Automated Strategy Vaults
The world of on-chain autonomous investment managers: Optimizers.

Filippo Caprioglio, Unexployed

3six9 Innovatio, Cognitio Research.
{filoxxx, unexployed}@3six9.io

Abstract: The invention of yield optimizers marked the beginning of automated strategy vaults. The concept was first introduced by Yearn, which built its vaults using the ERC-20 standard for yield-bearing receipt tokens. The sector has developed substantially since with the introduction of a new token standard, ERC-4626, designed to further enhance liquidity and capital efficiency. The protocols central in this piece are Yearn, BadgerDAO, mStable, Sommelier,Spool and Beefy Finance. We find that it is essential for optimizer protocols to particularly focus on decentralized liquidity management and security. The first can be achieved by using the Keep3r or Gelato Network set of keepers, while for the latter it is essential to provide users audit reports and build on top of blue-chip protocols to create a flywheel effect.

Keywords: **Automated Strategy Vaults** · **ERC-4626** · **Yield** · **Optimizers**

Introduction

This is the second installment of our '*Automated Strategy Vault*' series in which we highlight different structured, yield-generating products. In the first paper, we gave an introduction to automated strategy vaults. We explained how important these products are for DeFi as they take away management and monitoring from the user and automate the whole yield generation process. They can be seen as on-chain hedge funds with the caveat that they are publicly accessible for anyone, regardless of size or citizenship. The main subject of the first paper was *DeFi Option Vaults* ($DOVs$). These are projects that offer options strategies, the most common ones being *Covered Calls* and *Cash-Covered Puts*. Strategies like these tend to earn high base yield, because of options premiums, but also have their weaknesses. For one, the yield is not guaranteed, since options can expire in-the-money resulting in negative returns for that epoch.

This paper examines the second category of strategy vaults: optimizers. In fact, it was the invention of optimizers that marked the beginning of strategy vaults.

In February of 2020 *Yearn Finance* was launched by *Andre Cronje*. The concept of his project was simple: aggregate the best yield on stablecoins across DeFi in a decentralized and autonomous manner. It was something that *Cronje* had been doing manually in his spare time, however, after a while he found the process of comparing various DeFi projects and their interest rates boring. He started what later would become *Yearn Finance*. The protocol became a huge success and amassed multiple hundreds of millions in TVL within 8 months after launch. Even more so, *Yearn* pioneered an immature industry of yield optimizers. One that would grow to dozens of projects with a wide variety of yield-generating strategies.

The core premise of *Yearn* is straightforward; A platform where users are able to deposit assets which are then deployed in an autonomous and algorithmic manner to high and safe yield venues usually involved in at least one of the following elements: lending, borrowing, and liquidity providing. Over the course of its existence, *Yearn* has expanded its product suite considerably and now offers stablecoin vaults as well as volatile asset vaults.

Nowadays there are a ton of other yield optimizers on the market, but *Yearn* still stands out as it maintains a thriving community of strategists and curators, and has a proven track record of being safe, reliable, and efficient.
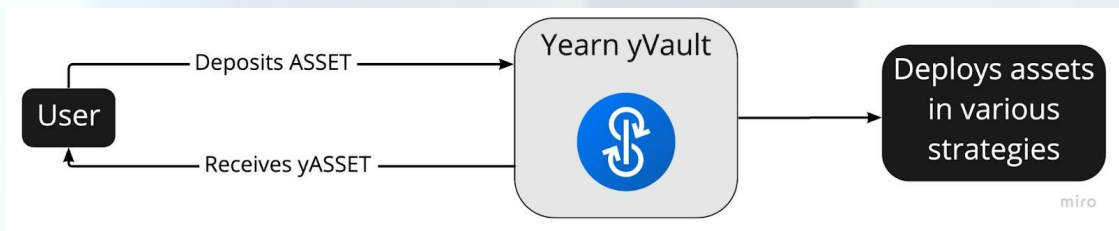
Credit to Marco Worms, Medium, "Yearn Finance Explained"

What happens with DeFi vaults is that, all of a sudden, these strategies are available to anyone and, in most cases, without a minimum amount to opt-in. This opens up a world of opportunities regardless of a users' historical net worth or annualized income which is a huge contrast to the gated TradFi world.

