be converted to other synthetics or burned at will, with zero slippage and no need of a counterparty.

Synthetix users start by staking the project's native volatile coin—SNX. Then, they are able to issue assets such as sUSD, sEUR, sETH or sBTC, representing exposure to various markets (including cryptocurrencies, ForEx currencies, some commodities and some equities). These assets track the value of original assets in USD, so to issue 1 sBTC, the SNX staker has to lock $BTCprice \times ColRatio / SNXprice$ of SNX.

Of particular interest is Synethetix's debt structure—instead of individual CDP owners having their own respective debt levels, there is the overall platform debt level. SNX stakers are not liable to a particular amount of asset—rather, they are liable to a percentage of the global debt pool. This means that the actual value of debt of a particular staker can change dynamically—it will increase if the issued synths appreciate in value, and will decrease when synths depreciate.

This means that debt in the platform is essentially socialized—the extra debt generated by sBTC appreciating (regardless of the issuer of those sBTC) will be allocated to all SNX stakers pro-rata their share of the debt pool.

[10] A good postmortem report on these events can be found on Maker.Dao blog.

> *Let there be two stakers in the system. One issues $1,000 sUSD, and the other $9,000 sUSD. Their respective shares of debt are 10% and 90%. Now suppose that the first staker converted their sUSD to sBTC and BTC appreciated x2 (the first staker now has $2,000 worth of synths). The total platform debt is now $11,000. The first staker being liable to $1,100 and the second—to $9,900, even though the entirety of new debt was, technically, generated by the first staker.*

As a corollary, SNX stakers have two distinct strategies: issuing stable synths (e.g., sUSD) or issuing volatile synths (e.g., sETH):

Issuing stable synths (and holding them, so that they are not converted to volatile assets) is a bet against the performance of holders of volatile synths (whether they issued those synths or bought them somewhere)—if volatile synths depreciate, stable synth issuers' debt decreases, while the value of their issued assets remains the same. They can repay their debt and retain extra stable synths to sell on the secondary market.

Issuing volatile synths is akin to trading underlying volatile assets with leverage less than 1. Part of the accrued debt from volatile synth appreciation will be allocated to stable synth issuers, while volatile synth issuers get 100% of value appreciation of an asset. If 20% of debt in the system are stable synths, then volatile synth issuers will receive 80% of debt and 100% of value appreciation—this corresponds to a leverage of 0.2x.

In practice, this means that staker strategies may be fairly complex—while stakers are exposed to a large amount of risk, they also have several possible sources of revenue:

- Fees from synth conversion on Synthetix are paid out pro-rata to the share of debt pool;
- SNX rewards for staking are paid out regularly;
- Profits from issuing volatile synths;
- Some situational incentives.[11]

The collateralization ratio is quite high at 800%, because of the wide variety of potential exposures gained by synthetic assets and their non-trivial nature. SNX stakers are incentivized to keep their debt (and, consequently, the platform) healthy, because a staker below the target collateralization ratio does not receive SNX rewards. Stakers below 200% collateralization can be liquidated, similarly to Maker.

The protocol allows to instantly convert arbitrary amounts of synthetics between different kinds of exposure with zero slippage, which can be powerful to hedge against price movements. Synthetix also offers inverted positions, synth assets that start at a particular price level of a real asset, but move in the opposite direction, at the same rate, representing short positions. Of course, trading synths is not a silver bullet, since the on and off ramps into synthetic assets are still prone to slippage and the usual market dynamics.

For instance, a capital holder with great confidence in the overall robustness of the system could slowly convert their capital to synthetics to be able to counteract short-term price movements by exchanging their volatile synthetic for a stable synthetic during price drops, since there is no slippage and infinite liquidity. On the other hand, converting these synthetics back to a real asset would also have to be done gradually, as slippage and liquidity depths would apply to the pair of synth vs real asset.

[11] E.g., there are currently SNX rewards for issuing iETH or iBTC, to balance the amount of long and short synths issued in the system.

Stabilization Mechanisms: Collateralized Debt model (CDP)

Price is 1+r (higher than peg)

Lower the Price by Increasing Supply

As all market agents bring newly created stablecoins to the market, supply increases driving price to parity

Price is 1-r (lower than peg)

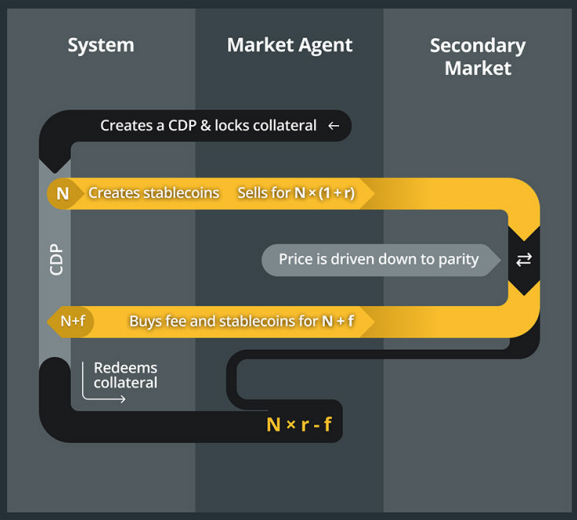
Support the Price by Creating Demand

By closing positions CDP owners will submit stablecoins to the system, driving excess supply from the market

Passive

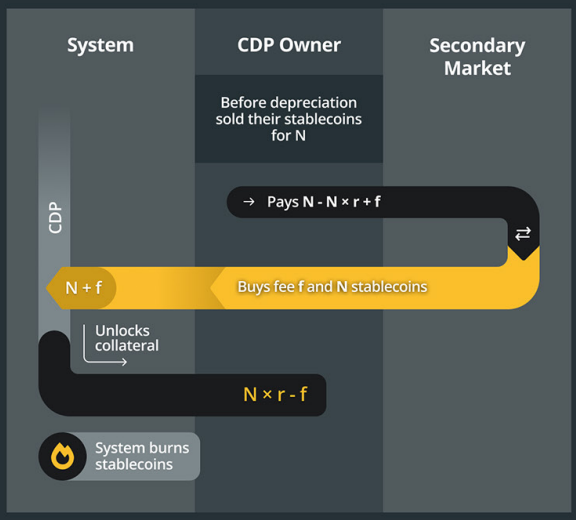
Arbitrage

N*r-f is the arbitrageur's risk-free premium that he competes for with other market agents. The competition ensures that price deviation is remedied quickly



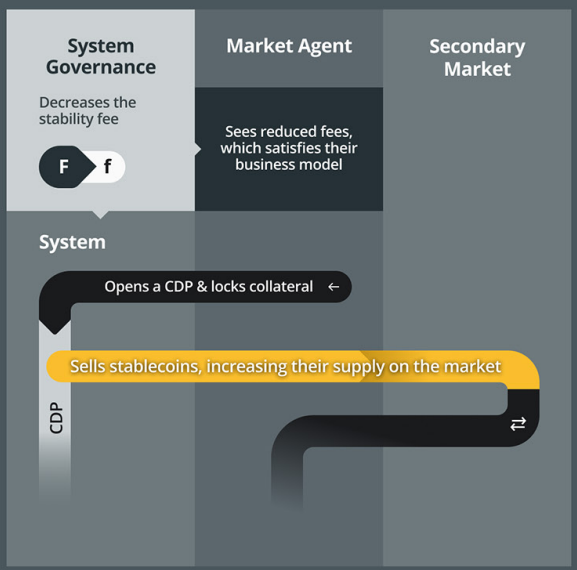
Arbitrage for Existing CDP Owners

Some users will create a CDP and immediately sell their stablecoins, entering a short position specifically to earn from arbitrage when the stablecoin depreciates



New Users

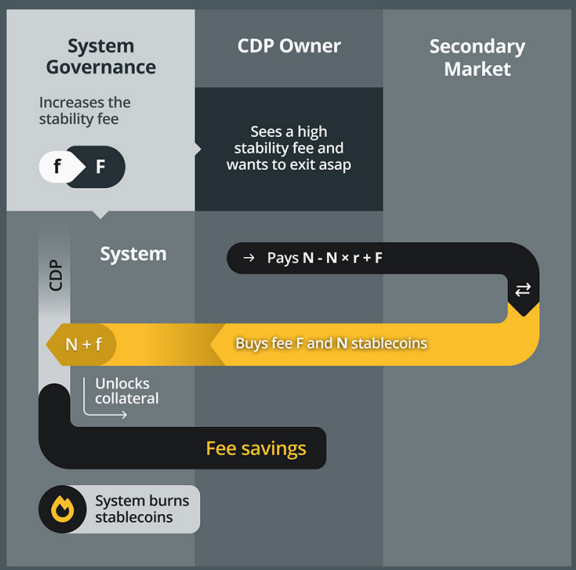
While not all new users will add their new stablecoins to circulating supply, some will. Thus, stablecin supply will gradually increase, as long as fees are kept low.



Active

CDP Owners Exit Positions Due to High Fees

If the fee is too big for the CDP owner, they will exit their position until fees drop again. When the stability fee lowers they may return



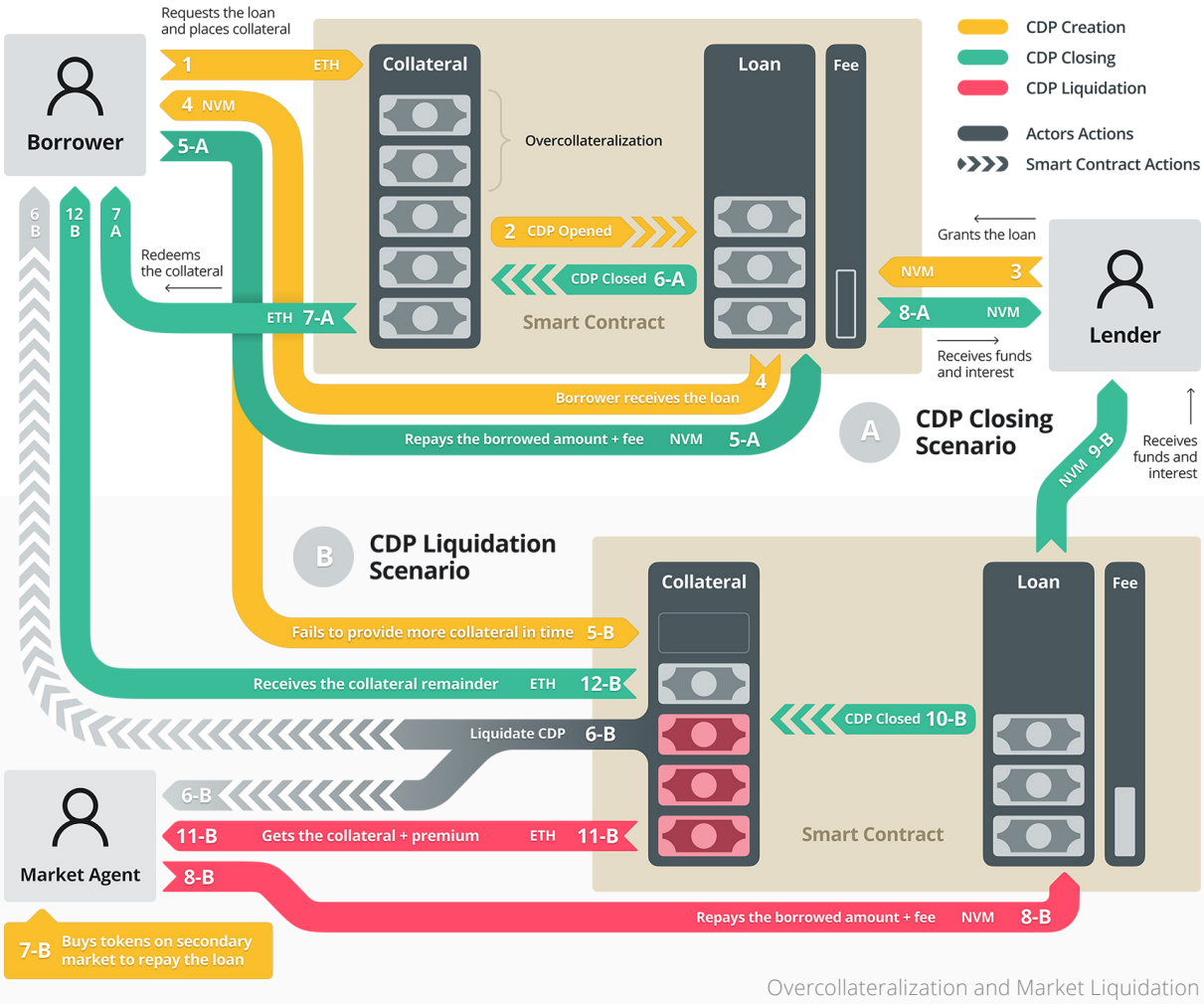
Stablecoin Other Crypto

PATTERN

Overcollateralization and market liquidation

The quest to eliminate centralization bottlenecks often calls for alternative solutions in most mundane mechanics. It is possible to set up lending markets without reputation systems and even mutual acquaintance between counterparties. The borrower puts up some collateral that covers the value of the borrowed assets, plus a safety margin. The loan is given out. Everything is governed through smart contracts that enforce the initial ratio of collateralization, take custody of the collateral and relinquish custody of the funds. Same is done, in reverse order, upon repayment.

The subtle problem is introduced by shifts in asset values. If the collateral asset depreciates the ratio of collateralization for the loan may fall below the target. The borrower may take action to add collateral or partially repay the loan, but may also fail to intervene. Since smart contracts cannot act on their own, a third party must, having monitored the market and the state of the loan, interject and perform a liquidation by selling the collateral and repaying the loan with accumulated interest.



Overcollateralization and Market Liquidation

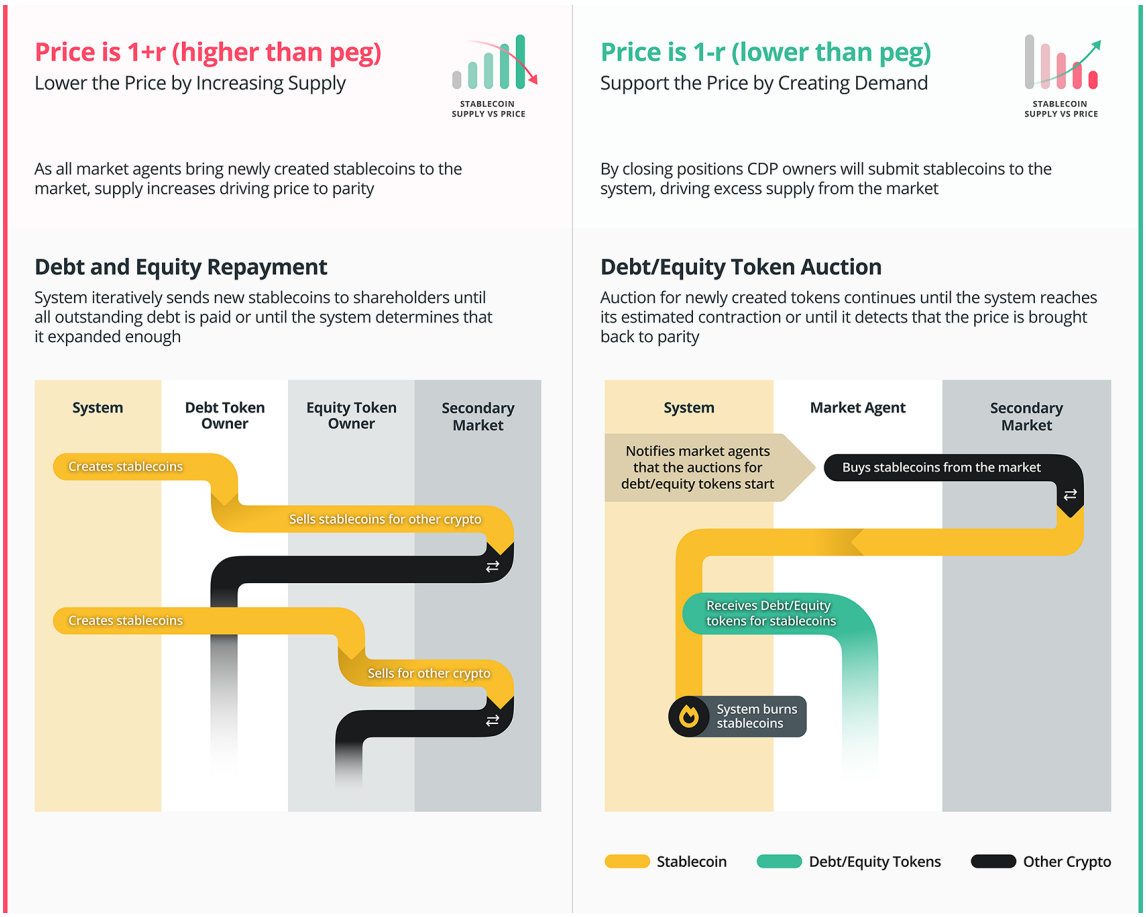
For the reasons of decentralization (to prevent potential foul play on behalf of the platform), a common approach is to outsource this task to arbitrary market agents. If a loan is under the desired collateralization ratio, anyone is allowed to, depending on the implementation, either buy out the collateral directly under some rules, or start an auction for the collateral. Both options incentivize market agents by offering the collateral at a discount to its market value.

2.1.2 Seigniorage share

In the seigniorage share model, the system does not necessarily utilize collateral—instead, it replicates some existing strategies that central banks use to regulate the supply of fiat currencies, namely, supply expansion and contraction through the issuance of treasury bonds.

This means that the seigniorage share model has a single stablecoin and one or more volatilecoins that act as IOUs from the system to holders.

The concept was introduced to a large extent by **Basis**, which subsequently failed to launch due to regulatory pressure. However, newer projects, such as **Reserve** or **Frax**, have emerged to improve and properly implement the concept.



Stabilization mechanisms: segniorage share

CASE STUDY

BASIS

Basis is the first practical project that aimed to implement the seigniorage share model, although the theoretical concept has existed for some time before it. Just like Maker for the CDP model, Basis is an implementation of the "pure" seigniorage share model.

Basis is a three-token system—it has the stablecoin, a bond token, and a share token. It starts with some initial supply of both stablecoins and volatile tokens. The system as a whole aims to stabilize the value by keeping the supply up with

the demand, using price as a signal—if the price is too high, the supply needs to be expanded to match the high demand. If the price is too low, the supply has to be contracted to match the low demand.

When the stablecoin depreciates in value, and the system needs to contract supply, it issues secondary tokens that represent debt/equity and auctions them off for stablecoins, burning the proceeds. Debt tokens assume a fixed premium in the future, while equity tokens gain dividends pro rata.

When the stablecoin appreciates in value, and the system needs to issue additional stablecoins, it issues them to the holders of debt and equity tokens. If the system issued both debt and equity, debt is repaid first, and the rest is distributed as dividends.

The base functionality of the system boils down to two processes of expansion and contraction. Because of the way Basis manages the price, it does not necessarily require collateral—one could say that the value of the token is backed by the platform debt to bond owners.

Ultimately, Basis would not launch due to a regulatory roadblock—the SEC classified bond and share tokens as securities, which immediately severely limited the scope of protocol users. This is especially unfortunate, considering that the seigniorage model has some significant theoretical concerns, which Basis could either validate or disprove in practice, but didn't get to.

INTERLUDE **On the feasibility of the "pure" seigniorage share model**

The seigniorage share model has some strong assumptions on the behavior of market agents. Firstly, if the stablecoin isn't collateralized, there is no lower limit on its value. This means that market agents will only try to acquire secondary coins if they believe that the stablecoin will appreciate in value in the future, granting them interest or dividends. If the stablecoin deviates from the peg too much (e.g. due to a black swan event), investors may lose faith in it and refuse to purchase the secondary coin, leading to a negative feedback loop that drives the price further down.

Secondly, it is generally not trivial to estimate how exactly the change in supply will correspond to the change in price. Most seigniorage share stablecoins assume the QTM (Quantitative Theory of Money) model and modify the supply proportionally to the price. However, QTM has historically shown itself to be only a heuristic—while the relations described by the QTM generally hold, the exact magnitude of effect of changing the money supply is challenging to evaluate. The system may try to detect the point where the price returns to parity, and stop expanding or contracting there. However, this moment may be delayed from the actual supply change, which will make the system expand or contract too much.

Note that in the case where price stabilization is mostly achieved through arbitrage, it is not required to estimate exactly how much the supply should be inflated or contracted—as long as there is profit to be made (i.e., as long as there is the difference between the market price and parity), arbitrageurs will continue to do transactions, driving the price in the appropriate direction.

While these concerns could be seen as purely theoretical, it is telling that since Basis' unfortunate end, no projects with the "pure" seigniorage model have emerged.

However, some recent stablecoin projects opt for a hybrid approach—using both collateral for stablecoins and bond tokens for supply contraction and expansion. This allows to achieve a lower collateralization ratio than the CDP model. Due to the system being at least fully collateralized, the market agents can always arbitrage between the system (which trades at the peg) and secondary markets, stabilizing the price. Prominent projects that have chosen this approach are Reserve and Frax.

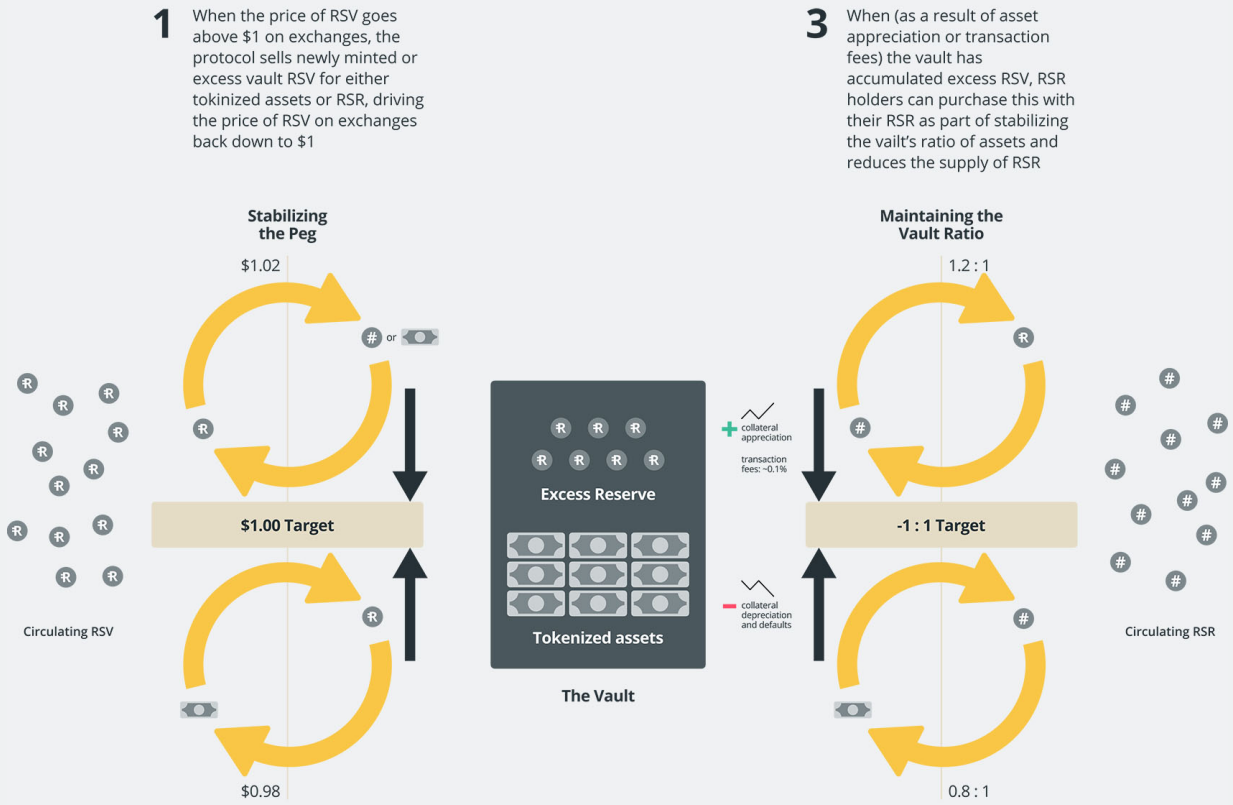
CASE STUDY

RESERVE

Reserve (RSV) is an in-development stablecoin that combines the seigniorage share design with collateralization. Because the stablecoin is fully collateralized, the system can maintain the peg through arbitrage—the contract always buys/sells at the peg, so arbitrageurs can expand/contract the supply naturally.

If the underlying collateral depreciates in value, the protocol mints and auctions off a secondary token RSR in exchange for additional collateral—RSR is sold through an auction. If the protocol cannot attract a sufficient number of RSR buyers, it will gradually lower the peg. RSR holders can also use RSR to mint additional RSV, as long as the demand is high enough to support it.

Reserve solves one of the major issues of the "pure" seigniorage share model—the necessity to estimate the size of the required expansion or contraction of supply. Because the peg is fully maintained by arbitrage, RSV will be minted or burned exactly as long as there is profit to be made.



Source: reserve protocol description

CASE STUDY

Frax is a newer stablecoin design that facilitates gradual transition from a full backing by another stablecoin (such as DAI or USDT) to a fully non-collateral token. Initially, Frax is minted 1 to 1 to another stablecoin.

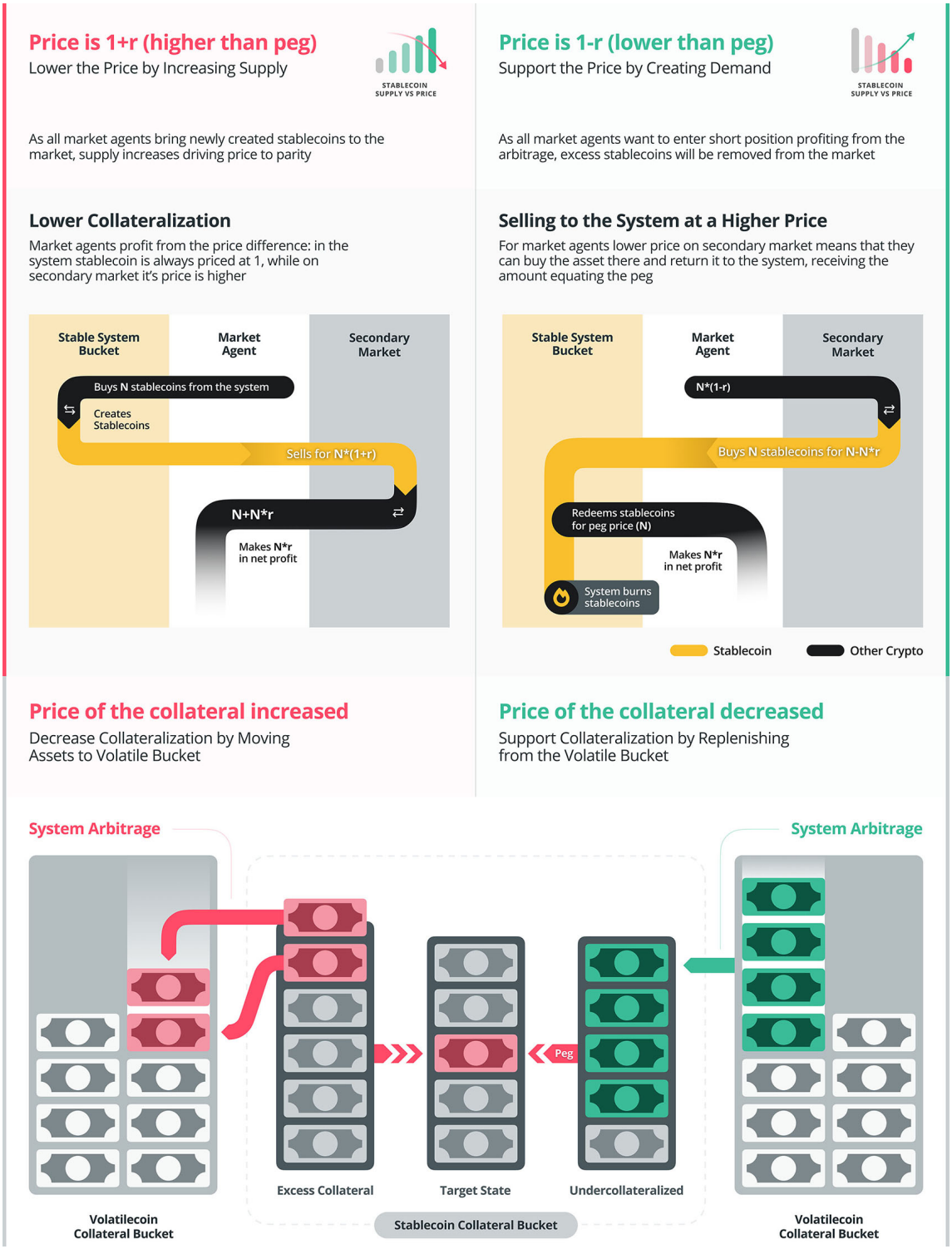
When Frax accrues a big enough supply, it starts to slowly remove the backing. The system starts to mint new Frax tokens to the holders of the seigniorage share token that is not backed by the reserve stablecoin. As long as the currency consistently trades above or at the peg, the collateralization ratio is decreased. However, to maintain the system's solvency, Frax stablecoins can be always redeemed for a portion of the reserve stablecoin (depending on the current reserve ratio) and some FXS, which can be later used to earn a premium.

This is an interesting design, as it is not fully reliant on market expectations—if the stablecoin depreciates, it can always fall back to a reserve stablecoin, and even increase the collateralization ratio back in especially severe cases by auctioning off FXS for Frax. However, it does not exactly solve the issue of supply change estimation—Frax functions on the assumption of QTM, expanding or contracting the supply proportionally to the price.

2.1.3 Profit/loss consumption

PL consumption has two types of coins—the stablecoin itself and a number of volatile coins that consume volatility from the stablecoin directly. Both types are collateralized, and collateral is continuously being redistributed between two (or more) "baskets" to keep the USD value of one basket constant.

Consider the simplest P/L consumption two-coin design, consisting of one stablecoin (STC) and one volatile coin (VLC), both collateralized by NVM. The system always exchanges NVM to STC at the rate of $1 USD worth of NVM to 1 STC. The peg is ensured by oracles. This means that the peg on secondary markets is maintained through arbitrage—when the price is higher than the peg, it is profitable to buy STC on the secondary markets and redeem it in the system, and vice versa.



Stabilization mechanisms: profit/loss consumption

The STC basket always has exactly enough collateral to redeem all STC for NVM. Conversely, redeeming VLC yields the amount of NVM that is dependent on a total amount of collateral after redeeming all STC. If the collateral depreciates to the value of all issued STC, VLC will be worth 0 in the system.

> *To demonstrate how rebalancing between baskets works, suppose there is 5 NVM = $1,000 of stablecoin collateral and the same is true for volatilecoins. NVM has appreciated 10% to $220. In this case, the stablecoin now only needs 4,54 NVM to collateralize all issued stablecoins, and it moves 0.46 NVM to the volatilecoin basket. Therefore, the value of volatilecoins increased not by 10%, but by roughly 20% (5.46 × $220 = $1,201.2).*
>
> *At the same time, suppose that from the same initial setup NVM depreciated by 10%. The stablecoin basket now needs 5.55 ETH to capitalize all stablecoins, therefore it draws 0.55 NVM from the volatile basket. The value of volatilecoins is decreased not by 10%, but by 20% (4.45 × $180 = $801).*

In a way, VLC represents a leveraged position on NVM, with leverage dependent on the ratio of the volatile basket value to stable basket value. The ratio also represents the resistance of the system to black swan events—the collateral asset will have to depreciate m times for the system to become insolvent, where m is the current ratio.

P/L consumption is a lot simpler than other mechanisms conceptually—while CDP-based stablecoins have an opaque relationship between the stability fee and CDP demand, and seigniorage share coins have the difficulty of estimating the required expansion/contraction, P/L consumption relies entirely on arbitrage and the simple process of rebalancing between baskets, without the need to estimate any "global" parameters or relationships.

However, VLC is a very specific asset that caters only to professional traders that are friendly to risk. This means that it is challenging for the system to attract enough demand for VLC to fully back STC.

The most prominent P/L consumption stablecoin project, Money-on-Chain, solves this somewhat by keeping a portfolio of several volatile coins which cater to different investor profiles.

CASE STUDY

MONEY on CHAIN

Money on Chain is a more recent P/L consumption design that implements two volatile tokens instead of one to increase the potential audience of volatile token consumers.

The first volatile token has small leverage, as part of the leverage is transferred to the second token with the higher level of risk. Because the second token "borrows" leverage from the first token, the holders of the second token pay interest to the holders of the first token. The interest rate also allows to incentivize system capitalization—as the coverage of the system (the ratio of the size of volatile buckets to the stable bucket) decreases, the interest rate increases, which incentivizes the purchase of the first type of volatile tokens.

Money-on-Chain also implements mechanisms to improve its resistance to black swan events—when collateral depreciates and the coverage in the system becomes too low, the system first starts to issue volatile tokens with a discount to attract more collateral, and, in extreme cases, liquidates, ensuring that all stablecoins can be redeemed at the peg.

Diversifying the profile of volatile token holders allows to make a P/L consumption system more secure, as it ensures that there are enough users bringing in collateral in different market conditions.

> *Therefore, there are two distinct profiles for volatile token holders—the first volatile token caters to long-term collateral asset holders, which enjoy slightly increased exposure and passive income in the form of interest. The second volatile token caters to professional margin traders who want larger exposure to the collateral asset.*

CLOSING REMARKS

Why Do Stablecoins Matter?

Stablecoins fill a very important niche in DeFi, as they are used in many applications across the board to manage various risks.

In the simplest case, stablecoins are the perfect medium for payment and everyday operations. Many companies that choose to perform their accounting with crypto utilize stablecoins for this purpose, since paying in volatile currencies necessarily subjects at least one counterparty to currency risk while money is in transit.

Stablecoins themselves can be a very efficient form of collateral, not only because opportunity costs only constitute the stablecoin's current lending interest rate, and are thus more predictable than volatile currency opportunity costs, but also because the risk of liquidation is much smaller, which allows to achieve a higher debt-to-collateral ratio.

Finally, most stablecoins provide opportunities for zero-risk return—their mechanisms for stabilization rely heavily on liquidation and arbitrage, which can be leveraged if one has the capital.

> *In the simplest case, stablecoins are the perfect medium for payment and everyday operations. Many companies that choose to perform their accounting with crypto utilize stablecoins for this purpose, since paying in volatile currencies necessarily subjects at least one counterparty to currency risk while money is in transit.*

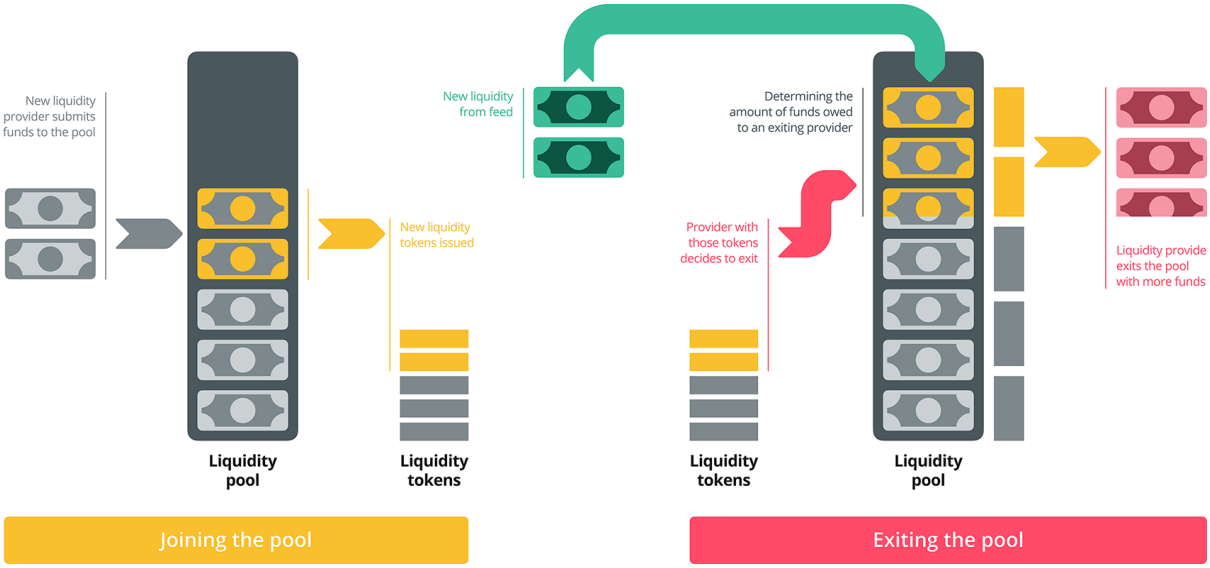
PATTERN

Tokenized Pooling

The decentralized approach to capital provision makes full use of the perfect accounting capabilities of blockchains. A pool of a particular asset (or assets) is associated with a list of users that own a particular share of that pool. This is tracked with a separate token that represents share ownership: whenever a user adds some value to the pool, pool tokens (often also called "liquidity tokens") are minted and assigned to this user in a way that the minted amount represents the same share of the token mass as the added liquidity in the whole liquidity in the pool. When the user exits, the pool tokens are burned, and the corresponding share of total liquidity is sent to the user. Entry and exit are usually at will.

Aside from liquidity deposits and withdrawals, usually some additional operations are allowed on the pool with a requirement that the total value of the liquidity in the pool (or other assets the system otherwise has custody of) is never decreased. For trading pools, the additional operation is exchange (adding one of two assets and removing the other one); for lending pools the additional operation is lending, with collateral being placed somewhere else into the system and liquidation mechanisms enforcing solvency of that collateral.

Assuming the pool funds have been employed, generating income in feed, and no other liquidity provider has joined the pool



Tokenized pooling mechanism

Tokenized pools often have an incentivization mechanic included—if the system based on the tokenized pool generates revenue, this revenue can be paid to pool owners by simply adding it to the pool. As no new pool tokens are generated, the value of each individual token increases proportionally. Prominent examples are fees in **Uniswap** pools, interest in **Compound**, or liquidation rewards in **Maker** 1.0 PETH pool.

3 Instruments

3.1 Trading

Trading is an essential part of a financial ecosystem: it facilitates liquidity of capital and efficient price discovery for the traded assets, the two paramount properties. It also creates niches for profit-seeking agents who get paid for improving market efficiency in those two senses.

In traditional finance, trading is done on large centralized exchanges that are exceptionally fast. Trades are processed within nanoseconds, and the exchange guarantees that all the instruments behave correctly: that orders are matched properly, settlement happens on time, and the balances are updated promptly and fully reflective of the actual changes in asset ownership. These results are achieved by consolidating a tremendous amount of computing power in one place. Most often the orders are represented as an order book, a collection of offerings by individual agents with a number of parameters set by the agents (such as the amount in units of the offered asset, price limits, cancellation conditions for some order types, etc.). Each agent can have many active orders in the same market at one time.

Due to inherent limitations of decentralized networks, this approach is extremely challenging to reproduce in DeFi. Decentralized networks must synchronize among a multitude of nodes that are distributed geographically and vary in their processing power. This introduces severe limits on processing (how much computation can happen in a unit of time), latency (how fast a particular single operation can be performed), network throughput (how much data can be shared on the network in a unit of time), and storage (how much data directly relevant to representation of the current market state may viably be held). Something like HFT is certainly not possible (yet), and diverse attempts have been taken to redefine some of the key components (such as custody, matching, clearing and settlement, and liquidity provision) in ways compatible with the given scalability constraints.

Existing designs of decentralized exchanges (DEXes) make trade-offs in the space of scalability, decentralization, and functionality. What can never be given up is the custody of funds: no party other than the owner should ever be able to freeze and/or revoke the assets deposited into the exchange. This is, of course, contingent upon the proper functioning of the smart contract code that holds the assets in the exchange.

The baseline design would be to perform everything on-chain, keep and settle orders on-chain, but move matching (the computationally heavy part) off-chain. This is very clunky and offers the worst UX, since every action has to go through the blockchain. It is also hard to guarantee execution, and the blockchain is cluttered with stale orders that either have to be cleaned up (processing load), or kept forever (dead weight in storage).

From that point, four powerful thoughts have guided the development of trading instruments in DeFi. They are not fully independent of each other but do not represent a linear evolution either. At the time of writing each thought is a representation of a design of at least one major DEX on Ethereum blockchain.

1. **Keep orders off-chain, settle on-chain** by showing to the smart contract orders matched somewhere else and signed by both counterparties, in effect acting as atomic asset swap transactions between the counterparties. The orders are processed by a central agent or a peer-to-peer network, presenting a choice between UX and censorship resistance.
2. **Reserve networks:** pool considerable amounts of liquidity together in a smart contract, so that on-chain matching can be reduced to a single interaction with one pool (or a single interaction with one of the few pools from a relatively stable set).
3. **Pool-based exchanges:** As in the previous approach, pool liquidity in a smart contract, but keep exactly one contract

per market (a pair of assets traded against each other), and set the exchange rate with a formula that keeps some kind of invariant property. Use the tokenized pooling pattern to allow liquidity provision by arbitrary agents. This group of designs is very efficient with regards to the blockchain resources: it uses a constant amount of data (just the pool sizes and ledgers of pool token holdings), provides the exchange rate via a simple calculation, and can execute in one action a trade of arbitrary size (up to the size of the pool).

4 **Layer-2 non-custodial exchanges:** Use advanced cryptography to connect a centralized exchange (with all its benefits of UX and latency) to a non-custodial asset-holding contract that executes settlement based on cryptographic proofs of the history of trades signed and executed by the user. This design acts as a layer-2 non-custodial exchange because everything that happens with deposits and withdrawals is supported by a cryptographic structure that layer 1 (the blockchain) validates.

INTERLUDE **DEX classification by matching and custody**

We propose a rough classification of decentralized exchanges based on 3 criteria: location of taker's capital, location of market maker's capital, and counterparty discovery mechanism. We provide a breakdown of each axis below.