ERC-4626

Before deep-diving into the various yield-bearing strategy protocols, it is important to lay out the current standing of this field. First, a vault may refer to a multisig smart contract solution securing funds of a protocol, a DAO, or someone's individual holdings. The term vault can also be used in reference to smart contracts that store and manage crypto assets according to a strategy. These vaults can be manually managed or automated and are the core subject of this paper. Broadly speaking, there are two main categories of automated strategy vaults: smart contract vaults with yield-bearing assets and tokenized vaults ($ERC-4626$).

The first category allows users to deposit any $ERC-20$ token to a smart contract, which pools all the tokens together and runs a yield generating strategy. In return, the user receives representative vault tokens (from now on referred to as $vTokens$, but the name may differ across protocols). These $vTokens$ can be returned at any point in time to the smart contract in order to redeem users principal, plus any yield accrued. The smart contract takes the pooled tokens and, depending on the strategy, deploys them to other protocols to earn yield and/or generate revenue. The strategy followed by the vault can be preprogrammed, like in $Yearn$, and done entirely automatically and in a decentralized manner through the use of smart contracts and keepers. The strategy can also be centralized and managed by the protocol as in $JonesDAO$. In the $Jones$'s case, a certain group is responsible for the deployment of the assets in the protocols. Generally speaking, the group can be the team behind the protocol, strategists or a collective.

Yield-bearing tokens represent fractional ownership of a pool of assets within a smart contract vault. Examples of such models are $yvUSDC$ and $xSUSHI$. As the underlying smart contract accrues value, the pool of assets grows, and so does the value of the representative tokens. Yield-bearing tokens follow the $ERC-20$ standard, meaning they are fungible and can be used elsewhere. The main problem with tokenized vaults is the integration of the tokens with different protocols. In particular, the hassle stands in the need to research and have knowledge of their model of yield accrual and the risks involved, and then adjusting it into the codebase of a protocol integrating that token/vault. This results in not only being stressful and time-consuming, but can increase the smart contract risk because of potential errors.

In December 2021, a group of smart contract engineers including *Joey Santoro* (*Fei* protocol founder), *T11s* (researcher at *Paradigm*), and a team of four other *Ethereum* developers proposed an *Ethereum Improvement Proposal* (*EIP*) to the *Ethereum* community. The proposal outlined the idea behind the tokenization of vaults and a new token standard called $ERC-4626$. The standard was approved in May 2022 after going through several reviews and deliberations.

The $ERC-4626$ standard optimizes the composability and flexibility of yield-bearing vaults opening to the possibility of new use cases and being more user-friendly. The end goal of this introduction was to standardize tokenized vaults and to make protocol integration easier and less prone to error, enhancing quick development and composability. The standard has been adopted by *Yearn Finance*, *mStable*, *THORSwap*, *Balancer* and *Umami* among others. The main benefit of the $ERC-4626$ is that it is able to streamline the integration of 'traditional' yield-bearing tokens in other DeFi protocols and can best be understood as a unified plug for connecting tokens to services through vaults.

Technically speaking, all $ERC-4626$ tokenized vaults must implement $ERC-20$ to represent vault shares. If a vault is to be non-transferrable, it may revert on calls to *transfer* or *transferFrom*. The $ERC-20$ operations *balanceOf, transfer, totalSupply*, etc. operate on the vault shares, which represent a claim to ownership on a fraction of the vault's underlying holdings. The $ERC-4626$ tokenized vaults may implement $EIP-2612$ to improve the UX of approving shares on various integrations.