Location of taker's capital:

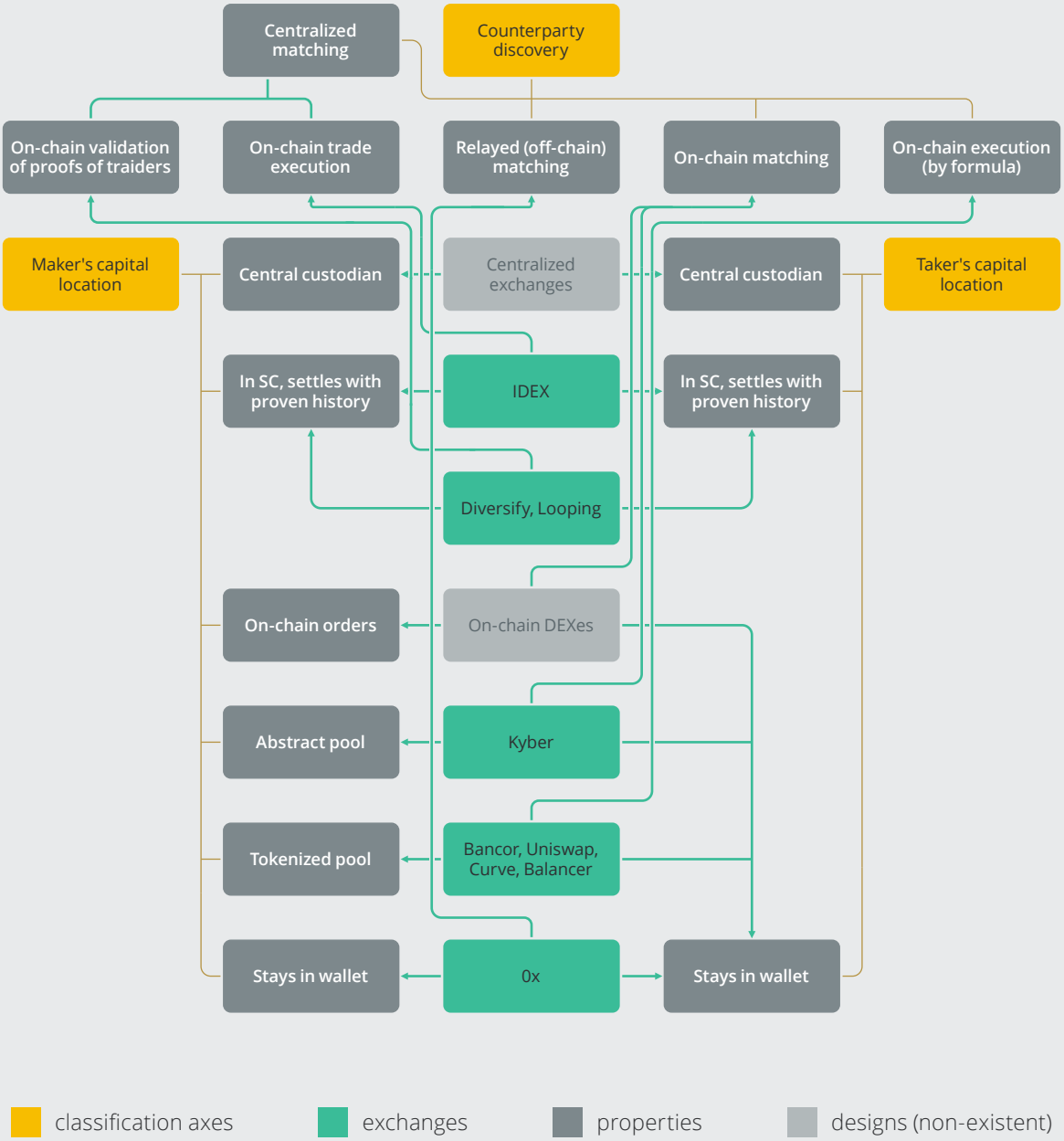
- Central custodian. Prior to trading activities, the user relinquishes the funds that were intended for trading into the custody of a third party (the exchange). This approach is used only for centralized exchanges and is included here for context.
- In smart contract custody. The user or the exchange provides to the contract sufficient data to validate the correctness of clearing and settle the trade. This is done either by processing each order individually or via a zero-knowledge proof.
- In the user wallet. When a match happens, the processing smart contract attempts to execute the trade atomically, either successfully moving the pre-approved funds from the users to each other or reverting.

Location of maker's capital (in addition to the aforementioned options):

- On-chain orders. The funds are locked in a market order that offers an arbitrary counterparty a deal on pre-defined terms. Setting aside custody issues, trades are usually processed in this way in centralized exchanges (the market maker's funds are locked in open orders).
- Tokenized pool. The funds reside in a pool that offers exchanges depending on the pool's inputs, the current size of the pool, and a deterministic formula. Similar to an order, the capital is locked until withdrawn, but unlike orders, the fulfillment element almost never takes up the entire capital allocation. More on that later.
- Abstract pool. The funds are locked in an entity that offers trade. The two previous options (wrapped in an interface) can in theory fall under this category.

Counterparty discovery:

- Centralized matching. As the title indicates, the counterparties are matched by a central agent. The two sub-varieties differ by whether the central agent also performs clearing.
- Relayed matching. Independent nodes share in a p2p manner outstanding orders that are known to them, and market takers can request that relayers match orders for them. The trade is then performed as a direct on-chain swap of assets.
- On-chain matching. The calculation to perform a match is done on-chain by a smart contract. This approach is not suitable for rapidly changing sets of outstanding offers.
- On-chain execution by formula. The maker counterparty is a smart contract, and therefore, technically, matching is performed by the taker. But with a tokenized pool, the number of individual liquidity providers whose holdings are affected by the successful completion of the trade can be arbitrarily high, with zero overhead on computation.


Centralized matching
Counterparty discovery
On-chain validation of proofs of traiders
On-chain trade execution
Relayed (off-chain) matching
On-chain matching
On-chain execution (by formula)
Maker's capital location
Central custodian
Centralized exchanges
Central custodian
Taker's capital location
In SC, settles with proven history
IDEX
In SC, settles with proven history
Diversify, Looping
On-chain orders
On-chain DEXes
Abstract pool
Kyber
Tokenized pool
Bancor, Uniswap, Curve, Balancer
Stays in wallet
0x
Stays in wallet
classification axes
exchanges
properties
designs (non-existent)

3.1.1 Off-chain matching with on-chain settlement

The first decentralized exchanges closely followed the design principles of their centralized counterparts while also attempting to introduce DeFi features into the mix. This led to the conception of non-custodial exchanges with centralized (off-chain) matching: while processing and matching of orders is carried out through a centralized engine, the trades themselves are completed atomically on-chain within an exchange-controlled smart-contract.

The orders themselves can be either stored on-chain, broadcasted to peers, or managed on a centralized service. Settlement always happens on-chain.

CASE STUDY

On-chain and off-chain orders

One of the first decentralized exchanges was EtherDelta, a DEX that had on-chain storage of orders as a fallback option, but otherwise stored orders on a centralized service and also used a centralized service for automatic matching of orders. The exchange was still non-custodial, since all the funds remained on the user's Ethereum account until the order was matched and the trade was executed atomically. However, EtherDelta was not censorship resistant, since the owner of the exchange could effectively censor some users by refusing to accept their orders.

EtherDelta is not particularly sophisticated, and has run into a number of technical and regulatory issues over the course of its existence. However, it was an important milestone in DeFi: the first fully non-custodial trading platform for tokenized assets. EtherDelta's success and failings significantly influenced the direction of further advancement of trading protocols in DeFi.

Simplest designs of this type have severe UX drawbacks. With off-chain matching and on-chain orders, it is hard to guarantee execution of a particular trade: the orders at the edges of the spread (biggest ask / lowest bid) are always targeted first by the matching engine, so if multiple takers want to make a trade at roughly the same time, there will be multiple transactions trying to compete for the same order. Only one of these transactions will succeed, and every other trade will fail, with the user having to manually sign some more blockchain transactions if they still want to take a trade. It gets worse if a taker wishes to match with multiple maker orders: if any of them is filled by another user, either the whole trade reverts, or the trade executes only partially, and the remainder has to be traded again in another transaction.

This dynamic can be averted by either having on-chain matching together with on-chain orders (not viable in a decentralized setting), or pushing the orders off-chain as well. Which reverts the design back to a non-custodial centralized exchange.

CASE STUDY

Off-chain orders, on-chain settlement per order

IDEX is a newer exchange engine that functions in a fashion similar to EtherDelta, albeit with many UX and efficiency improvements. Whereas in EtherDelta the user (taker for off-chain orders, and first maker, then taker, for on-chain orders) submits orders to the settlement contract, in IDEX, only the exchange is allowed to process orders. Therefore, the users have to make deposits onto the exchange and can make withdrawals only when the exchange uploads all of their orders (there is a time limit for this, preventing IDEX from freezing user funds).

This design offers several advantages (as described in the IDEX whitepaper), such as resistance to network backlogs and any kinds of racing conditions (as when multiple takers compete for a particular order by ramping up the gas prices for an on-chain transaction that would settle the order).

CASE STUDY

Relay networks

Another approach to improving off-chain order processing with on-chain settlement is to attempt to decentralize the matching part. This has been done by 0x, which is not a dedicated exchange but rather an engine that allows for the quick launch of a matching engine. The matching engines are called "Relayers" in 0x jargon; the user can pick any relayer of her choosing and submit the order to it. Relayers are incentivized to make matches happen with fees (similar to most other exchanges).

0x's useful tools for initial liquidity provision have made it a mainstay in DeFi. Some of the projects discussed further use 0x to connect their off-chain services to the chain. 0x itself hosts a set of in-house products and APIs for trading.

The future of this design is to enable the relayers to share outstanding orders between each other in a p2p manner, similar to blockchain gossip. This protocol modification is currently under development by the same team.

The potential for censorship resistance of the relay networks is powerful, but it remains to be seen whether this resistance can be implemented with correct balancing of incentives, and, if so, outweigh the UX and simplicity of the recently introduced scalable L2 exchanges. The two other approaches in this section do not have any advantages over L2.

3.1.2 Reserve networks

In the traditional approach to exchange design, orders are stored and processed individually. While this is manageable for centralized exchanges, the resulting storage costs are prohibitive in the decentralized setting.

This is why over time DEX designs sought to describe markets as efficiently as possible. Instead of individual orders, it is more expedient to describe a market within a smart contract by a small number of parameters that get continuously updated based on traders' interactions with this market.

The first protocol architecture that reflects this approach is Kyber Network. It is a protocol that acts as a router between traders and reserves that supply liquidity. In Kyber, a reserve for some asset can be created by any entity with a sufficient quantity of that asset. These entities may be large token holders, centralized exchanges with an on-chain connector, or even other DEXs.

The core contract of the network tracks all reserves. Takers (traders that want to exchange assets) come into the protocol through a single entry point—the core contract. When a taker asks for a trade, the core contract queries all reserves that provide liquidity for the required pairing, determines the best rate, and executes the trade atomically.

The business model of reserves is based on the bid and ask spread; therefore, the reserve owners (market makers) have to execute their strategy when implementing the reserve interface that is exposed to the Kyber Core contract. At the same time, they pay a KNC fee (Kyber's native token) for each trade.

The disadvantage of reserve networks is that they typically have significant entry barriers—e.g., in Kyber providing liquidity as a low-capital reserve is unlikely to be profitable due to KNC fees exceeding profits from any reasonable spread. As a result, big market agents can settle in easily, while low-capital agents have to coordinate without any assistance from the protocol.

This leads to underutilization of the long tail of liquidity providers, which constitute individuals and communities that want in on the trading fee profits.

While reserve networks made an important step towards a DEX design better suited to a decentralized setting, on their own they did not create any new market dynamics or business models that would improve on the status quo in traditional finance—the market depending on a small number of large agents providing liquidity.

It was the arrival of pool-based exchange designs that provided a way to democratize liquidity provision, engaging the long tail, while also facilitating markets that self-regulate, removing the necessity to be maintained by a centralized party.

3.1.3 Pool-based exchanges and constant product

An entirely different way to approach market making arrived with tokenized pools that allow many individual capital holders, however small their holdings, to pool their capital together and use that in market making via a so-called automated market maker (AMM). The tokenized pools are used to source liquidity for trading, while the AMMs are the vehicle for price discovery and an entry point for trading itself.

To explore how these patterns are applied in practice, let's consider the examples provided by the first two pool-based exchanges: Uniswap and Bancor.

CASE STUDY

Early pool-based exchanges

Uniswap and Bancor are the largest and most prominent pool-based exchanges.

Uniswap V1 and Bancor work quite similarly—they maintain a pool for each traded asset. This pool contains the traded asset and a dollar-equivalent amount of some "intermediate asset". The intermediate asset is traded to all other assets, meaning that to trade two arbitrary assets to each other, one has to first trade the first asset to the intermediate and then trade the intermediate to the second asset. In Uniswap V1 the intermediate is ETH, while in Bancor the intermediaries are the platform's native "smart tokens".

The pools are filled by free market agents with capital to spare in exchange for liquidity tokens. Trades accrue fees, which are also added to the respective pool as a reward for liquidity providers. The liquidity provider must always add liquidity in the same proportion as the current quantity ratio of the asset and the intermediate, and the provider will always receive liquidity back in the current ratio. This means that adding or removing liquidity changes the depth of each market, but not its composition.

The AMM in each market handles trades. This market maker determines the exchange rate of assets based on the ratio of their quantities using a mechanism called "constant product". The AMM keeps the product of the quantities constant across all trades, so the value Asset1Qty × Asset2Qty always remains the same, as long as no liquidity tokens are created or redeemed. Based on this invariant, the AMM determines how much of Asset 2 it should give when someone brings it 1, or 5, or 10 units of Asset 1.

> *As an example, suppose that there is currently 100 ETH and 20,000 DAI in the pool (which means that the spot market price for 1 ETH is 200 DAI). If a trader brings 1 ETH to the system to exchange for DAI, the new size of the ETH pool is 101. Thus, the AMM will maintain the invariant by solving 101 × newDAIPoolSize = 100 × 20,000. The solution is 19,801.98, which means that the AMM will give 20,000 − 19,801.98 = 199.02 DAI for 1 ETH. Notice that the trader received slightly less than the point market price—the constant product mechanism implicitly handles slippage and reduces in continuous fashion the exchange rate for larger trades.*

Uniswap and Bancor created an entirely new business model for liquidity providers and implemented an entirely automated exchange design. However, there is a side effect of the constant product mechanism that impairs the fee-driven business model—impermanent loss.

INTERLUDE **Impermanent loss**

The fact that the exchange rate is entirely determined by the relative sizes of pools leads to interesting consequences. On the one hand, tokenized pools in their basic form do not rely on oracles to determine prices (which removes a possible attack vector); instead, any discrepancies with other markets are entirely handled by arbitrage. On the other, this arbitrage opportunity creates a risk for liquidity providers called "impermanent loss".

Consider the following example:

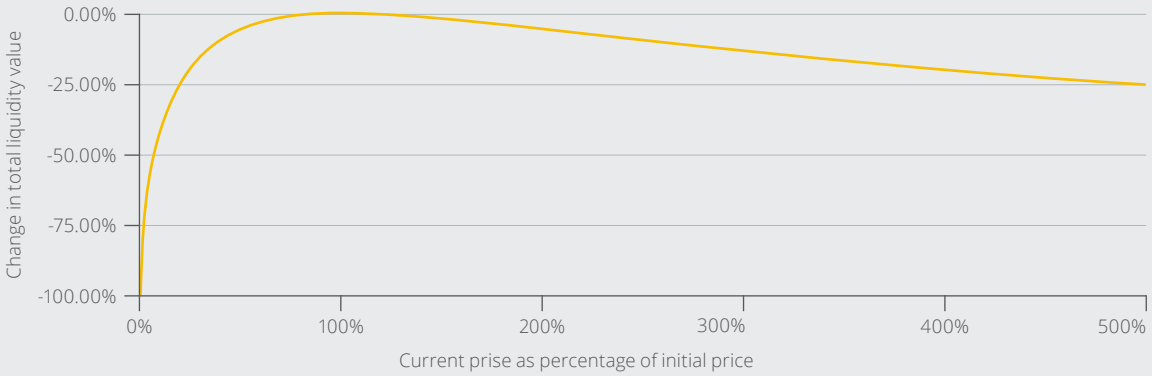
- We again start with the initial conditions of 100 ETH and 20,000 DAI in a market.
- Suddenly, the price of ETH on other markets surges to 400 DAI (a 100% increase).
- The arbitrageurs can take their DAI and purchase ETH at a highly favourable price, until the AMM starts to sell ETH for 400 DAI. This will happen when the size of the DAI pool is 28,284.27 and the size of the ETH pool is 70.71.

- Notice that 70.71 × 28,284.27 = 100 × 20,000, and 28,284.27 / 70.71 = 400.
- If 1 DAI = 1 USD, and 1 ETH = 400 USD, then the total value of the pool is 56,568.54.
- However, consider what would happen if the liquidity providers held DAI and ETH in equal value instead of depositing them into a pool—their total value would be now equal to 100 ETH × 400 + 20,000 DAI = 60,000 USD.
- The 60,000 − 56,568.54 = 3,431.46 USD of value was extracted by arbitrageurs as their profit.
- While the liquidity providers have not, technically, "lost" money, they have incurred an opportunity cost that is called "impermanent loss".

Impermanent loss is called "impermanent" because the liquidity providers would get it back if the exchange rate between ETH and DAI were to go back to 1:200, returning the relative pool sizes to the initial values. In fact, the impermanent loss is completely determined by the proportional change of the exchange rate and is the same regardless of the direction of the change.

Losses to liquidity providers due to price variation

Compared to holding the original funds supplied



As shown in the figure, the loss is relative to a benchmark of simply holding the assets. Therefore, the losses to liquidity providers are in terms of the benchmark to holding instead of staking and providing liquidity. The loss is the same whichever direction the price change occurs. A doubling in price of ETH results in the same loss as a halving. For example, a 1.25x price change results in a 0.6% loss relative to holding. A 2x price change results in a 5.7% loss relative to holding, and a 5x price change results in a 25.5% relative loss.

Impermanent loss is a significant risk incurred by liquidity providers, who remain in a constant "tug of war" with the yield from fees. Impermanent loss necessarily exists in a constant product pool if the pool uses arbitrage to align prices with external markets.

There are generally two ways to do away with impermanent loss.

One is to feed external market prices through oracles. For example, **Bancor V2** opted to use **Chainlink** oracles for price feeds, maintaining a constant product of asset values, rather than amounts. While this way impermanent loss is eliminated, the approach introduces an additional attack surface, as vulnerabilities in oracles can be leveraged to break the protocol.

Another way is to introduce more complex mechanisms and invariants into markets. This approach is generally harder and often leads to non-trivial trade-offs, but can achieve the same effect without oracles. This kind of approach is used in Curve, Balancer and Mooniswap.

CASE STUDY

Curve

Stable exchange rate pools

Curve is a pool-based exchange focusing specifically on trading highly correlated assets, such as USD-pegged stablecoins or various "wrapped BTC" assets. Curve's solution significantly reduces slippage and impermanent loss for asset pairs where the exchange rate does not significantly deviate from 1:1.

Curve's invariant is tailored to maintain a close to 1:1 exchange rate when the pool is balanced (e.g., the ratio of quantities of assets is close to 1), but starts to sharply deviate when one of the assets gets depleted—even more sharply than the typical constant product. As a result, the in-market exchange rate will be close to 1:1 for all but the biggest transactions. As a consequence, impermanent loss is also next to non-existent. Note that this solution only functions when the exchange rate is stable—otherwise arbitrageurs would quickly drain the pool when the external exchange rate changes.

Another important innovation of Curve is the introduction of interest-bearing pools—instead of underlying assets, Curve holds interest-bearing tokens, such as cDAI or cUSDC in the pool. As such, liquidity providers are able to accrue interest on their liquidity as well as transaction fees. At the same time, this does not hinder trading—interest-bearing tokens can be exchanged to underlying assets atomically, within the same transaction.

As a result, Curve presents a very sustainable business model for liquidity providers, as they are able to obtain a decent ROI with very low risk, compared to other tokenized pools. On the other hand, Curve does not provide a lot of choice in terms of investment strategy—for example, one cannot acquire exposure to arbitrary assets, when supplying liquidity on Curve.

CASE STUDY

Balancer

Multi-asset invariants

Balancer is a new tokenized pool exchange project that tries to mitigate somewhat the impermanent loss issue by allowing exposure customization.

Balancer's unique feature is that it allows users to customize the invariant in the "constant product" mechanic. Whereas in Uniswap or Bancor the invariant is a product of quantities of two assets, Balancer allows users to set up a pool with up to 8 assets having arbitrary weights. Using automated market controllers also allows adjusting weights on the fly, depending on the amount of available liquidity on both sides.

The pools with a non-standard invariant have several important use-cases:

- For new projects, setting up a 90/10 (or any large ratio) pool with project token and ETH/DAI allows bootstrapping liquidity for the project without a large capital requirement to buy the reserve asset (ETH or DAI). Over time, the weights can be adjusted towards a more even ratio, as the project acquires traction and more capital, which gradually increases market depth.
- Liquidity providers that are bullish on a particular asset can increase their exposure to it by setting a large weight. If a weight assigned to an asset is close to 1, then the liquidity provider experiences a level of exposure that is nearly the same as simply holding the asset. For example, setting up a 0.99/0.01 ETH/DAI pool allows almost full exposure to ETH. This is closely tied to the concept of impermanent loss—setting a large weight ratio reduces impermanent loss significantly on the asset with a large weight. If this asset is traded against, e.g., a stablecoin, impermanent loss becomes insignificant.

- Depending on the direction of trades in a particular pair, the weights can be updated dynamically to allow for a higher market depth. E.g., if in a BAT/DAI pool traders mostly trade DAI for BAT, then BAT's weight can be increased so that there is less slippage in this particular direction. Conversely, if market dynamics reverse and now traders are selling BAT, the weight of DAI can be increased to match liquidity depth with demand.
- Updating weights dynamically can stabilize pools that trade interest-bearing assets against each other. If a pool trades cDAI/cUSDC (Compound liquidity tokens for DAI and USDC, respectively), it may incur impermanent loss even when DAI/USDC exchange rate is stable. This is due to cDAI and cUSDC having different interest rates. If the interest rates are 5% on cDAI and 3% on cUSDC, over time the value of cDAI in the pool will deviate from the value of cUSDC, leading to impermanent loss. As a solution, the weights in the pool can be updated according to interest rate, keeping the exchange rate stable.

There is a price of setting a high ratio of weights, however. Large weights on assets lead to capital inefficiency, since one has to submit liquidity proportionally to weights. For example, in a 0.99/0.01 ETH/DAI pool, one has to submit 19,800 DAI per each ETH of liquidity—the capital requirements for the same market depth grow proportionally to the ratio of weights. This means that in an uneven pool, slippage will generally be high (unless trading is one-way).

CASE STUDY

Mooniswap

Pools with delayed balance updates

Mooniswap is a tokenized pool exchange that reduces arbitrage profits on slippage through a mechanism called "virtual balances".

Typically, when a high-slippage transaction happens in a pool, the fastest arbitrageur corrects the balances to reflect the external exchange rates, extracting all the extra value that the trader added to the pool with slippage. Note that the arbitrageurs profit is non-linear from the balance deviation, since they get more of asset A for each unit that the quantity of asset B deviates from the "optimal" balance.

Mooniswap introduces virtual balances of underlying assets that move towards the real balances over the span of 5 minutes after a trade, instead of immediately. Balances are only affected in the direction opposite to the previous trade (i.e., the direction in which arbitrageurs will trade).

Since virtual balances move towards real balances over time, arbitrage trades will not be immediately profitable for arbitrageurs due to gas prices. At the same time, arbitrageurs cannot wait until the virtual balances reach real balances, as there is competition and other arbitrageurs may execute a trade first.

As a result, arbitrageurs will cover the deviation in several smaller trades over time, instead of one big trade. Since profit is non-linear in the deviation covered, the total profit of arbitrageurs will be significantly smaller, and the pool will capture most of the value from slippage.

While Mooniswap does not solve impermanent loss per-se, it compensates by creating a new source of profit for liquidity providers.

3.1.4 Layer 2 non-custodial exchanges

All DeFi protocols that exist entirely as a set of smart contracts on-chain are limited by the chain's scalability. For example, each swap on Uniswap is a separate Ethereum transaction, which means that the absolute upper limit for Uniswap at the time of writing is around 15 transactions per second.

This is a long-standing issue for blockchains as a whole, and the most prominent way to solve it today are Layer 2 networks.

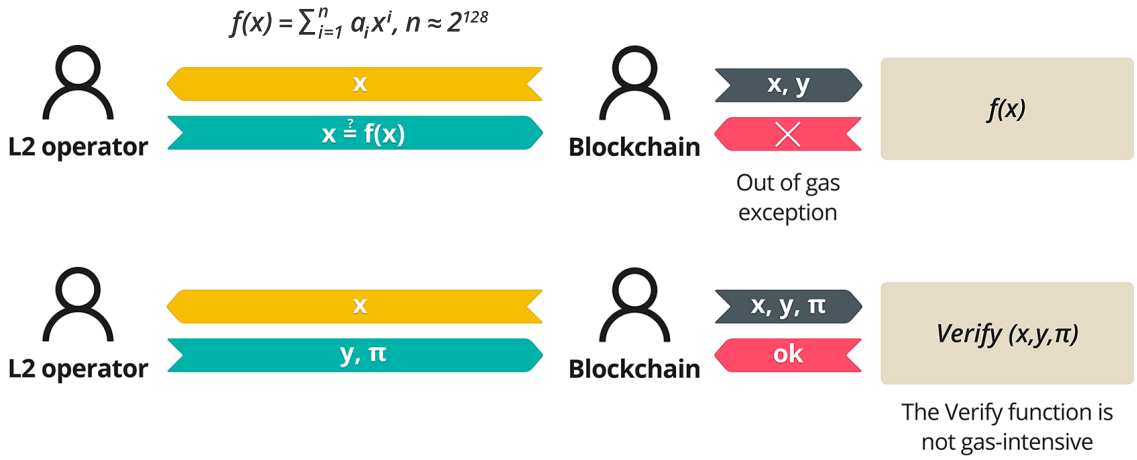
Layer 2 networks are separate networks that have their own nodes (or often, for the sake of scalability, a single *operator*), which are separate from the main network (also called Layer 1). The security of the L2 network is achieved by providing some sort of proof in a smart contract on Layer 1 that the "rules" within L2 have not been broken. While the full set of rules can vary greatly depending on the use case, some of the common rules are:

1. A user can only transact with funds that belong to them;
2. A user can only withdraw from L2 funds that belong to them;
3. The total amount of funds in L2 is equal to the amount deposited into the L1 contract.

While great strides have been made with theoretical research into the sphere, producing concepts such as **Plasma** or **zk-Rollup**, the first practical L2 products ready for end consumers were deployed only recently on Ethereum—**DeversiFi** and **Loopring** are powered by technology called *SNARKs*.

SNARKs (succinct non-interactive arguments of knowledge) are state-of-the-art cryptographic gadgets that produce small, quickly verifiable proofs of formal statements, regardless of the statement size. Usually the statement encompasses some formalized computation, and the SNARK proves that the computation was done correctly—that is, that its intermediate and final results adhere to some public set of constraints.

In the case of exchanges, the statement to be proven is that a particular snapshot of user balances has been reached by applying only correct transformations: deposits (reflected in the L1 anchor contract), trades (signed by private keys of both users when both users had sufficient balances in their respective assets), and withdrawals (again, reflected in the L1 anchor contract). The L1 contract periodically receives from the exchange proofs of the most current snapshot, and the user, in order to make a withdrawal, only has to present a proof tracing back their balance to that snapshot. This verification can be established by L1 without any interaction from the exchange, as long as the user has all the necessary data (see data availability and L2 solutions).



Blockchains can't handle large computations due to scalability constraints, but the computation may be outsourced and the provided proof used to verify the correctness in the on-chain smart contract (which is a computation that can be handled by a blockchain). In this example, the outsourced computation is a very large polynomial. In practical applications, computations are considerably more complex, but the same basic principles apply.

CASE STUDY

Zk-SNARK and zk-rollup exchanges

DeversiFi and Loopring are two exchange solutions that function quite similarly, their differences being mostly technical in nature.

L2 exchanges structure a normal exchange operation as one of the SNARK-enabled formalized computations, which allows the exchange operator to prove that no cheating has occurred with off-chain funds deposited by users. The smart contract on L1 completes the verification of the proof by passing the inputs, the proof, and the output to a special "Verify" function. If L1 is secure, then the L2 app must be secure as well, but if the operator has broken the rules somewhere, the L1 contract will reject the provided proof, and, consequently, the output.

The input and output to the computation are the balance sheets before and after a single exchange tick has been processed. While these balance sheets may be too large to place on the blockchain, their hashes can be used instead as the input and the output for "Verify". The prover needs to additionally prove that the actual balance sheets hash to the provided values, but this can be done within the same proof.

In practice, all of this means that DeversiFi and Loopring remain non-custodial (i.e., operators cannot arbitrarily take someone else's funds for themselves, as this would break the proof) with much higher processing speeds than fully on-chain protocols such as Uniswap. This is a remarkable achievement, but it comes with its own set of trade-offs.

Firstly, while the solutions are non-custodial, they cannot be considered entirely decentralized—the operator may simply refuse to process transactions or even accept deposits from some users. Cryptographic proofs can ensure that accepted transactions are executed correctly, but they cannot ensure that all transactions are accepted.

Secondly, there is the so-called "data availability problem", which can be leveraged to mount attacks on L2 networks, and which both solutions grapple with.

INTERLUDE

Data availability and L2 solutions

The tradeoffs made by L2 solutions revolve around the concept of data availability. To benefit from the security of the main chain, an L2 solution must have an anchor in it, which is represented by an L1 smart contract. This smart contract verifies the proofs of correct behavior that the L2 operator supplies, and to do that, it must have data to verify those proofs.

The challenge is that the entire state of the L2 network (against which the contract would verify the proof) is usually prohibitively big to put on L1. This can be solved by using **a cryptographic accumulator**[12], that is, a gadget that compresses a large amount of data into a single small value and then proves that some chunk of data was indeed compressed into that value.

This approach allows for the representation of a large set of data on L1; however, it comes with a tradeoff: any user who wishes to withdraw funds from L2 must first prove to the L1 contract that there is an account in the accumulated dataset that belongs to them. Usually, in order to construct **a witness** that proves this, one has to have access to the entire accumulated dataset, and the only user who (reliably) has such access is the operator.

[12] The most well-known cryptographic accumulator type is the Merkle Tree, extensively used both in Bitcoin and Ethereum. To construct the Merkle Tree, the data is split into chunks, then those chunks are hashed. Each pair of hashes is then hashed together, and this process is repeated until only one hash remains. The final hash represents the entirety of data, and anyone can prove that the chunk was hashed into a tree by providing a correct path of hashes in the tree.

This means that users who want to make withdrawals must rely on the operator to provide the necessary data. However, the operator may choose to withhold this data, and, unless someone else happens to have it, the user will have to use a witness for some accumulator that represents an old state. However, there is no way for the contract to differentiate between genuine users who have encountered an availability problem and malicious users who intend to withdraw money that no longer belongs to them by referring to some state in which it once did.

This is why data availability is a significant challenge for all L2 designs. There are various approaches to solving the problem—Loopring opts for storing the state on L1 as efficiently as possible, whereas DeversifFi relies on a committee that attests to the availability of the state at a particular time, with the committee held liable for this attestation. In practice this means that Loopring is somewhat slower than DeversiFi but has more robust security guarantees.

It is important to note that the above is also a key challenge in designing sharding solutions for scalable blockchains, as the relationship between the shard chains and the coordinating chain resembles that of an L2 solution and an L1 chain. As such, the research into this topic has already borne fruit with the discovery of so-called "data availability proofs"—a cryptographic protocol that will be used in, e.g., ETH 2.0. As of now, however, this protocol is not deployed in production.

Finally, L2 exchanges are currently separated from the rest of the DeFi ecosystem. Because typically only on- and off-ramping functions are on-chain, other on-chain protocols cannot interoperate, resulting in the loss of many synergies and integration opportunities.

Still, despite their limitations, Deversifi and Loopring massively contribute to the improvement of UX in DeFi, and the research leading up to their creation has moved entire disciplines within cryptography forward. This is why Deversifi and Loopring are and will remain for some time among the most exciting developments in DeFi.

CLOSING REMARKS

Exchanges interconnected

It is interesting to consider how DeFi asset exchange infrastructure shapes up very differently from the way exchanges in traditional finance operate.

Large traditional centralized exchanges can be considered self-contained ecosystems, each having their own liquidity, asset roster, and users.

In contrast, DeFi exchange protocols strive to integrate and consolidate liquidity as much as possible. Kyber serves as the single entry point to all token swaps, presenting the user the best rate out of all protocols, whether they represent semi-centralized exchanges or tokenized pools. Tokenized pools incentivize increasing market depth within a single protocol as much as possible, as that brings bigger volumes and more revenue for liquidity providers. 0x presents a convenient interface to connect off-chain engine to on-chain settlement, which is, consequently, used by DeversiFi and other projects.

As a result, the exchange infrastructure is a unified space of liquidity, assets, data, and users that consistently creates new business models and is available to anyone at any time. It is a prime example of why DeFi is so exciting.

In addition, as a consequence of their own development, exchange protocols augment the rest of the DeFi ecosystem, as when the latest Uniswap release brought forth a robust new oracle protocol, or when L2 solutions have more generally moved cryptography forward.

PATTERN

Leveraged positions with reborrowing

It is possible to achieve a leverage on a position using only borrowing and trading instruments. The procedure is simple:

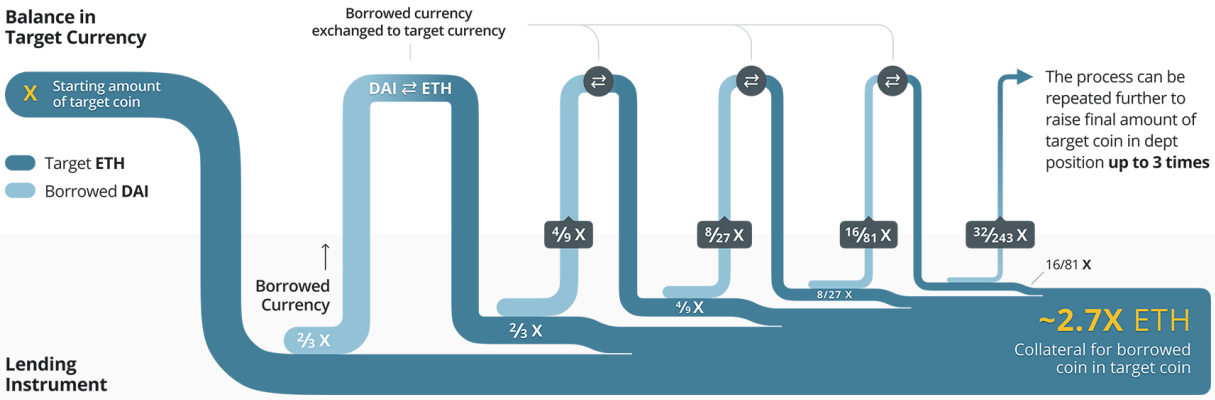
1. Place some collateral and take a loan.
2. Convert the currency of the loan into the currency of the collateral at an exchange.
3. Use the new funds to add collateral, increasing the money available to borrow.
4. Repeat.

One can acquire long exposure by borrowing a stable asset against a target asset, while a short position requires borrowing the target asset against a stable asset. This means that, e.g., Maker does not support short positions, but Compound does.

Each borrowing cycle brings diminished returns (as the collateralization ratio in borrowing instruments is bigger than 100%) and increases the leverage. Reducing the ratio of collateralization (up to the limit set by the instrument) increases the risk and the leverage: if the collateral price relative to the borrowed asset goes down, a call for recapitalization or liquidation will happen, bringing chain losses on each of the re-borrowed pieces. On the other hand, if the price goes up, each piece brings its own capital gain that can be used to cyclically close the positions (i.e. use some external funds to repay the first loan, sell its collateral for increased price, use gains to repay the previous loan, etc.).

With flash loans, the external funds can be borrowed without putting up even more collateral. We provide a full walkthrough of that scenario in the section on Synergies.

Re-borrowing

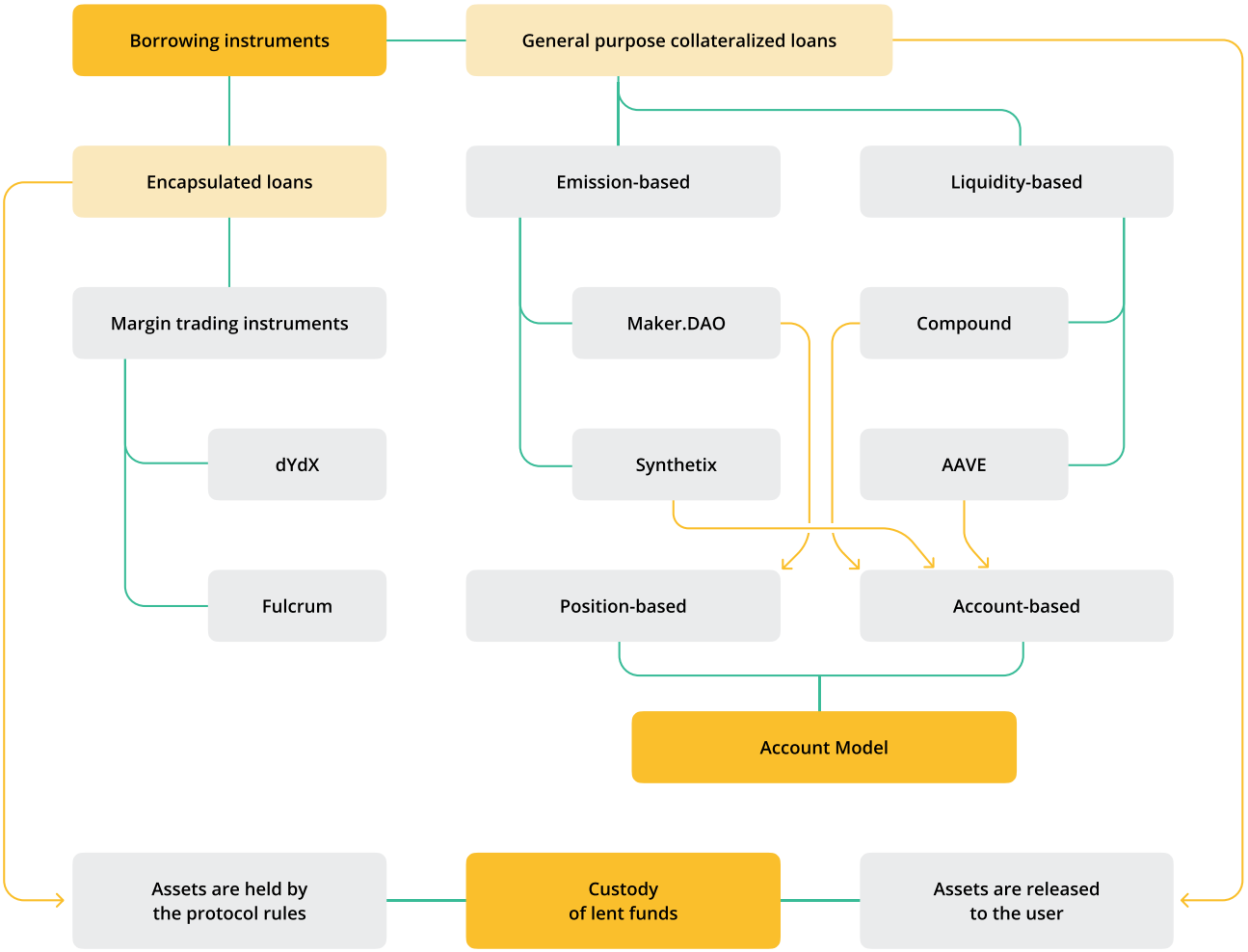


3.2 Borrowing and debt markets

DeFi wouldn't have been a financial ecosystem without the ability to draw debt to increase one's capital. This is where numerous lending/borrowing solutions come into play.

Lending in DeFi is most often based on two patterns—tokenized pooling and market liquidation, which allow to largely automate the process. A borrower does not have to source potential lenders or negotiate the interest rate, because a lending liquidity pool is a single entry point for all loans, and the interest rate is determined algorithmically from market conditions, based on the supply and demand in a particular pool. Market-based liquidation keeps the protocol capitalized, regardless of the behavior of borrowers, which ensures lender security.

At the time of writing, loans come in several shapes—based on the purpose, they can be collateralized and liquidated in different ways. Currently, there are general purpose overcollateralized loans, as well as borrowing of leverage for margin trading. One important difference between the two variations is custody: general purpose loans relinquish custody of the lent assets to the borrower (and therefore require overcollateralization to guarantee solvency for the liquidity providers), while specialized loans used in margin trading generally hold custody of the lent assets within the protocol, offering instead control over the leveraged positions: the user is always able to close her position (unless it is automatically liquidated during a margin call), but she has no direct control over the assets lent to her, they are controlled by the protocol.



Classification of borrowing instruments by function and account model

3.2.1 General purpose collateralized loans

In the early days of the DeFi ecosystem, borrowing was split between collateralized loans and peer-to-peer lending. These days, pure peer-to-peer lending is all but gone: there were ultimately too many barriers to enforcing loan repayment in a decentralized ecosystem. Loan repayment without any kind of financial incentives eventually falls back on institutional enforcement, which is exactly what decentralized finance tries to do away with. Instead, financial incentives have to be implemented for lending, which comes in a form of posting collateral.

Collateralized loans were already somewhat explored in sections dedicated to CDP stablecoins and in the "overcollateralization and market liquidation" pattern. The general principle of putting up excessive collateral to guarantee repayment, with market liquidation used to monitor solvency and convert collateral asset to the lent asset, is employed universally throughout this category.

While at the first glance overcollateralization could be seen as a source of great capital inefficiency, the unique properties of DeFi enable treating the posted collateral as a separate asset that keeps making money for the owner. Moreover, algorithmic enforcement means that the borrower does not relinquish ownership of their collateral—it is locked in a contract and cannot be taken by any third party, as long as the loan is properly capitalized. But even then, what happens to the collateral is directed by the protocol rules, which usually enact some kind of liquidation, automatically returning the remainder to the original borrower. Ultimately, the borrower doesn't sacrifice ownership of their collateral, and incurs very little opportunity cost due to posted collateral having its own yield.

The only downside is that some initial capital is required to utilize collateralized loans. However, in finance loans are typically used to improve one's liquidity, rather than raising one's capital from zero. This means that collateralized loans in DeFi are uniquely well-suited for loan use-cases typically appearing in finance.

The two variations of general-purpose loans are distinguished by the approach to sourcing liquidity:

1. **Emission-based:** the lent asset is minted by the protocol on the spot. Therefore, the system only serves borrowers, and not lenders (as the protocol itself is the only lender in every sense), and the potential size of all of the loans combined, the debt ceiling, is only limited by protocol governance. This is the historically first approach, taken by the first non-p2p lending protocol, Maker.DAO, but the definition also covers protocols such as Synthetix which is very similar in that regard. To be viable, the lent asset has to present some unique properties: stability in the former case, and synthetic exposure plus zero-slippage trades in the latter.
2. **Liquidity-based:** the lent asset is provided by third parties, so the whole system is somewhat similar to a peer-to-peer setting, but peers on the liquidity provision side are aggregated into a tokenized pool.

Emission-based loans

The history of collateralized debt instruments started with Maker.DAO, which established the hallmark design pattern and set a great example of how debt could work in a decentralized ecosystem. However, at the time Maker only focused on two instruments—DAI and ETH, respectively.

While the collateral aspect could (and, in fact, was, in Maker 2.0) broadened by adding more assets, not much can be done with the debt asset without finding ways to attract liquidity from arbitrary agents.

CASE STUDY

MAKER

Borrowers without lenders: Maker.DAO revisited

As was briefly mentioned in the section on Maker.DAO in the Stablecoins section, in Maker, generating stablecoins is considered a loan from the system which is priced at a rate called stability fee set by the protocol governance. Indeed, it has all the functional characteristics of a lending system: one places collateral and receives a tradable token, which can later be repaid with interest in order to recover the collateral.

As a loan tool, Maker construction stands out in the way that the technical lender (the system itself) does not own any capital. DAI lent is minted on the spot and is destroyed on repayment. The interest is paid in a different token (volatile governance token MKR), which is also destroyed[13]. First of all, this means that the liquidity depth available on Maker is limited only by its debt ceiling (governance parameter that binds the maximum amount of DAI possible in existence). One reason for that is to manage the risks of destabilization: a bigger debt ceiling enables bigger potential jumps in supply, affecting the price accordingly. Correcting $100 million of DAI underpriced by $0.02 is a different challenge from covering the same $0.02 difference on a supply of $400 million. The tools are complicated in themselves, and on top of that social consensus has to be achieved on how and when to use them.

Another corollary of minting at will—at launch, it solves the chicken and egg problem. One side (the collateral) is brought by the prospective lender, the other side (the borrowed asset) is already available.

Liquidity-based loans

The next big step in lending required a subtle shift in architecture which was first implemented in **Compound**: moving to an account-based system. At functional level, it performs quite similarly: one can put in collateral and borrow assets up to a certain fraction of value of the collateral. However, instead of creating CDPs holding collateral and tracking debt per individual loans, the protocol tracks the deposits and loans of a user in an aggregated manner, associated with the user's Ethereum address. The protocol has multiple pools with different assets. All collateral and all debts are summed up for the user in ETH(based on their oracle-supplied price feeds) across all pools, and the collateralization ratio applies to these aggregate numbers.