The $ERC-4626$ vault interface is designed to be optimized for integrators with a feature complete yet minimal interface. Details such as accounting and allocation of deposited tokens are intentionally not specified by the standard, as vaults are expected to be treated as black boxes on-chain and inspected off-chain before use. $EIP-20$ is enforced because implementation details like token approval and balance calculation directly carry over to the shares accounting. This standardization makes the vaults immediately compatible with all $EIP-20$ use cases in addition to $EIP-4626$. $EIP-4626$ is fully backward compatible with the $EIP-20$ standard and has no known compatibility issues with other standards.

Benefits of the $ERC-4626$ standard	Consequences
Reduce development obstacles for new yield vaults as everything is available using a single API.	Increases operability of the vaults; Potentially leads to more supply.
Ensures innovations and security are centralized	Makes contracts robust and battle-tested;

since they both fall under the same token standard.	Increases growth and adoption of yield-bearing instruments.
Does not require each vault to go through a separate audit.	Increased growth and adoption of yield-bearing instruments as composability solves fragmentation; Quicker, cheaper and safer development.
The ability for built-in auto-rebalancing, harvesting, and strategy rotation plus other gas-saving benefits.	Improved marginal yields and transactional efficiency.

Yield optimizers

A classic subcategory within automated strategy vaults are yield optimizers. The core philosophy behind yield optimizers is seeking the maximum possible return on investment for its users. One of the longest, most successful examples is *Yearn Finance,* where users can deploy assets and immediately earn the most optimal and safest yield on different assets. Yield optimizers rely on Keepers constantly checking certain yield-generating venues, like lending platforms, and comparing their interest rates.

Above all, yield optimizers are platforms that help manage capital from normal users and DAOs and are deployed into yield farming opportunities to maximize performance. The optimization is conducted by the deployment of data analysis, automation and optimization techniques. This makes them able to find high-yielding opportunities and automatically compound rewards at optimal intervals. After deposit, users and DAOs are not required to do anything else, making yield optimizers incredibly appealing as a set-and-forget solution.

The nature of standard optimizers strategies and vault architecture makes them a prime target for black hat attackers. Thus, a security-first approach is crucial for both the creation and investment of a sustainable yield optimization platform.

Particularly interesting in this subcategory is the use of the *Keep3r Network* or *Gelato*. Both projects organize the automation of smart contract processes in a decentralized manner by matching organizations and other entities with technical professionals known as *Keepers* or *executors*, or, even simply, bots. In return for a reward these bots automatically execute an autonomous task that is important for the protocol to keep running.

In the case of *Keep3r*, if a bonded Keeper is selected for a job, they will only receive *KP3R* rewards once the job has been done successfully. Keepers have complete autonomy over their DevOps, infrastructure, and regulations for completing the task. The whole process is all done on-chain, and the advantage of this is that everyone can confirm that a task has been done.

Another protocol, Gelato, works similarly: instead of bonded Keepers, executors with staked *GEL* on their platform are chosen for the task. Task issuers are able to put up a reward amount. As of writing, *Gelato* works only with whitelisted executors.

Yearn

Website URL:	https://yearn.finance/
Chain(s):	Ethereum, Fantom, Arbitrum.
Token:	*YFI*.
Fee structure:	20% on performance, 2% on management.
Automation process:	*Keep3r Network*

Previously known as *iEarn*, *Yearn* was solo-created by *Andre Cronje*. After suffering an exploit in February 2020, *Cronje* first stepped away and then returned into the project rebranding it into *Yearn*, introducing a suite of new products.

Yearn offers a wide variety of yield-optimizing vaults varying in interest rate, strategy, and token composition. In exchange for depositing, users would receive *yTokens* as a receipt for their deposits. However *Yearn* recently switched to the $ERC - 4626$ token standard for their newer vaults. The short-way to name a *Yearn Vault* is *yVault*. The protocol uses Keepers as maintainers of their vaults. To be more specific, they delegate the automation of vault tasks, like harvesting and rebalancing, to the *Keep3r Network*.