When a user puts in assets into a pool, they essentially become a liquidity provider in this pool, and are eligible for a fraction of the interest rate paid by the borrowers from that pool. This means that placed collateral creates passive income for the borrower, as long as they are not liquidated. It also means that one can simply supply an asset into a pool for passive income, without taking out a loan. Participation in the protocol pools is tracked with cTokens: a hypothetical liquidity pool of NVM would have cNVM as its pool participation token, denoting the amount of NVM the user has placed into the protocol.

The multitude of assets poses a challenge. While in no-lender systems the interest rate is set through governance, in two-sided designs, this would introduce two problems:

1. There are possibly as many assets as there are ERC20 tokens—no amount of governance would be able to properly manage rates for all of them;
2. The effective rates have a practical impact on liquidity, as they incentivize capital owners to provide their liquidity for borrowing.

[13] This is the way to incentivize participation in governance, as the supply of MKR is reduced by debt repayments, pushing the price up. MKR can only be minted in an emergency event when all other means to stabilize DAI fail, as a last attempt to auction it out and recapitalize the system at the expense of the (evidently, unfit for duty) governance participants.

Because of this, rates in Compound are variable and are determined algorithmically. In each pool, the rate is determined based on the utilization of that pool, i.e. the fraction of the total pool supply that has been borrowed. This means that both borrower interest rate and lender yield increase with increasing demand for an asset, and decrease when more supply is added. Depending on market conditions, the yield from a borrower's collateral can actually be larger than the interest that they pay on their loan.

As usual, when the user's collateralization ratio is not maintained, they are liquidated. Liquidation works slightly differently in account-based models: whenever the sum of the user's total outstanding debt across all assets exceeds a certain threshold, calculated from the value of the total collateral placed by the user across all assets and the minimal collateralization ratio,—liquidators can buy any asset of their choosing from the user's collateral pool, until the account is recapitalized to a proper state, with some safety margin. At liquidation, the protocol burns the borrower's cTokens that are being liquidated and returns the corresponding underlying asset tokens to the liquidator.

CASE STUDY

Compound

Account-based collateral: Compound

Compound protocol was proposed in the spring of 2018. It was defined as *a protocol for money markets: pools of tokens with algorithmically derived interest rates, based on the supply and demand for the token*[14]. The concept of liquidity-based borrowing pools and the shift to account-based design were both introduced by the protocol.

CASE STUDY

AAVE

Stable and variable rate loans: Aave

Aave is similar to Compound, but with some additional features. It was seemingly the first protocol to introduce flash loans, which have now become a staple in DeFi. The second addition of Aave are so-called stable rates. The purpose of stable rates is to provide a predictable pricing policy for borrowers that allows informed financial planning. At loan creation, the user is able to choose between variable rate (recalculated dynamically to follow current utilization) and stable rates (bigger than variable rate at any particular utilization point, but not changing with utilization shifts). Stable rate is fixed based on utilization ratio at the time when the loan is taken and is only adjusted in two cases: down, if the rate is more than 20% bigger than the current rate, and up, if a critical utilization rate of 95% is reached.

CASE STUDY

Equilibrium

Multi-currency loans with aggregated rates: Equilibrium

Another approach proposed recently is setting the rates based on the volatility of a portfolio rather than dynamic and unpredictable pool utilization ratios in the protocol. Other parameters considered are collateral value and the current loan collateralization ratio, the rates are still dynamic. In this model, the rates borrowers pay and liquidity providers receive depend on the overall market conditions rather than protocol-specific circumstances. Equilibrium uses traditional models to estimate the system risk and expected volatility and adjusts rates based on them.

[14] R. Leshner, G. Hayes, Compound whitepaper v. 0.3 (Wayback Machine).

Position-based and account-based designs and debt markets

Buying and selling debt is a well-known instrument in traditional finance, which takes many forms and has its uses. In DeFi, viability of secondary markets for debt positions is uncertain: inherent overcollateralization, high liquidity of tokenized assets, and the general possibility to close the loan instantly without further capital requirements (selling off part of the collateral to repay the debt) only leave a very niche use cases that do not currently seem to attract much interest.

In theory, it could be viable in the following scenario. A loan has accumulated a substantial amount of interest (of course, still under the collateralization ratio), so that most of the collateral would be used up, if the owner were to close it. There might be counterparties willing to take a speculative position in the collateral asset, buying it from the debt owner (together with the debt) at a price point below the current market price for collateral, but paying to the debt owner more than what can be recovered after selling off the collateral. The speculative bet, therefore, would be that the collateral will appreciate quicker than the interest rate, paying off the debt at a later time. This is hard to imagine with the present-day market dynamics, but it is not impossible.

Technologically, the ability to resell a debt position comes naturally to position-based designs (CDPs), and is a bit more complicated for account-based designs.

For the former, the unit of operation is a loan, which is characterized by collateral size and the amount of debt. A debt position is a separate entity, one blockchain account can own multiple positions, and positions themselves could be passed between accounts, theoretically allowing secondary markets for open debt positions without much code.

For the latter, the unit is a blockchain address, and all of its placed collateral and outstanding debt are aggregated to form a single number—total debt—as well as current aggregate collateralization ratio and allowance. This approach provides a streamlined UX (no need to manage and watch multiple debt positions across different assets), but does not offer a ready interface for secondary markets. It is still possible to build, by implementing a smart contract with transferable "ownership" that would act as the account for the loan instrument and as a tradable entity to be plugged into a secondary market, but no solutions to date offer this functionality.

The common denominator for every general-purpose collateralized lending solution is the concept of system debt and total collateralization ratio. It is important for both emission-based and liquidity-based designs, albeit for different reasons. For the former, the value of the lent asset that the protocol mints ultimately falls back on the excessive collateral that could be liquidated to support the price. For the latter, it represents the ability to repay every liquidity provider with interest.

"Reserve banking" (there is none) and total supply in asset pool

On August, 14, 2020, the total amount of DAI supplied by Compound was $1.123 billion, while the total amount of DAI in existence (as minted by Maker.DAO and not yet burned) was $428 million. This dynamic could be observed for several weeks, giving the wrong impression of reserve banking happening in the lending instrument: how could it give away more currency than there exists, unless it used a fractional model that put their liquidity providers at risk?

The answer to that has two parts: "how" and "why." There is no reserve banking in Compound, every dollar worth of value given out is collateralized by some combination of assets. The system may not be currently able to return its entire supply of DAI, specifically, but it is solvent on the level of system debt, i.e. if it sells its entire collateral base (assuming low slippage and temporary price stability), it will

be able to buy all of the assets it needs to repay every liquidity provider, with interest, liquidating in the process the lending positions of the borrowers. Total system debt and collateralization level represents the safety margin to perform that operation.

The perceived effect of reserve banking is, in fact, historical: on June, 15, Compound started distributing its governance token to lenders and borrowers in the protocol, proportional to their participation in the pools. One strategy to "farm" the governance token would be buying a particular asset, supplying it into a pool, then re-borrowing it from the same pool, repeated in a cycle. The collateralization ratio entails that this cycle has diminishing power, unless it is collateralized by another asset. The resulting pool participation tokens produce governance tokens, and the borrowing APY has to be measured against the value of the governance tokens that can be received, based on participation of other market agents in the same pool. Stablecoins are the natural choice for this operation, since they are not volatile and therefore only the volatility of the collateral can affect the ratio of collateralization and the possibility of liquidation.

3.2.2 Encapsulated loans: margin trading

Trading with leverage is a ubiquitous technique in traditional finance that allows a trader to add borrowed assets to their own and enter a position. Leverage allows increased exposure to an asset with only a limited amount of capital, which enables customizing risk and returns to the trader's particular risk profile.

It is possible to leverage with ordinary collateralized loans using the re-borrowing technique, but the process itself is cumbersome[15], and leverage is limited by the required collateralization ratio. The 150% collateralization requirement translates to a theoretical leverage limit of 3x, and in practice, no more than 2.5x can be achieved without the risk of immediate liquidation. This is why margin trading protocols were one of the long-awaited features in DeFi since its inception.

One might find it jarring to find the discussion of margin trading, not in the respective "trading" section, but in the section pertaining to lending and debt. However, the choice is entirely intentional—margin trading has an intrinsic connection to the debt market, since acquiring leverage is equivalent to taking out a special kind of loan.

This is especially notable in DeFi—trading applications do not discriminate based on the asset type, unlike exchanges in traditional finance, so margin trading instruments aren't focused on trading per-se (this can be done on any DEX), but on facilitating a money market for leverage and generating tokens that represent leveraged positions.

We will review the two most prominent margin trading protocols—**Fulcrum** and **dYdX**.

[15] Although this is somewhat remedied by advanced user-facing interfaces—see InstaDapp.

CASE STUDY

Fulcrum is a margin trading and lending protocol that is a part of the **bZx** network. On the lending side, Fulcrum functions similarly to collateralized loan protocols, such as Compound—liquidity providers are able to supply assets within tokenized pools and earn interest, which is determined by the utilization of a particular pool.

However, the borrowing mechanics are greatly extended. While it is possible to borrow exactly the same way as in Compound (i.e.,i.e. placing collateral and drawing debt up to a certain fraction of collateral value), Fulcrum supports undercollateralized positions, which exactly coincides with trading with leverage—one can bring, e.g., only 10% of the capital, borrow the rest, and enter the position with the total sum.

Unlike overcollateralized positions, undercollateralized positions remain in the custody of the protocol. The borrower does not get the borrowed asset sent to their account—instead, the asset is immediately swapped. When entering a long position, the borrowed asset is usually a stablecoin, which is then swapped to the asset that the borrower wants to be exposed to and held by the contract. Conversely, when shorting an asset, it is borrowed from the respective pool and swapped to a stablecoin.

Undercollateralized positions that remain within the protocol also have more relaxed liquidation mechanics—generally, the position will be liquidated when there is a risk that it will go lower than debt plus interest. To allow liquidating agents to react, there is a small safety margin above that.

Typically, margin trading pools have higher interest rates than overcollateralized lending pools, because the usage of funds is immediately ensured by the speculators' demand for leverage. At the same time, lending in instrument-based pools can be riskier, because the safety margins are generally smaller compared to overcollateralization, e.g., with a leverage of 10x a sudden price drop of 10% (considering that the network may already be congested from liquidations in other protocols, it may not be liquidated in time) will burn all of the trader's money, leading to the system being unable to repay lenders in full.

To protect against black swan events, Fulcrum keeps track of system debt (positions becoming undercapitalized due to sudden price movements) and allocates part of the fees into an insurance fund, which is used to pay out this debt.

CASE STUDY

dYdX works similarly to Fulcrum—it supports a number of asset pools where the traders can borrow from to margin trade.

The difference, however, is that Fulcrum uses Kyber for asset swapping and entering positions, while dYdX is built on 0x. When a trader enters a position, the borrowed funds are taken from the pool by the contract, and then an order for the total sum is routed through **0x**.

This means that dYdX has a more traditional order-based structure compared to **Fulcrum**, and all the advantages and disadvantages of 0x-based DEXes.

PATTERN

Flash loans

Flash loans are a very recent mechanism that can be arguably considered one of the most important developments in DeFi.

Flash loans are loans that span the length of a single network transaction. They do not require any form of collateral, but only that the full sum (plus small interest) is returned by the end of the transaction. If the sum is not returned, the transaction simply reverts entirely, undoing all the changes that the transaction tried to make. This property allows the loan provider to guarantee that the funds are always returned (in essence, they are not given out unless they are returned 'later').

Flash loans demonstrate how powerful DeFi can truly be—purely through algorithmic enforcement flash loans do not require loan negotiation or any kind of collateral. Capital essentially comes free as long as the operation it was used in is at least a bit profitable, and if a borrower miscalculated or got unlucky, everything is rolled back, barring network fees. This is a kind of a mechanism that is hard to imagine in traditional finance, because after giving out a loan (even a very short-lived one) repayment can only be enforced institutionally.

Flash loans solve in DeFi a long-standing inequality in traditional finance that leads to path dependency and general inefficiency of capital use—that is, capital prerequirements. To engage with the markets and earn, one has to either seek a counterparty to borrow funds from, or be fortunate enough to receive windfall. A huge part of the world's population is barred from both of those opportunities.

Flash loans do not have capital pre-requirements, and only demand a level of expertise to use them, thus being a large step towards making finance more meritocratic. Due to the atomicity requirement, however, the number of use cases for flash loans is currently limited. Flash loans do not allow one to enter positions—however, they can be used to engage with risk-free transactions, such as arbitrage, liquidations, or staking. Even when one starts off with no capital, they have multiple ways to build their starting capital that can then be used in other transactions.

There are also several important convenience and capital efficiency functions of flash loans, such as collateral or protocol swapping in collateralized loans. Flash loans allow one to atomically take a flash loan to repay a collateralized loan, take out collateral, swap it or move it to another debt position, and then re-borrow the funds and repay the flash loan. This allows for much more flexibility in optimizing interest rates than was previously possible.

Flash loans were also used in several prominent attacks on DeFi protocols, although this fact only reinforces their usefulness—all of the attacks could be performed without flash loans, by simply using pre-existing capital, but it was much more convenient to use flash loans instead of sourcing funds.

3.3 Risk management and hedging

Hedging is an important concept in finance, as traders often need instruments that allow them to fine tune risk against rewards, and businesses need to limit exposure. Insurance has a business or a trader sacrifice a small part of their profits ("insurance premium") to the underwriter that agrees to take on their risks, whether that is a market black swan event, a natural catastrophe, or counterparty default.

DeFi has tried to implement various hedging techniques, including mutuals, options, and prediction markets, with varying degrees of success.

3.3.1 Insurance Mutual Funds

Insurance mutual funds pool assets from investors to create a fund that promises insurance against catastrophic events, in return for premiums. Insurance funds try to predict the probabilities of insurance claims being fulfilled in any given period, and factor that into their business model by adjusting the premiums.

However, payouts after insurance-covered events are not always guaranteed due to fraud on both sides—insurance buyers try to fake catastrophic events for a big payout, while insurance companies may try to bend or constrain the definition of what constitutes an insurance-covered event to avoid paying out.

Insurance funds are a mechanism that ultimately falls back on social consensus (or if that fails, a judicial system, which does not exist in the blockchain), as due to fraud the assessment of claims is subjective and can be manipulated. Social consensus is susceptible to collusion and other types of corruption. It is evident from the example of Nexus Mutual that such systems function with a lot of friction when put into the context of DeFi.

CASE STUDY

Nexus Mutual

The best known representation of an insurance fund in DeFi is **Nexus Mutual**—a cooperative insurance pool that covers financial loss due to smart contract failure. Nexus Mutual works through a bonding curve—the price of the NXM tokens purchased from the contract increases with the amount of funds in the contract, divided by the sum of all taken out policies. NXM tokens are used for a wide array of actions—they are needed to post claims, participate in claims assessment, or participate in governance.

The fact that NXM holders assess claims is actually quite important—the value of NXM tokens is directly proportional to the amount of funds in the insurance pool, so paying out a claim will reduce their price, at least, in the short term. This leads to very skewed incentives for token holders, as they have to choose between the short-term value of their tokens and long-term health of the project. The latter, however, is less predictable, and the token holders are naturally inclined towards refusing to pay out claims.

To date, the only significant payout by the mutual was 31,000 DAI payout after the **bZx** attack, despite about $3M of coverage purchased in total. More than that, the first claim on the bZx attack were almost unilaterally rejected, as well as claims after the Maker zero-bid liquidation exploit. This generally seems to be aligned with the skewed incentives of the mechanism.

3.3.2 Options

Options are derivative contracts that give one the right, but not the obligation, to buy (call options) or sell (put options) an asset at a pre-agreed price level, before the agreed-upon expiry time. Options are used in a variety of scenarios: they can be used by business owners to limit their price risk on input materials, or by traders to protect against a black swan event. The writers (option sellers) take a small premium at the sale (which can be considered an insurance premium) for protecting their counterparty against upward or downward price swings.

Options are not only used for insurance, but are widely recognized as a reliable oracle of risk—if the market expects the asset's price to be volatile at some time in the future, then options that exercise (expire) at that time will be become more expensive, as more market agents will seek to protect themselves from risk.

Options have much less counterparty risk than traditional insurance funds, as after option purchase the writer is locked into providing coverage, regardless of the reason for exercising. For example, **Nexus Mutual** declined the first claim after the **bZx** attack, as it was considered to be an oracle attack at that time, which isn't covered by the policy. Conversely, buying a put option to sell 1 DAI for 1 USDC will always protect against DAI breaking the peg, whether it was due to a code bug, an oracle attack, or a particularly catastrophic market event. This makes options a natural choice for hedging in DeFi.

Within DeFi, options have been implemented only quite recently, and, as such, are mainly represented by a single protocol—Opyn.

CASE STUDY

opyn

Opyn is a protocol for collateralized option writing. Opyn allows writers to put in collateral and mint option tokens (oTokens). The writers can then sell those tokens on the open market to those who require coverage.

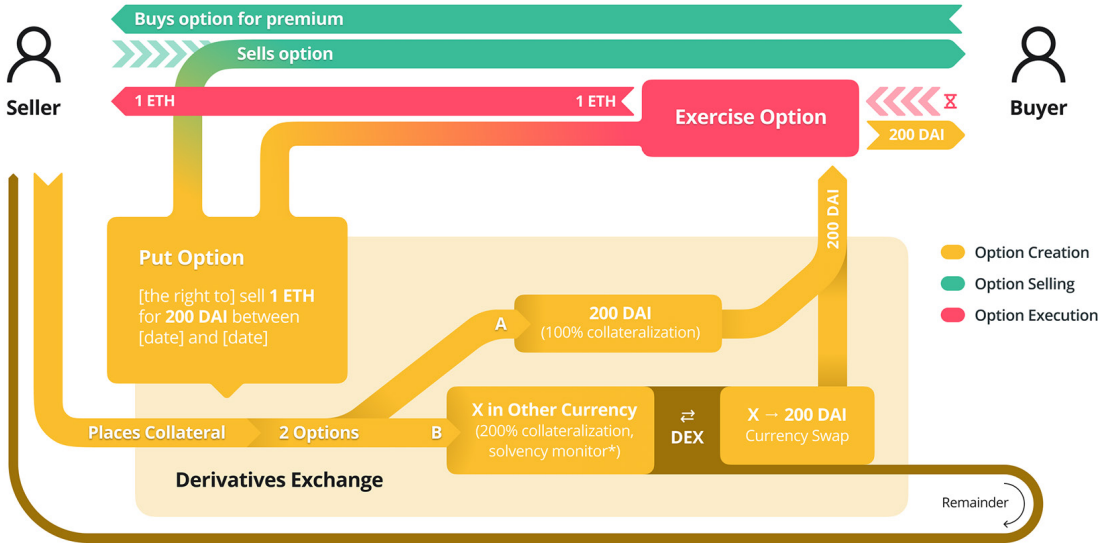
If the writer puts in collateral in the currency that they have to pay out at exercise (underlying currency), they only have to be 100% collateralized, but they also can put in collateral in other currencies. For example, if they collateralize DAI to USDC options with ETH, these options will pay out the required USDC amount in ETH at the current market price (which can immediately be converted to USDC through, e.g., **Uniswap**).

If the options are collateralized with currency other than the underlying one, they also have to be overcollateralized in the same way as Maker or Compound (since the underwriter can become insolvent if collateral depreciates), and can be liquidated by buying oTokens from the market and redeeming them.

All oTokens from the same series (e.g., same base and underlying currency, exercise time, and other parameters) are fungible, and the collateral is pooled, so liquidation of a particular underwriter does not influence oToken buyers. The underwriter themselves can "unwind" (exit position) by purchasing oTokens from their series, and redeeming them to free the collateral.

oToken owners can then redeem oTokens before expiry to buy or sell the underlying asset at the strike price.

Opyn oTokens can be used for a number of use-cases—such as covering one's collateral against liquidation, protecting against protocol hacks (by purchasing options for protocol tokens, such as Compound's cTokens), or simply reducing downward exposure for volatile assets.



* For explanation please refer to the Pattern: Overcollateralization and market liquidation

Decentralized put option lifecycle

3.3.3 Prediction markets

Prediction markets are marketplaces where individuals can bet on certain events occurring. They are often used to hedge against black swan events—payouts are usually reversely proportional to the event probability, so winning on a low-probability black swan event leads to large payouts that can cover losses incurred on other instruments.

Traditional prediction markets are most often represented by betting exchanges that facilitate two or more counterparties entering into a contract that is settled when the target event of the bet is resolved. This mechanism is adapted to the DeFi ecosystem by its most prominent prediction market project, Augur, which implements a traditional betting exchange with an extensive outcome reporting mechanism to prevent malicious reporting.

CASE STUDY

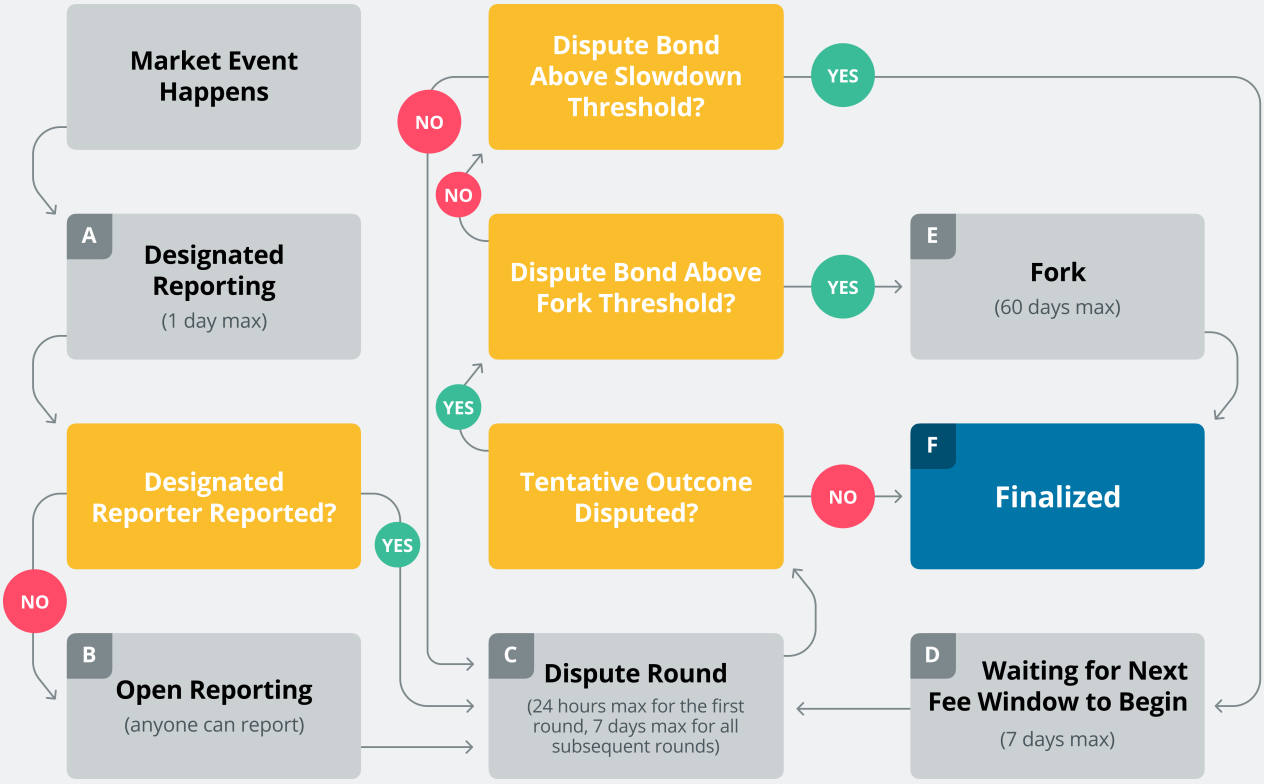
augur

Augur is a decentralized prediction market protocol that functions similarly to traditional betting exchanges. Betting exchanges trade contracts that resemble cash-settled futures, but, unlike futures, the settlement payouts are "all-or-nothing". The exchange keeps an order book that contains the outcome, the price, and the quantity. Once the order book accumulates enough orders to comprise a full set of outcomes with a total price of 1 (representing probability 1 of one of the mutually exclusive outcomes happening), these orders enter a contract.

> **Example.** *There's a market with mutually exclusive outcomes A and B. There's an order on A with Q = 2 and P = 0.7, and an order on B with Q = 3 and P = 0.3. They will be matched into a contract with Q = 2. The contract settles on a set date (usually the date when the outcome can be concretely determined), and parties can exit their position by selling it to someone else, to prevent further exposure, or lock profits. The order book determines the "prices" for each outcome, which can be used as a measure of the markets' confidence in the outcome.*

In **Augur**, the in-contract matching engine tracks orders for all outcomes and matches them when it finds a complete set of shares (i.e. a set of orders for all outcomes that sum up to 1). For each of these shares, an ERC777 token is created and sent to respective parties. Then, these shares can be traded freely in Augur itself, or on secondary markets. Shares essentially represent positions in a contract in prediction markets of traditional finance.

Reporting on **Augur** is where it gets interesting. Augur has its own service token called REP, which has to be staked in the reporting system. Reporting has several layers of accountability, and if a particular market continuously fails to resolve, more severe actions are taken to converge to the true outcome, and punish adversaries that seek to push a false outcome.



Augur reporting system flowchart. Source: Augur whitepaper.

There are three distinct conditions of a market that differ by severity:

- **Normal reporting**
  Most of the markets resolve without issue. When the market is created, the creator sets the reporting date and appoints a designated reporter, and stakes a small amount of REP as a bond. If the designated reporter does not report within 24 hours of the set reporting date, the system moves to open reporting,

where anyone can report on the outcome and take the bond for themselves. As this is essentially risk-free arbitrage, it is expected that the report will be delivered by some free market agent rather quickly. Then, the first dispute round starts and lasts for 24 hours. If the outcome is not disputed within 24 hours, the market is finalized and settled. When a market is finalized, the reporter gets a REP-denominated fee.

- **Disputing**
  If the reported outcome was disputed, the disputing party must stake an amount of REP that depends on the total value put in the market. This amount is staked on the alternative outcome. There may be several rounds of disputes, however, the stake amount always increases from round to round. The dispute amount can be crowdsourced, and if the total staked REP in the market is 0.02% of the total amount, there is a delay imposed to allow the community to coordinate and dispute a false outcome. If the total staked amount constitutes 2.5% of total REP, the market moves to the forking stage.
- **Forking**
  When the forking stage condition (2.5% of total REP) is met, **Augur** pauses all current markets (the shares from these markets are still tradeable, however, they cannot be resolved) and creates several universes for each outcome, as well as a universe for the "Invalid" outcome. Within 60 days, all markets and REP holders must choose which universe they will migrate too. At the same time, REP staked for outcomes of the forking market are forced to migrate to a universe corresponding to their outcome.

After the forking stage ends, the universes cannot interact with each other at all, and REP in each universe can be considered a separate token. If a wealthy adversary wants to break some market by disputing and staking REP for a false outcome, their tokens will be transferred into a universe that corresponds to this outcome. The free REP holders that have a choice will naturally want to migrate to the universe that will end up with most REP—since it will be the most secure and the most utilized one, the REP in this universe will be the most valuable. Then they have three choices:

1. **Migrate to the universe of the adversary**—they will not want to do that, as it is known for a fact that there is a wealthy adversary that manipulates markets. Even if the adversary bribes those who migrate to their universe, the REP holders risk to end up in a universe which no one uses and their tokens lose value, so they cannot predict their loss or profit after taking the bribe.
2. **Migrate to the universe with the true outcome**—a natural Schelling point,[16] as it is special among all universes, beside the adversary's universe.
3. **Migrate to the universe with a false outcome, but without the adversary**—these universes are not "special" in any way, and are thus not Schelling points. As such, there's also a risk of being stranded after making the choice that is not obvious to everyone else.

[16] A Schelling point (wiki) is, informally, the most "natural-looking" choice in a given situation, one that is tempting as the default option. It is used in game theory and mechanism design to denote the option that can be arrived at by most of the participants without coordination or communication. In the context of reporting outcomes with more than two choices, the truth (as long as it is strictly defined) is often a Schelling point: agents reporting truth will have the same answers, while uncoordinated attackers will report different untruths. This construction implicitly assumes some form of Sybil resistance (wiki) and penalties for reporting an untruth.

The users cannot choose to remain in the original universe—after the forking stage, it is locked, and their REP cannot be used, losing all value. Thus, the forking stage allows the system to resolve to a correct outcome even if a wealthy adversary tries to force some market to a false outcome. At the same time, forking is extremely disruptive, and should only be used in extreme cases.

The complexity of the reporting system is the testament of the difficulty of challenges faced when designing a system within a decentralized setting that relies on external data. This has been one of the primary areas of research in DeFi, with some exciting recent developments being reviewed in the oracle pattern section.

INTERLUDE Oracles

Oracles are on-chain services that are able to report external data to blockchains, which are generally oblivious to the outside world. Oracles power a multitude of DeFi solutions, in most cases providing price feeds and reporting real world events (e.g., for prediction markets).

Price data plays an integral role in a lot of protocols and mechanisms, driving automated decision-making, and, thus, can be used for powerful attacks when it is compromised. For this reason, it is extremely important to make oracles as robust as possible.

While, admittedly, the journey is not yet over, significant strides have been made towards the goal of reinforcing oracles for use in DeFi.

External (off-chain) data feeds

Oracles for external data in their base form are usually implemented in one of three ways:

- A centralized oracle that has a single party providing data. Provable is the most prominent example in this category, and also utilizes authenticity certificates to ensure that retrieved data isn't tampered with.
- Some solutions use consensus (for example, some variation of BFT consensus) of several nodes to provide data. The nodes may be pre-approved (although this arguably moves the mechanism into the first category), or there may be a different mechanism of node selection, such as having them stake some token.
- Schelling coin-based systems aggregate values from many individual peers, and then reward those that are close to the aggregate (usually median) and punish those who are far from it. These systems, however, can be gamed through Sybil attacks, so some sort of staking or whitelisting must be implemented.

After constructing initial base oracles, more robust data feeds can be constructed through aggregation. One example of such an approach is Chainlink, which is used by a multitude of projects. Chainlink receives the same data feed from multiple sources through a commit-reveal scheme—a scheme that prevents adversaries from knowing supplied values in advance, or sources copying answers from each other—and constructs and aggregate, which it then supplies to consumers as a final feed.

Internal (on-chain) data feeds

There are also advancements in the realm of on-chain price feeds. On-chain price feeds are generated by on-chain decentralized exchange services. The validity of data is naturally ensured, so these price feeds, if sufficiently robust, could be used to replace external feeds, or, at least, greatly strengthen them. However, decentralized exchanges suffer from small liquidity, which allows wealthy adversaries to easily skew prices by placing large orders. Fortunately, there are great strides towards remedying this—Uniswap announced their own oracle solution for their 2.0 release which increases the difficulty of manipulation prices, even with low liquidity.

In Uniswap 1.0, the current price is always reported by the price feed. This leads to a possibility of manipulations in the middle of the block. An example would be a derivative settling when ETH/DAI price reaches a certain level. An attacker can within a span of a single transaction buy a lot of DAI on Uniswap to inflate the price, trigger derivative settlement, and sell DAI back to recover the exact same amount of ETH (in fact, with flash loans they don't even need to own any ETH).

In Uniswap 2.0 the exchange rate for each pair is measured at the beginning of each block (as the price after the last order of the previous block) and only that price is reported by the oracle. With the new system in place, the derivative would only receive the price at the beginning of the block, that does not include the manipulation.

Even with that, there's still a possibility that the attacker performs the manipulation between blocks. While this is generally much more costly than an atomic transaction, due to the attacker competing with arbitrageurs to get their ETH back, the profit from manipulation may still be worth it. To further counteract manipulation, Uniswap 2.0 introduces TWAPs (Time-weighted average prices). TWAPs track two cumulative sums—the cumulatives of price and of time.

The cumulative of time is simply incremented by the time that it took to mine a block. The cumulative of price is incremented by the current block price feed, multiplied by the block mining time. These cumulatives are computed each block, and saved within the contract. Now, consumers of the data feed can simply pick a window that suits them, and divide the price cumulative change by the time cumulative change over the period, getting the average price.

The best thing about this mechanism is that it is very flexible. Usually, moving averages work well enough in stable markets, but are too slow to react during extreme events, which may lead to delayed liquidations, leveraged positions not closing properly, and other disastrous consequences for instruments. But TWAPs provide a large array of public information, and thus the window can be configured dynamically—when an instrument detects a possible instability in the market, it can decrease the window size to be more responsive, while in a more stable market the window can be enlarged to defend against manipulation. Overall, the cost of manipulating the price in a Uniswap 2.0 price feed scales linearly with the size of TWAP window and liquidity, which allows to make a cost of attack expensive in most cases.

The recent project Mooniswap also introduced VWAPs—volume-weighted averages. Mechanically they work similarly to TWAPs, but the time increment is replaced with a volume increment. VWAPs are robust against skewed reporting even in the short term—since the attacker would have to perform a very large transaction to skew the price, losing a lot of funds to slippage.

These mechanisms are an important step towards reducing the influence of external price oracles on DeFi, which will increase the robustness of the ecosystem as a whole.

3.4 Infrastructure and utility

A conversation about DeFi cannot be complete without discussing numerous utility solutions that are used as the infrastructure layer for actual instruments, or bridge the gap between complex on-chain interactions and ordinary users.

In the early days of DeFi, there were only a small number of instruments with limited liquidity, with the only access point being direct interaction through basic Dapps provided by instruments themselves, or through on-chain transactions. As the number of solutions and liquidity grew, the need for richer modes of interaction and analysis arose.

DeFi is naturally very observable, and, thus, can potentially have very low information asymmetry, leading to a more efficient market than traditional finance. More than that, due to its open nature, DeFi insights have much shorter impact cycles: the aggregated information can be used immediately to optimize various operations. But all of this requires proper tools that observe the ecosystem in an automated manner and provide a user interface to interact with results.

Tools that appeared in 2019–2020 can be roughly categorized into 4 classes: data sourcing, entry-level tools, professional connector tools, and optimization tools.

3.4.1 Data sourcing

Analytical tools in DeFi provide market observability by aggregating on-chain data into readable metrics, such as market activity, ROI, amount of locked collateral, etc.

While getting access to something like price or volume daily data for tokens is relatively simple, the true power of DeFi analytics often lies in comparing and combining metrics from individual protocols, which requires free access to data to build adequate processing pipelines.

While there are many analytics dashboards in DeFi, the truly empowering projects are those that provide convenient direct access to on-chain data for the users' processing needs. This is complicated by large upkeep costs of synchronizing blockchain nodes and continuously parsing blockchain state updates. Therefore, we focus on projects that provide blockchain indexing and aggregation services, and expose the resulting datasets to users.

CASE STUDY

The Graph

The Graph is an infrastructure layer project that facilitates a network of nodes indexing data from on-chain contracts, processing it based on some schema, and then serving the resulting dataset to external consumers.

Any user can create a schema, or "manifest", to indicate which on-chain contracts to track, which events and variables to watch for, and how to aggregate the resulting individual data points into concrete metrics. The result is a document-oriented database (called "subgraph") that can be queried by anyone using GraphQL[17] from indexing nodes.

Currently The Graph indexing nodes function as altruistic volunteers, providing access to 20+ "official" subgraphs and many more community subgraphs. Many of those indexing nodes belong to major DeFi projects teams, such as Uniswap or ENS. However, there are also plans to implement an incentive layer to make the project feasible in the long run.

Many official Dapps for protocols (such as Uniswap, Bancor or Compound) source data from The Graph.

[17] GraphQL is a query language for document-oriented database structure, similar to, e.g., MongoDB query language.

CASE STUDY

Pocket network is a decentralized alternative to third-party blockchain node providers, such as Infura.

Pocket is able to process requests to the node interfaces of many protocols by routing those requests to a distributed network of data providers. The interesting feature is that applications can customize the number of service providers, as well as the trade-off between security and speed. The application may choose to have all service nodes to process the same request, preventing tampering, or have them all processing different requests in parallel for increased scalability.

Pocket's use cases lie in the regions not covered by The Graph. While the latter supports extensive aggregation capabilities, it requires the particular schema to be supported. Pocket is able to process arbitrary requests, as long as they adhere to a node interface.

3.4.2 Retail-oriented products

Recently, the number of wallets adding DeFi functionality has grown considerably. However, most of those projects combine a wallet and a dashboard that allows basic interaction with major protocols, such as Uniswap, Compound, etc.

This leads to a sort of identity crisis—for entry-level users, such interfaces are too sophisticated, since they do not necessarily understand the basics of protocols included into the dashboard.

Therefore, the few projects that greatly simplify interaction with DeFi for new users are especially notable and are the focus of this section.

CASE STUDY

Dharma is a savings-focused ETH/DAI wallet.

The main feature of Dharma is that it automatically places all funds in the wallet into Compound, so funds continuously earn interest without any input from the user. This is especially useful for entry-level users as they are not exposed to any Compound complexity whatsoever.

Dharma keeps its own on-ramping gateway, allowing users to purchase ETH or DAI with consumer payment services—credit/debit cards, Apple Pay, etc. Another interesting feature is that Dharma allows adding users into a contact book based on their Twitter handle and sending assets to them.

Dharma fully abstracts the majority of DeFi complexity and jargon from the end user, resulting in a solution that is well-suited to be an entry point into DeFi for newcomers, which, in the end, results in more capital and adoption for the entire sphere.

3.4.3 Connector tools

While products for newcomers are important for bringing adoption to the ecosystem, there is also a high demand for advanced features from existing users.

Users that are sufficiently well-versed in DeFi often seek to implement advanced strategies to improve their ROI or hedge risks. These strategies require advanced techniques for efficiently moving capital among different instruments, assets, and protocols.

This section focuses on wallets and dashboards that expose access to one or several major protocols within a single interface, while also providing users advanced features (such as leverage with re-borrowing) on handling their capital.

CASE STUDY

InstaDapp is a dashboard for advanced interactions with lending protocols.

InstaDapp connects to Maker and Compound and provides a streamlined interface for handling the user's collateralized loans. InstaDapp has three advanced lending features—leveraging, unwinding, and protocol swapping.

Leveraging refers to the pattern "leveraged positions with reborrowing". However, instead of several transactions, as described in the respective section, InstaDapp compresses all required actions into one transaction, greatly simplifying the interaction.

Unwinding allows the user to partially repay their debt even without access to the borrowed asset, by withdrawing the unused part of collateral, swapping it to the borrowed asset, and repaying. As with leveraging, this typically must be done iteratively through several transactions, but InstaDapp combines all interactions into a single transaction—this allows to repay the entire debt with this method within a scope of one transaction.

Finally, protocol swapping allows the user to swap their position between Maker and Compound—arbitraging between interest rates if there is too big of a difference.

CASE STUDY

Dedge is another advanced tool for Compound that allows swapping both collateral and debt assets on a debt position, without necessarily having direct access to those assets.

Swapping collateral and debt allows efficient risk management on debt positions. A debt position where both collateral and debt are volatile can be secured against liquidation in a falling market by swapping collateral to some stablecoin. Conversely, a debt position that is already short against a volatile asset (i.e., stable collateral and volatile debt) can be swapped to protect against a rising volatile asset.

The swapping operations are very efficient because they use flash loans—each collateral/debt swap consists only of taking out a flash loan to repay the collateralized loan, and then repaying the flash loans while the new collateralized loan is created.

CASE STUDY

Zapper.fi

Zapper is a tool for liquidity providers that allows them to efficiently move liquidity between various protocols and pools to quickly adapt to changing market conditions.

Moving liquidity between pools is usually a fairly complex operation—one has to redeem liquidity tokens to take out liquidity, then exchange resulting assets (which will be returned in some arbitrary proportion) to the assets required in the new pool, and then submit them. Manually, this can require many transactions, especially in pools with many assets, such as in Balancer. Zapper combines all of these required transactions into one, greatly simplifying the experience.

This also extends to on-ramping as well—Zapper allows submitting a single asset, and automatically exchanges it to pooled assets in required proportions.

3.4.4 Optimization tools

The class of tools that only recently started to gain traction is tools that implement automated decision making based on gathered DeFi data to improve portfolio performance—be it optimization of fees, slippage, or ROI.

When we discuss actionability of DeFi data, this is exactly what we mean—non-custodial decentralized apps that implement complex strategies to improve one's portfolio performance, sometimes even without any user input required.

CASE STUDY

1inch

1inch is a DEX aggregator that splits exchange orders among multiple trading protocols to optimize exchange rates and fees.

In the simplest case, the order will be executed on a single exchange that currently has the best rate—this usually happens for small orders, where slippage is insignificant. If the order is large, however, 1inch will select the distribution among available exchanges with the best overall outcome, and will execute all trades atomically, sending acquired tokens to the user.

1inch can greatly improve the performance of large-scale traders, but it is also useful for dApp developers—using its API for tokens swaps allows the developer to focus on the concrete application functionality, without considering slippage handling.

CASE STUDY

Yearn.finance is a tokenized pool project that aims to optimize APR for borrowing liquidity providers.

Yearn allows liquidity providers to supply stablecoins into the pool and then dynamically reallocates that liquidity among Compound, Fulcrum, Aave and dYdX, depending on where the APR is highest for each stablecoin.

Essentially, Yearn provides a layer of abstraction on top of lending pools that simplifies the choice of strategy for liquidity providers—instead of considering various borrowing pools separately, they only have to compare Yearn (which represents the best possible lending APR) to pools with other business models—fee-accruing pools, SNX staking, etc.

CASE STUDY

Dca.land is a dollar-cost averaging tool for ETH.

Dollar-cost averaging is a practice of buying a constant dollar value of a volatile asset at regular intervals. This allows to gradually build exposure to a volatile asset while decreasing the influence of volatility on purchase timing.

Dca.land automatically exchanges a fixed amount of DAI to ETH at regular intervals, without any user input.

These instruments, while already extremely useful, only have limited power. With time, as liquidity and number of implemented mechanisms grows, we can see aggregators that fully abstract DeFi instruments from end users, leaving only conceptually simple options, like savings accounts that only expose a single ROI value, while having a lot of rebalancing and optimization internally. This will allow to move DeFi adoption further and further towards end consumers, as opposed to DeFi-savvy users.

PATTERN

Yield farming

Until recently, most DeFi projects were considered to be in a somewhat experimental state and the control over their key exogenic parameters was done manually by the team (with rare exceptions, e.g., Maker). As the protocols matured, however, the need to distribute governance power were called for more and more. The natural way to facilitate governance rights are governance tokens similar to MKR, but distributing them to users in a way that is fitting for DeFi is non-trivial.

The projects opted for the approach whereas governance tokens are distributed to users in proportion to their "usage" of the platform. This may mean different things for different protocols, e.g., in lending pools tokens can be issued in proportion to accrued interest on borrowing or lending, while in swap pools they are issued simply in proportion to provided liquidity.

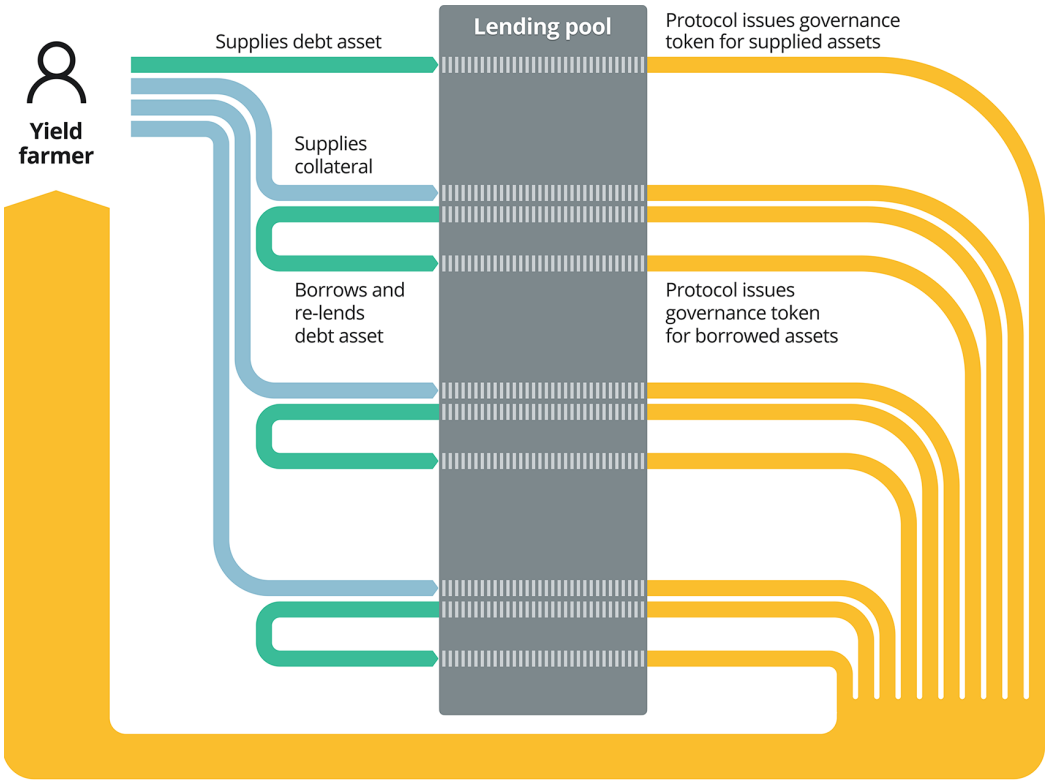
The first project that introduced this mechanism was Compound with its COMP token. A fixed amount of COMP is issued per day, and it is distributed to lenders and borrowers proportionally to the dollar equivalent of their accrued interest.