To put things into perspective, let's consider the different iterations of *Yearn*.

In *v1*, *Yearn* could only handle one strategy at a time. The standard used for the *yVault* tokens was the $ERC - 20$ standard, allowing for exchangeability and composability within the ecosystem. V1 also charged users a 0.5% withdrawal fee on top of a 20% performance fee. In its initial version, *Yearn* only earned yield by lending, trading and getting liquidity incentives.

6

On February 2021, the $yDAI$ Vault on $Yearn\ v1$ was hacked for a total of $11\ million$. The impact of the attack was limited by the quick response of the $Yearn$ security team and the multisig wallet signers that protected the remaining $24\ million$ of the vault. The attacker used the vault's automatic interactions with the $Curve\ 3pool$ to take advantage of favorable exchange rates.

$Yearn\ v2$ was introduced in mid-January 2021. It is able to run up to 20 strategies at the same time and includes a mechanism to funnel staking fees to the protocol's treasury, which are ultimately meant to make development sustainable in the long term. Furthermore, the protocol has replaced vault controllers with the strategist and guardian roles, automated the $harvest$ and $earn$ calls through the Keep3r bots network, and removed the withdrawal fee.
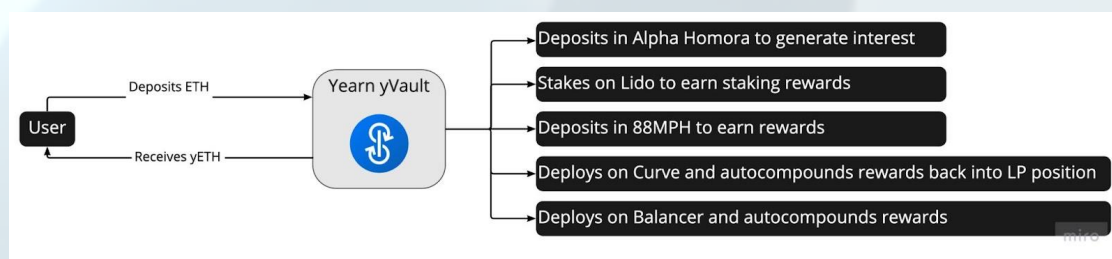
Yearn yVaults mechanics, 3six9 Cognitio

A vault that benefits the entire platform considerably, is the $yveCRV$ vault. $Curve's$ native token, CRV is used to vote on the reward distribution on Curve, and since Yearn relies heavily on a lot of the farms on $Curve$ it is in the protocol's best interest to direct rewards to those farms. Users are able to deposit and lock CRV in the $yveCRV$ vault and in return users receive a portion of the protocol fees, optimized $Curve$ rewards, while their locked CRV are made liquid in the form of $yveCRV$. Ten percent of the rewards are used to boost all $yVaults$ that rely on $Curve$ pools.

All the vaults available on $Yearn$ are created, curated, and maintained by community contributors and the core team. The level of decentralization by the protocol is impressive. Anyone can propose a strategy, and if successfully implemented, is able to earn from the performance fees charged. The vetting process is as follows:

a) A certain degree of due diligence is performed to gauge the viability of the strategy;
b) The strategy is put into code and gets ready for testing;
c) After a thorough peer review, the strategy is deployed to Mainnet for testing;
d) If successful, the core devs perform an audit;
e) The strategy is put into production. The creators of the strategy, the $Strategists$, receive a performance fee from the vault.