On paper, this distribution method aligns incentives well for governance, as active users of the platform that have direct interest in maintaining its health will have the largest voting power. However, because of speculative interest, incentives were quickly warped. COMP (and soon after YFI and others) is distributed in fixed amounts, but it is a volatile token. The initial issuance of governance voting power in a popular protocol attracted a lot of interest from traders, which led to the price of the asset quickly inflating.

As a result, platforms arrived in a strange state where the projected APY of governance token issuance dominated the actual yield from liquidity provision, sometimes up to 1–2 orders of magnitude. This upside is, in most cases, notional, as inflated token prices would inevitably drop, after the initial publicity wave wears off and the token supply increases.

However, high projected APYs still attracted large speculative interest from traders hoping to cash in before the crash. This started the practice called "yield farming" or "liquidity mining", which is providing liquidity in a specific way in order to maximize the return of governance token issuance.

In some protocols yield farming led to unintended use of the protocol. For example, Compound's case was briefly covered in the discussion of alleged "reserve banking" in DeFi. To recap, traders started to repeatedly lend assets in lending pools and then immediately borrow the same asset, only to lend and borrow it again—this can be repeated as long as the trader has enough collateral to support the position. As a result, the trader earns governance tokens both from paid and earned interest on the debt asset, as well as earned interest on the collateral asset.



Compound yield farming

While farming yield does not damage platform's solvency to a great extent (all positions are collateralized, even when exactly the same tokens are being borrowed), it does raise borrowing rates, as all available liquidity is engaged in farming, as well as generate poor publicity (e.g., the "reserve banking" misconception).

Aside from disrupting the operation of individual protocols, yield farming may also impede DeFi on macro level—with the changing market dynamics farming on different protocols may become more or less profitable. Liquidity providers then follow the profits, hastily moving large amounts of liquidity between protocols. This leads to large swings in available liquidity supply, which results in volatile rates and market depth (in the case of swapping pools).

At the same time, usage-based governance token distribution can still be utilized by project teams productively, as long as incentives to farm are kept within reasonable levels by carefully considering all mechanisms. Usage-based distribution can facilitate community governance that is extremely proactive and oriented towards sustainable long-term growth. This can already be observed to some extent even in current governance ecosystems, which are very young:

1. Dozens of standing proposals are being raised in discussed in YFI, Compound, Curve and others;
2. Community governance initiatives that benefit the ecosystem as a whole are being proposed—e.g., YFI holders voted to allocate 1% of platform revenue to Gitcoin grants;
3. Compound governance proposed and accepted so-called "borrow caps" to combat the aforementioned Compound yield farming technique.

For better or worse, yield farming seems to be here to stay. The mistakes of the initial governance token issuances clearly burned the community, and now project teams and shareholders actively seek measures to limit speculative craze in the future—either by reducing issuance or plugging exploits, as in the Compound's case. Yield farming is an example which demonstrates that despite all its success, DeFi has a long way to go before it is truly mature.

[18] Although sustainability and ecosystem health may not be the only motivator - these measures can also be seen as existing shareholders raising entry barriers for new shareholders, after mining a lot of governance tokens themselves during the initial periods of high yield.

4 Properties of DeFi

Introduction

After a deep dive into the specific mechanisms that underpin various DeFi instruments, we can now identify the unique properties of this new financial ecosystem and look more carefully at the effects and opportunities to which it gives rise.

If we were to describe the typical mode of operation between the user and DeFi, we might characterize it as a set of *arbitrarily composed* and *atomically executed* interactions with *self-sustainable economic abstractions*, each of which operates in a *predefined* and *tamper-proof* fashion and has *liveness*.

Each of these components and properties of the described interactions requires more detailed exploration.

4.1 Economic abstractions

Operating within the normal market maker/taker terminology, makers in DeFi are extremely unique. Decentralized finance closely follows the design philosophy of Bitcoin by taking some well-known economic processes and inserting an automated abstraction where a counterparty typically would be, while ensuring that the processes continue to function as intended.

By *abstraction* we generally mean an interactive entity that may operate based on certain complex machinery behind the scenes, but exposes a simple interface with a clear functional purpose. In the aforementioned case of Bitcoin, the abstraction would be the decentralized ledger of unspent transaction outputs that offers an interface for the functions of holding and transferring value[19], with the replaced typical counterparty being a payment provider, a bank, a remittance service, etc.

The main requirement for an abstraction, as opposed to a counterparty, is that it must operate predictably in order to truly remove counterparty risk. In the context of DeFi, the other requirements naturally follow:

1. The abstraction must operate under a predefined set of formal rules;
2. The abstraction must be tamper-proof, as otherwise it could be forced to deviate from the predefined rules;
3. The abstraction must have liveness, meaning that it must respond to every query it receives within some reasonable time frame (which may or may not be known in advance). If the abstraction doesn't have liveness, it might not respond at all, violating predictability;
4. The abstraction must be self-sustainable, as otherwise it could fail at any moment and would no longer be able to operate under the same set of rules. This requirement exists within reasonable boundaries, but a careful exploration of the limits of self-sustainability is a vital part of assessing the robustness of the abstraction.

Within DeFi, the first three properties are typically facilitated by the underlying technology (barring programming and architectural errors). The fourth property is a function of the abstraction's mechanism design and is most typically achieved by choosing an invariant and ensuring, through technology and/or alignment with naturally expected behavior of market forces, that this invariant holds. Some of the examples of invariants we've seen in practice are:

- The total held value of all liquidity in the pool cannot decrease after a transaction (pool-based exchanges);

[19] The functions of multi-signature custody, quasi-anonymous ownership and transactions, hash-locked and time-locked contracts, etc., that Bitcoin also provides are, in this case, secondary to the discussion at hand.

- The total value of collateral cannot be lower than 100% of the total value of debt (collateralized debts);
- The operator of an L2 solution cannot successfully pass proof verification when they violate L2 rules (off-chain exchanges with zero-knowledge proofs).

Note that it is generally impossible for the invariant to be upheld under every possible set of market conditions, no matter how severe. For example, the value of collateral theoretically can go lower than 100% of the total value of debt—this might happen when, for instance, DAI peg breaks and DAI depreciates sharply. Many mechanisms exist to specifically prevent this scenario, however. The operator of an L2 solution can forge a proof if they can break the underlying cryptographic primitives. Again, companies implement elaborate measures to test their cryptographic innovations[20].

Therefore, the 4th requirement exists given reasonable assumptions about the severity of acceptable systemic risk. However, as we demonstrate below, the existence of a tractable invariant allows one to reliably predict the nature of future black swan events and embed countermeasures into the internal machinery of the abstractions, thus achieving a very high level of robustness in practice.

The motivation—why we would want to replace a counterparty with an abstraction in the first place—is two-fold: efficiency and forecasting.

From the standpoint of efficiency, seeking counterparties and hedging counterparty risks demands a considerable amount of resources and can be arbitrarily difficult depending on the taker's starting conditions (i.e. the issue of the unbanked). Even if the issue is manageable, it generates costs upon the majority of economic interactions. At the same time, an abstraction can mask arbitrarily complex socioeconomic machinery, but as long as the abstraction behaves in the way it's intended, it can remain a single point of entry (eliminating counterparty seeking) and does not generate major risk.

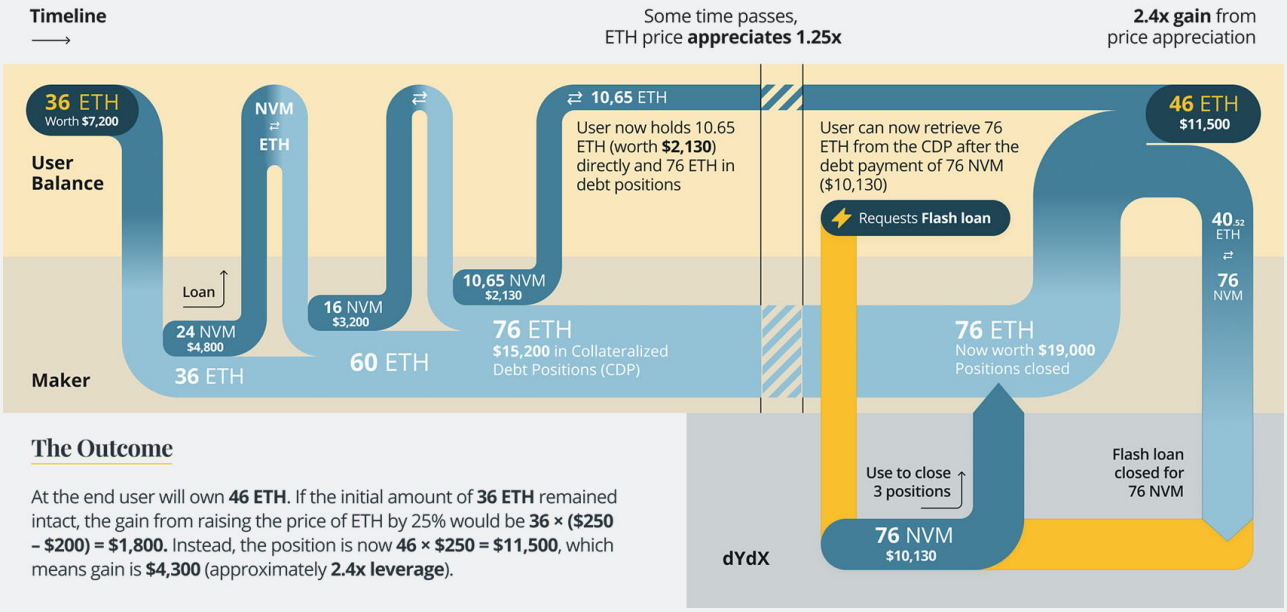
From the standpoint of forecasting, the advantages of the system that behaves predictably are obvious. Firstly, the potential ROI and other parameters that arise from the interaction with this kind of system are easier to estimate, which leads to more precise strategies. Secondly, when composing several instruments that rely on counterparties (which DeFi tries to do away with), the risk that at least one of them will fail increases exponentially. Low counterparty risk allows one to build longer and more complex interaction chains that are robust enough to be practical, which leads to the next point.

[20] The typical pipeline includes publishing papers on the construction, open-sourcing reference implementations, organizing peer reviews, and setting up bounty programmes with substantial financial rewards to compromise the primitives. Only relying on internal resources is generally considered bad practice.

CASE STUDY

Connectable abstractions

A brief example of arbitrarily connectable abstractions can be built up on a previously presented pattern of leveraged positions with re-borrowing. We extend the example by using a flash loan instrument to close the position after a market movement. This example is hypothetical in the sense that there is no single product that currently captures the whole interaction, and we are assuming a scenario instead of referring to an actual third-party representation. But all of the pieces are there, and the premises are reasonable.



The Outcome

At the end user will own **46 ETH**. If the initial amount of **36 ETH** remained intact, the gain from raising the price of ETH by 25% would be **36 × ($250 – $200) = $1,800.** Instead, the position is now **46 × $250 = $11,500**, which means gain is **$4,300** (approximately **2.4x leverage**).

The key takeaway from this streamlined interaction is that coupling DeFi instruments together is easy (given basic engineering knowledge, until someone publishes a free open-source instrument to that effect), and the correct behaviour of coupling is enforced by blockchain rules.

The concept of gluing instruments together is powerful, and we shall examine its implications in DeFi versus traditional finance in the next section.

4.2 Arbitrary composability of instruments

Within the investigation of DeFi instruments in the previous sections, there are numerous cases where an instrument aggregates, or "composes" on top of one or more other instruments, with the aim of improving ROIs, hedging risks, or creating all-new business models.

Composability is an important feature in DeFi for two main reasons:

1. New business models can be built on top of existing instruments at low integration costs. Therefore, it is possible to create products that cater to different profiles of investors without needing to bootstrap liquidity on the other side. For instance, a new pool can leverage the existing APR of widely-used instruments while distributing the proceeds to its investors based on its own exotic logic. An interesting example of this is PoolTogether, which essentially facilitates a lottery that is financed by supplying liquidity on Compound, creating an instrument that has the same expected APR as Compound, but which caters to more risk-loving investors.
2. Similar instruments can be grouped and presented as unified packages, thus simplifying decision making for investors: instead of choosing among a wide variety of instruments, the investor only needs to compare the groups, as ROI is optimized within each respective group on its own. An example of this is yEarn or Curve.

However, to be able to compose arbitrary instruments, it is necessary that each individual instrument is as safe and predictable as possible, as only a single failure is required to break the entire composite, and failure probability compounds.

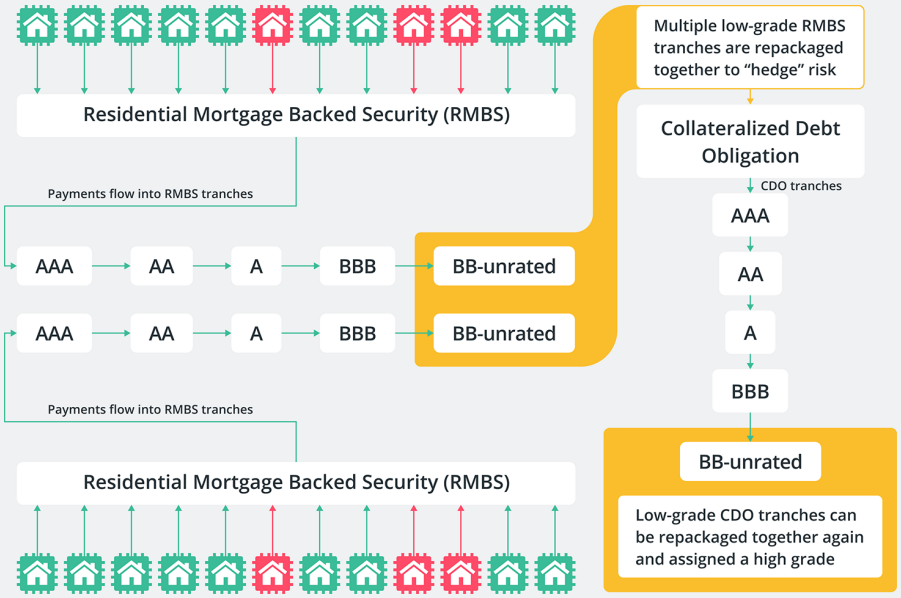
Let us first consider how composability operates in traditional finance. There are at least two ways in which the process can go wrong.

First of all, arbitrary composability is hardly achievable without an extensive network and considerable resource commitment. Every candidate instrument resides either in OTC markets (and therefore lacks standardization and coordination between numerous independent participants in these markets) or in heavily regulated markets that naturally restrict entrance based on composability and capital. The time and capital commitment of building a composite instrument increases with the number of involved components and, not infrequently, in a non-linear fashion.

Secondly, there is a question of adequate risk assessment of composite instruments. In traditional finance, instruments that serve as building blocks of a composite instrument reside in different markets, often in areas of economics that differ in their observability. As an example of a complex and highly opaque instrument in traditional finance, consider CDOs (collateralized debt obligations)—see Interlude.

CASE STUDY

Observability of CDOs



CDOs (Collateralized Debt Obligations) are instruments that lead to skewed incentives in the mortgage market and that ultimately contributed considerably to the 2008 financial crisis.

Let's break the instrument down into its constituents:

1. At the lowest level reside typical consumer mortgages—consumer loans collateralized by real estate;
2. Subprime (i.e., high default risk) mortgages are grouped into RMBS (Residential Mortgage Backed Securities). A single series of RMBS is used to finance all mortgages in a group, while the security itself is meant to hedge the default risk on a single mortgage. RMBS are split into several tranche grades based on their risk rating—AAA to unrated. The highest-rated tranches are paid interest first, which means that low-rated grades carry the brunt of the default risk.
3. The lowest-rated tranches are then grouped into the CDOs. The base premise is the same as subprime mortgage grouping—packaging several risky instruments into one allows hedging risks on each individual instrument. The tranches in a CDO are structured similarly to RMBS.
4. The riskiest tranches of a CDO can be repackaged again in the same manner—repeating ad infinitum.

On paper, AAA CDO tranches looked sufficiently robust to be purchased even by investors with low-risk profiles, such as pension funds. If twenty B-level RMBS are packaged into a CDO, it would take about a half of them not paying any interest for the AAA CDO holders to not get paid. Even with a chance of 50% of each individual RMBS tranche not paying, the overall failure rate of a AAA CDO tranche would be extremely low.

However, the general success of mortgage-backed securities created skewed incentives—institutions specializing in packaging RMBS were incentivized to create more mortgages, which led to progressively deteriorating lending

standards, which, in turn, led to a significant increase in the default risk on each particular mortgage.

Instruments that would have been considered "safe" under normal market conditions (i.e., with a 50% chance of an RMBS not paying), were no longer safe in the new market (i.e., with a 95% chance of an RMBS not paying). As a result, many AAA CDOs failed to pay.

At least two reasons can be identified for the overall "unsafety" of the instrument:

- While it is technically possible to decompose the instrument into constituents and attempt to determine the robustness of each individual element, in practice this would require documentation from dozens of individual sources[21], which is a large time and resource commitment.
- The constituent instruments are counterparty-based, do not behave in predetermined ways, and can only be reasoned about statistically. Consequently, it is hard to formalize the failure conditions for the overall instrument. A CDO would fail with a 99% probability if mortgage default probability were to reach *x%*. But the value of mortgage default probability is not easily tractable and, in the best case, requires resource-intensive market research.

If we observe the properties of economic abstractions in DeFi, we will see that they are uniquely suited to be arbitrarily composed.

Firstly, abstractions are typically much more interoperable than traditional financial instruments, since they generally exist within the same ecosystem, have clearly-defined interfaces and have a framework for interacting with each other[22]. From that standpoint, composing them to produce new instruments does not require the expensive process of setting up the initial communication channel—but only requires sufficient know-how to make abstractions talk to each other in a particular way.

Secondly, abstractions are considerably more observable than traditional financial instruments, since there is a unified data layer that can be queried to find out the addresses/identifiers of the constituents of a particular instrument, and then to determine the current parameters of the constituents. Currently, while proper data sourcing often requires running a costly full node, projects such as The Graph or Pocket Network will likely make queries significantly cheaper over time.

Finally, abstractions are guided by formal rules, rather than stochastic counterparty behavior. Because of this, it is relatively simple to formalize scenarios in which these abstractions fail and to design countermeasures. The following study of such a hypothetical scenario demonstrates this.

[21] In practice, owing to the statistical nature of the instrument, one does not have to observe all the individual mortgages, but only has to acquire a statistically significant sample. The sample still has to be reasonably large, though.

[22] There is a risk presented by DeFi instruments in separate blockchain networks that currently do not have a robust communication framework to preserve the level of decentralized trust. This risk will likely be addressed in the future by blockchain interoperability networks that can securely route arbitrary data from one chain to another. But as of now, this is a WIP field.

CASE STUDY

DAI peg breaking and automated collateral coverage

Imagine the following hypothetical scenario:

- The network is heavily congested;
- An attack is successfully mounted on core DAI[23] contracts, leading to uncontrolled minting of new DAI;
- This fact causes panic on external markets and a bear run, with DAI price depreciating more than 33% over a short span of time;
- Due to congestion, liquidators on Compound, Aave, Fulcrum, etc. are not able to clear unhealthy DAI-collateralized debt on time, and will not do so afterwards either, since now they will receive less value than they have available to supply liquidation.

In this situation, the invariant on lending markets breaks due to sudden collateral depreciation coupled with the inability of liquidators to eliminate dangerous debt on time.

Notably, this is one of the most likely scenarios in which borrowing pool self-sustainability would be broken, and the potential for its occurrence can be readily identified once the workings of the borrowing pool mechanism are investigated. From there, the risk can be addressed by either reducing its probability (in case of collateral depreciation, the course of action is not clear), or addressing the severity of consequences. We present a mechanism that could be used to eliminate undercollateralized debt.