$Yearn$ also launched $yInsure$, a pooled insurance that provides coverage from financial loss underwritten by $Nexus\ Mutual$. The main advantage in comparison to $Nexus$ is the fact that in order to own the pool token, a user does not need to go through KYC, and on $Yearn,$ the token itself is

7

transferable following the $ERC-721$ standard. $Yearn$ has been audited by various parties, among others $Quantstamp$, $CerTik$, and $Mix\ Bytes$

Pros	Cons
Impressive level of decentralization with functioning and backtested governance mechanisms.	Performance fees are considerably high compared to the competition.
Incentives aligned for the strategists and the protocol getting performance fees on implemented strategies.	
Grants are awarded to various contributors in the $Yearn$ ecosystem and are evaluated on a month-to-month basis.	
Key to $Yearn$'s market share growth was their ability to attract sizable partner protocol deposits.	

BadgerDAO

<table>
<tr><td></td><td></td></tr>
<tr><td>**Website URL:**</td><td>https://badger.com/</td></tr>
<tr><td>**Chain(s):**</td><td>Ethereum, Fantom, Arbitrum, Polygon.</td></tr>
<tr><td>**Token(s):**</td><td>$BADGER, bBADGER, DIGG$.</td></tr>
<tr><td>**Fee structure:**</td><td>0.1% on withdrawal, 10/20% on performance.</td></tr>
<tr><td>**Automation process:**</td><td>*Chainlink* keepers and self-hosted keepers.</td></tr>
</table>

Badger was launched in the fall of 2020 and is a DAO focused on incentivizing and accelerating the use of *Bitcoin* within DeFi across multiple blockchains.

Since its launch *Badger* has proven to house a powerful, passionate, and smart community, and has built various products and services to empower yield-generating strategies on its platform. *Badger*'s core product are its so-called *SETTs* (*Sett Vaults*). These function similarly to *Yearn*'s vaults, but rely on interest-bearing receipt tokens instead of $ERC-4626$. Users deposit assets and receive representative *bAssets*. The yield flow is standard as the deposited assets are deployed in a strategy while the user earns optimized yield without having to manage everything themselves.

BadgerDAO has two native tokens:

    a) *DIGG*, a decentralized stablecoin pegged to *Bitcoin* with an elastic supply. The mechanics of this token allow its supply to adjust on a periodic basis without changing the portion of ownership. The rebase happens according to supply and demand as follows: in case DIGG is in high demand and trading for more than 1 BTC, the model expands and DIGG wallets balances increase to fulfill the supply. In case DIGG is trading below 1 BTC, the market signals that there is an oversupply of DIGG on the market and contracts the supply of DIGG in wallets. The main benefit to *DIGG* is that it is a non-custodial *Bitcoin* derivative whereas *wBTC* and

renBTC rely on a central entity for their issuance. *bDIGG*, on the other hand, is an interest-bearing token that does not rebase and can be used in different DeFi platforms.

b) *BADGER*, the governance and distribution token of the platform. The token can be used for voting on proposals and for distributing rewards to those who manage the SETTs.

Some of the vaults are boosted by native token rewards. The protocol has a set framework for vault discontinuation when this is in the DAO's best interests.

Badger hosts a *Bitcoin* bridge in collaboration with *Ren Protocol*. The bridge facilitates the transfer of *BTC* from the *Bitcoin network* to *Ethereum*, allowing users to receive *renBTC* and *WBTC*. Any wrapped *Bitcoin* tokens can be deposited in the *ibBTC* vault earning users yield on otherwise unproductive tokens. *BadgerDAO* manages deposited assets and provides a base interest rate that is derived from the return of underlying Sett Vaults. Furthermore, the receipt token, *ibBTC*, can be LP'd elsewhere improving the overall liquidity and utility of the token.

Interestingly, *Badger* has dedicated a lot of time and effort to creating and optimizing their yield by efficiently utilizing the reward boosts from *Aura* and *Convex*. Initially, *bveCVX* was designed as a means for *Badger* to increase yields across certain pools by utilizing the voting power of *bveCVX* depositors. Eventually due to the increase of competition in the *Curve Wars*, *Badger* was no longer in the position to offer sufficient rewards to *bveCVX* holders. The team began to iterate on how they could improve the rewards offered to ve-tokenholders and experimented with different ratios between the value in voting influence tokens and the total value of liquidity the protocol wants to support.