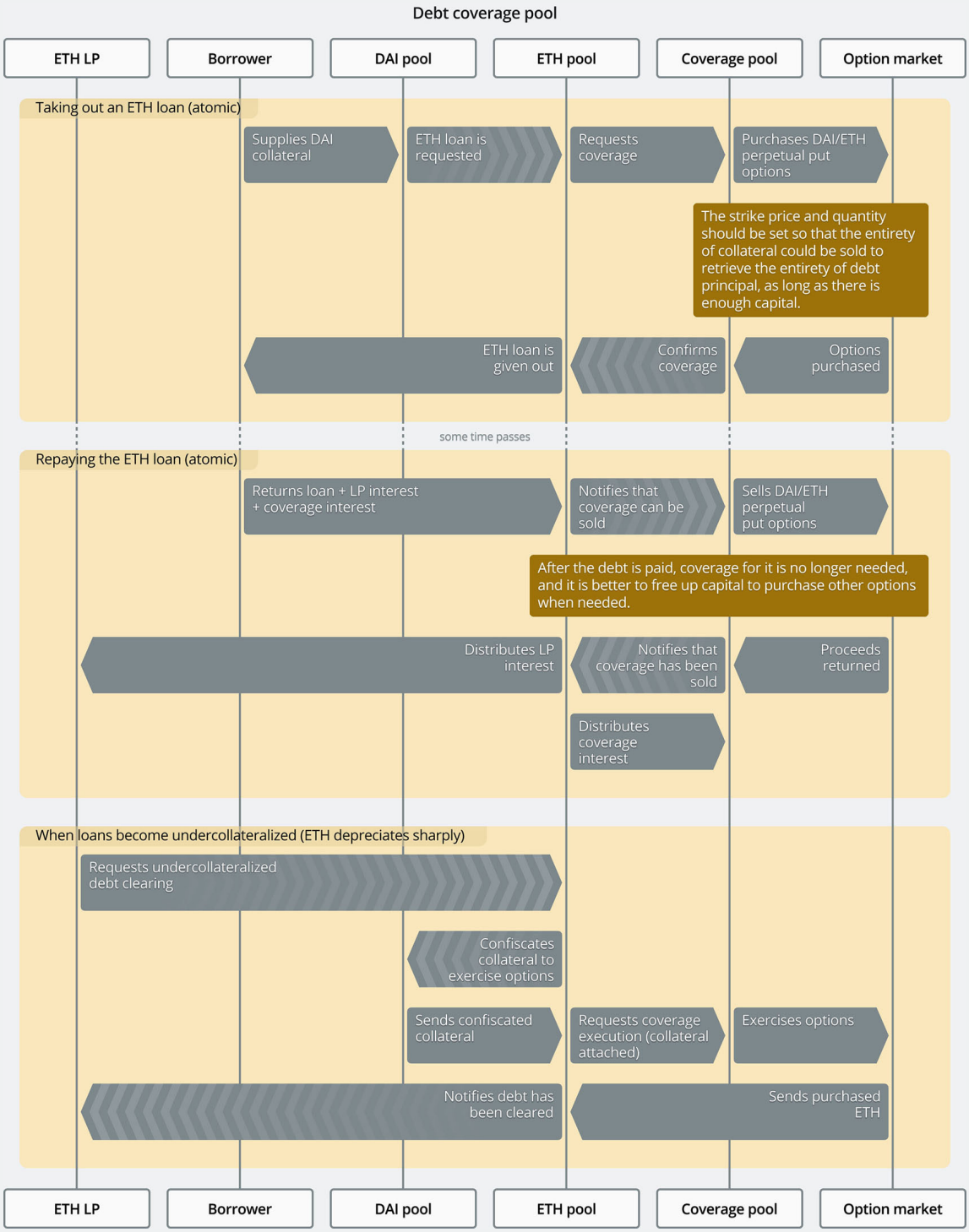
The proposed mechanism utilizes a special coverage pool and smart contract put options to cover collateral. Note that currently some instruments (e.g., Fulcrum) keep insurance pools that are manually controlled—these can be repurposed to interact with smart contract options autonomously.

Under normal operating conditions, a part of the interest rate paid by the borrowers is put into a coverage pool. The coverage pool is triggered each time some new debt is created; it determines the collateral structure of the borrower and buys put options from collateral to debt asset in required proportions. In the example case, the entirety of collateral is DAI, so the pool buys DAI/ETH put options.

If the debt is repaid normally, the coverage pool is triggered and sells options, as they are no longer needed.

If the black swan event happens and the loan becomes undercollateralized, the liquidity provider can trigger a special function to recover their principal (the outstanding interest rate has to be abandoned, as the strike price of an option has to be static by the nature of the instrument). The function prompts the coverage pool to withdraw collateral from its pool and send it to the option writer to exercise options. The purchased asset is then sent to the respective pool to be withdrawn by the liquidity providers.

[23] Technically, this case is viable for any collateral—DAI is chosen as an example.



From this example we can see that systemic risks in DeFi can be often effectively predicted and protected against due to the deterministic nature of instruments and high observability. This allows for a relatively safe composition of various instruments, opening the door to a multitude of possibilities for creative composite instruments.

> ***Example: composing assets***
> *A typical (at the time of writing) line of thought about composing instruments is using assets in custody for liquidity provision: for example, Curve, a pool-based exchange for stablecoin arbitrage, holds Compound liquidity tokens rather than underlying assets. All trading operations wrap around Compound (depositing and withdrawing underlying assets to facilitate the trade). In effect, liquidity providers in Curve have payouts from liquidity provision in Compound, and Curve is dependent on Compound's robustness for its own financial security.*

4.3 Atomic execution

A fundamental property inherent to the design of Ethereum[24] is *atomic execution* of complex transactions. Atomicity is a property of transaction processing that guarantees for complex transactions (i.e., a batch of smaller transactions in a sequence) that either everything succeeds or, if any one transaction fails, everything fails, in which case the state is reverted to its initial state, as if nothing had happened at all.

This is a well-known technical term in software development (specifically, database design), but smart contract execution represents the first time the concept has been brought into a wider context with the same technical meaning but much broader implications. An additional property introduced by the blockchain setting is that, during transaction execution, the rest of the state of the entire system does not change—at most one transaction is being processed at any given time.

Traditional finance only supports atomic execution for transactions that are limited in scope, generally related to a lifecycle of some standardized instrument. Moreover, atomicity is very limited between markets: for an atomic operation that includes assets in different markets, interoperability between corresponding exchanges is required, which can only be facilitated by the exchanges themselves. Interoperability of this kind between arbitrary OTC instruments and exchange-listed instruments is almost unimaginable.

If a trader wants to perform a complex operation involving instruments from many markets, they would likely have to split it into non-atomic transactions and have no guarantee that the state of the market will be stable in between. This can lead to the state of the market changing between the execution of transactions, making the overall complex transaction non-profitable. This may occur due to, for example, latency—in which case the next transaction in the sequence is received by the exchange significantly later than the previous one due to a short-term network issue.

This possibility produces a large amount of uncertainty for the trader when considering a complex arbitrage opportunity. They can potentially leverage the opportunity by sending several transactions, but if the market changes in the meantime, they will incur loss on the value of the entire transaction.

In comparison, in DeFi an atomic transaction can have an arbitrary (within the network-set block limit) number of interactions with any kinds of assets and instruments. Moreover, the transaction can also be made to revert entirely on an arbitrary sender-determined condition—this means that profitability of a transaction can be measured at any time during its execution, and the transaction can be reverted if it is not profitable anymore.

This feature removes most of the uncertainty when a trader is presented with an arbitrage opportunity, as now their risk is limited to the network fee, which in most cases is not on the same order as the value of the transacted assets.

As a result, atomic execution further contributes to composability, given that not only the included instruments are predictable, but the functional machinery of the composed instrument may be arbitrarily complex with very low risk. The overall state of the market does not change during execution, except for the action performed in the transaction, and if some invariant during execution is broken (e.g., the overall value of assets reduces dramatically due to slippage), the transaction can simply be abandoned altogether without incurring significant losses.

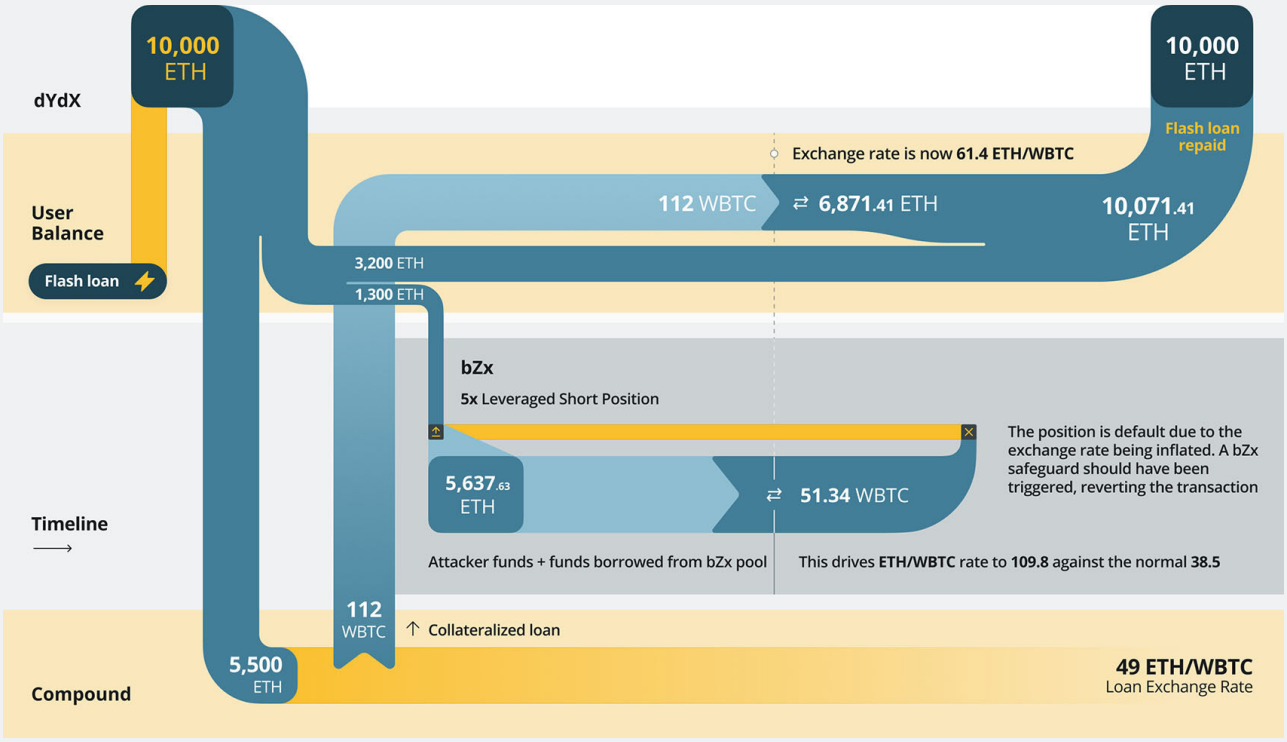
One particular example—despite actually being an attack on one instrument's smart contract—showcases how atomicity and complex interaction with many instruments can be leveraged for powerful results.

[24] As the pioneer in smart contracts of arbitrary complexity—many other protocols follow a similar model and have this property.

CASE STUDY

Fulcrum attack

Recently a highly complex attack was mounted at Fulcrum, a margin trading solution within the bZx network. This overview is based on an extensive analysis report by Peckshield.



The following events happened within the span of a single transaction:

1. The attacker takes out a 10,000 ETH flash loan at dYdX;
2. The attacker puts 5,500 ETH into Compound as collateral and borrows 112 WBTC (~49 ETH/WBTC);
3. The attacker opens a 5x leveraged short position on bZx, sending it 1,300 ETH of "their" funds.
4. bZx takes additional borrowed funds from its own pool and tries to exchange a total of 5,637 ETH through Kyber. Kyber determines that the best rate is at the Uniswap pool, and routes the transaction there. But Uniswap has limited liquidity on the pair, which results in the prices being driven up and only 51.34 WBTC sent back. At this point, bZx should have determined that the position was unhealthy (as at normal exchange rate it is worth slightly less than 2,000 ETH), but doesn't due to an implementation bug[25].
5. The attacker sends 112 WBTC to Uniswap and exchanges for 6,871.41 ETH.
6. The attacker now has enough ETH to repay the flash loan, which they proceed to do.

As a result, the attacker has a Compound loan with about 49 ETH per WBTC in it, while the normal exchange rate is 38.5 ETH per WBTC. This is typical for any debt position, due to overcollateralization. However, in this case the attacker didn't place any of their own funds as collateral; instead, through

[25] Currently, the defaulted position still exists and is considered protocol debt. bZx has issued a roadmap on debt repayment using an internal insurance fund (postmortem in bzx blog)

complex machinations, the position was created at bZx pool's expense. Finally, the attacker buys out WBTC in small quantities (to avoid moving the price) and gradually repays the loan.

This attack was carried out essentially without any pre-existing capital, although some capital was required afterwards to pay out the loan. The attack demonstrates the power of integration within DeFi but also the danger: a single instrument having an exploit within it can influence other instruments in unpredictable ways. In this case, the attack led to short-term instability in Uniswap (instability which, however, was quickly removed through arbitrage).

Less risky complex transactions lead to a possibility of more effective and creative strategies—investors can concentrate on selecting the best strategy regardless of complexity, rather than having to consider whether entering a complex position or arbitrage opportunity is worth the risk.

The ability to arbitrarily revert transactions also can produce entirely new business models. For example, it enables flash loans, which innovate considerably on how capital is acquired in finance.

4.4 Emergent threats and skewed incentives

While it is tempting to look only at the positives of DeFi, it is also important to understand that DeFi is a rather fragile ecosystem that at times can be viewed as its own worst enemy.

Self-sustainability of DeFi instruments relies on a careful balancing of technological and mathematical techniques with incentives, a balance which results in systems that are robust against outside manipulation but that can be extremely easily disrupted from within.

When a new instrument is deployed, the stakes are not extremely high—no funds are committed into the instrument by investors, and no instruments are built on top of it, so a flaw in mechanics will not lead to significant consequences.

However, when a new mechanic is introduced into an existing instrument, especially when this mechanic is not encapsulated and can influence the functioning of the entire instrument, the potential consequences of a mistake are much more severe.

Currently this consideration is most relevant for yield farming. The possibility of generating new wealth out of nowhere leads to overinflated speculative interest, which—as has actually occurred—may be manifested in a gold rush. The particular mechanics of governance token distributions have led to exotic and often undesirable effects, effects which should have been investigated in advance, but were not.

Aside from these immediate side effects, uncontrolled speculative interest may present an existential risk for instruments - as speculators start to realize their profits down the line, it is likely that governance tokens will experience a flash crash (this has already happened to some extent, but it is unclear whether the crashes that have occurred were the worst). Cheap governance tokens lead to cheap costs of changing vital parameters—a situation that can be leveraged by rent seeking attackers to steal funds from the protocol, destroying it in the process.

Overall, the most worrying fact is that the teams behind instruments (as far as they have control over the instruments) seem to be incentivized to prioritize high APRs over new business models and qualitative innovation, since high APR brings traction into projects. However, the nature of that traction is dubious, as it is unclear which fraction of volumes and liquidity will remain once the high APRs inevitably dwindle.

It is important to not allow the success of DeFi to lead to its own demise. High APRs are, generally speaking, not natural and are either temporary or associated with high risks. While an argument may be made that DeFi may produce higher yields than traditional finance, it is important to remember that the true value of DeFi lies in its novel business models, its contribution to self-sovereignty and its accessible, meritocratic nature. Thus, the main focus of the industry should be on reinforcing and utilizing DeFi's unique properties rather than trying to mint value out of thin air.

CLOSING REMARKS

DeFi as a different approach

At the core of decentralized finance lies the concept of protocol custody: digital assets being held by an economic abstraction (as defined in section 4.2) in strict accordance with its predetermined and immutable rules, and without any third-party inputs throughout the whole custodial lifecycle.

Whenever this principle is sidelined (i.e., the abstraction gives out assets from its custody, thereby weakening the guarantee of at-will repayment for every depositor), "the DeFi way" is to provide mechanisms that rely on market forces (rather than reputationally or legally enforced liability) to ensure the return of these assets, usually with multiple fallback mechanisms that still avoid central-party reliance. Because of this, observability becomes even more important: market agents are relied upon to provide solvency, so the markets in which they operate should be observable, with low delays and as much availability of information as possible.

The reluctance of DeFi to opt into fractional reserve banking also boils down to the same key requirement: to provide strong expectations of solvency without resorting to central agents or shaky grounds—such as using a mintable token in expectation that it can be minted and sold off for the missing liquidity. This mechanism still makes sense as a very last resort, but in essence, it is an embodiment of market-driven bailout: the newly minted asset represents the market belief that the system will recover. Otherwise, emission bailout drives a flash crash and a spiral of death for the asset.

The future shape of instruments is largely unknown to us, as the industry is in its very early days. Decentralized finance has already shown quite exciting properties achieved within a reasonably controlled and understood risk framework (as evidenced by the to-date history and a number of theoretical works and simulations), including realistic risk scenarios for the underlying blockchain infrastructure.

Some of the important properties DeFi stands on are inherent to its broader context and are hard to eliminate (such as observability). Concrete attempts to break those properties would be a strong signal that a particular project falls out of the scope of decentralized finance. Some other properties, however, are subtle and require careful consideration to maintain, properties such as full composability without accrual of underlying risk. The temptations to offer a short-term competitive advantage by breaking a property to form long-term value are present and may lead to suboptimal behaviour, as evidenced by traditional financial history. One can only hope that the long-term vision will be a stronger driver for the market.

5 Conclusion

As with blockchains and cryptocurrencies before, the widespread perception of DeFi is often overtaken by unsuccessful or bad faith projects, and wild returns associated with correspondingly high risks. As with blockchains and cryptocurrencies, potential value enclosed within the possibilities the technology brings resides in an orthogonal plane. Not unlike DLT in general, DeFi could be molded into a replacement of existing systems and institutions providing many financial services, greatly increasing both efficiency and resilience in fulfillment of the same functions.

Most of the costs associated with this transition are social: extensive operational models and company ecosystems exist to cover and mitigate the shortcomings of the existing systems, and system-wide replacement of foundations requires the work to cover all the bases, starting with regulatory compliance (or adjustments of regulations to achieve the same ends in a different environment), institutional upgrades, and adoption. Recent years have seen many innovations in technology and operational models for decentralized products, but adoption—being constrained by all of the above—is becoming an increasingly important hindrance.

Let us re-examine the construction so far and then discuss the conceptual relationships between traditional and decentralized finance.

5.1 Retracing the steps

Decentralized finance aims to build financial services resistant to central-party risk. It has a broad operational understanding of central parties, so many traditional forms of protection are unavailable for DeFi protocols. For instance, having a single operator of the transaction medium—trusted to be impartial due to scale—in order to enable e.g. chargebacks is generally unacceptable. The same principle applies to legal agreements and contract litigation for their enforcement: it assumes an entity to formulate and sign the agreement, a responsible litigator, a jurisdiction, courts of law, etc., every one of which could be considered a trusted central party.

Instead, decentralized finance has its own set of instruments and measures designed specifically for environments running on tamper-proof code execution. Here is where complex cryptography (such as zero-knowledge proofs) and interactive schemes with arbitrary market agents (such as fishermen[26]) come into play, combined with automatic rule-based rejection of transactions, staking, and code-driven stake slashing. All of these instruments are generally free of centralized regulators or mediators, bringing them close to full trustlessness. Central-party risks make a service centralized, and absence of such risks makes a service trustless or decentralized.[27]

The foundational layer, consisting of blockchains and smart contracts, offers the aforementioned trustless code execution, tamper-proof persistence of data, and several baseline volatile assets as a useful byproduct of maintaining these properties—Ethereum's native token, Ether, is one example.

Retaining the near-zero trust requirements, these properties can be leveraged to produce non-volatile payment mediums, stablecoins. Note that the ability to make this step trustlessly does not depend on availability of decentralized exchanges: as long as individual market participants are able to make trades somewhere and exchange liquidity with the system, stablecoin projects can work. So for the purposes of stabilization (driven by arbitrage profits), a multitude of varied centralized markets counts as a decentralized one.

With stable money available, trustless versions of several classes of day-to-day operations become available without mandatory exposure to assets with uncertain volatility:

- Delayed payments (without volatile currency risks[28]);
- Trading (with stablecoins as safe havens and on- and off-ramps);
- And lending (non-volatile liabilities or low-risk investments).

In the latter case, the liquidity provider side of the equation can aggregate funds from many agents and lend them out in a trustless way via a tokenized pool mechanic. If the provided liquidity remains in the custody of protocol, and decentralized trading is available, it is possible to implement margin positions without either overcollateralization and putting liquidity providers' funds at risk. Pooling liquidity also makes flash loans possible.

[26] A fisherman in DLT is a market agent who combs through the network history to spot provable malicious behaviour and report it to the protocol (rather than a stakeholder) for automated action that punishes the attacker and rewards the fisherman.

[27] Important nuances and caveats to this statement are given in the foreword.

[28] Of course, traditional fiat currency risks still apply if the "eventual" source and destination fiat currencies differ.

5.2 Going into the future

The potential of emergent properties of decentralized finance is a source barely tapped so far. The wide market adoption of the industry is still heavily influenced by its regulatory status, as the questions of legal classification of many instruments are still uncertain, and the future in context of AML compliance is also shaky.

On the one hand, it is nearly impossible to imagine a general audience adopting these instruments without the support of regulatory enforcement. On the other hand, inconsiderate insertion of enforcement tools into the instruments would immediately break them, dropping the unique emergent features and reverting everything to traditional finance running on some new software with additional overheads. Compromises are possible at certain places and have not yet been established in others—with neither side of the discussion arguably willing to compromise. A common target for installing compliance might be on and off ramps, which would in turn have to implement measures to decipher and deanonymize the on-chain interactions to reason about their "legitimacy". Selective transparency for auditors or regulatory bodies is possible even for certain classes of anonymized solutions via cryptographic means.

"Proper" DeFi can offer several advantages over traditional instruments:

1. Observability of risks and market dynamics—which, with novel cryptography, can be achieved without breaking anonymity;
2. Composability of instruments that does not add up opacity;
3. Leverage for special cases of guaranteed arbitrage[29] with flash loans.

But even leaving these advantages aside, several technologies and concepts developed in the context or within the scope of DeFi can be used to shift the traditional usage of finance for increased resilience:

1. Protocol custody (custody without central party risk);
2. Self-sovereignty of asset holding (shifting to user-owned cryptographic keys rather than centralized custodians);
3. Automatic market making and different price discovery mechanisms (such as pool-based exchanges).

In summary, in terms of mass adoption substantial uncertainty persists, and the bigger dialogue is still yet to happen. Meanwhile, in the grey areas and in between strong jurisdictions, decentralized finance is already demonstrating most amazing constructions and emergent effects with what seems to be great potential in creating efficient equal opportunity markets without incurring prohibitive social and financial costs of making that possible. This is just the beginning.

[29] The atomically available on-chain kind.