This led to the creation of *Badger*'s *graviAURA* product. A token structure that rewards LPs more efficiently and is built around vote-locked*AURA*. The token is native to a similarly-named platform which can best be conceived as the *Convex* for *Balancer*. It enhances the return on *Balancer* pools by creating a flywheel around it. *Badger* utilizes both *Balancer* and *Aura* for their yield farms, and their *graviAURA* token model makes it easier for *Badger* to build sustainable *Balancer* pools as the *graviAURA* votes for the underlying pool. *Badger* allows LPs of these pools to earn trading fees, *BAL* and *AURA* token incentives, bribes, and have all their rewards compounded automatically. This is very valuable for investors as it provides them with a means to aggregate and earn all the yields involved in the *Balancer* pools. The *graviAURA* token model can be perceived as a more advanced flywheel designed specifically for *Badger*.

Over the years *BadgerDAO* has profiled itself as a true DeFi powerhouse. Not only has it developed a full suite of products dedicated to enhancing the use and utility of *Bitcoin*-like assets in DeFi, but they also participated successfully in the *Curve Wars* and proven capable to enhance the flywheel effects of ve-tokens in the form of *bveCVX* and *graviAURA*.

The protocol's smart contracts have been audited multiple times by *Zokyo*, *Hacchi*, *Quantstamp*, and *Citadel*. Security is a top priority for the team as they follow a guarded launch approach in initial testing periods on new vaults capping liquidity at a low ceiling and running both a council of white hats and two bug bounties programs. Nonetheless, the platform was the victim of a $120 *million* hack in December 2021. The hack had nothing to do with smart contracts but was a result of a front-end exploit.

It was made possible by an unauthorized API key which allowed malicious snippet to set user Web3 permission to the attacker wallet

Pros	Cons
The platform has a diverse offering of 67 different vaults.	Some of the vaults are very similar to each other and this can result in the low conversion of investors.
Products are community-initiated and created.	Was victim of a $120 $million$ exploit in December 2021,
Demonstrated product-market fit with > $80 $million$ in TVL.	

mStable

mSTABLE	
Website URL:	https://mstable.org/
Chain(s):	Ethereum, Polygon.
Token:	*MTA*.
Fee structure:	10% on performance.
Automation process:	*Gelato* and *OpenZeppelin*'s Defender.

mStable is a yield protocol focused on pegged assets. The platform bundles same-based assets into $meta-assets$, resolving the issues of fragmentation and impermanent loss. The meta-assets are minted and redeemed on-chain via non-custodial contracts, making *mStable* a peer-to-pool protocol. Each meta-asset represents a share of liquidity in the underlying asset pool.

$Meta-assets$ are designed to produce competitive interest rates when deposited into the protocol's Save contracts, which can be considered vaults. The deposits into the vaults are then rooted to third-party protocols to earn yield on the deposited assets in the underlying pools.

Previously, *mStable* used representative, yield-bearing assets but switched over to standardizing their vaults with the $ERC-4626$ token standard. The team is working on adopting the $ERC-4626$ token standard to leverage the composability of *mStable Save* and is building new, generalized infrastructure

12

dubbed *Meta Vaults*. These vaults will be fully compatible with other $ERC-4626$ vaults and will combine, compose, and optimize multiple yield sources into a single vault.

There are three types of vaults on $mStable$:

a) **Basic vaults** - they are similar to yield-bearing asset vaults. They only accept a single currency that is deployed elsewhere and aggregate yield from underlying money legos;

b) **Multi Asset Vaults** - these types of vaults are not compliant with $ERC-4626$, but allow multiple tokens as deposits. They are built on top of other $ERC-4626$ vaults, thus there can also be multiple sources of yield;

c) **Meta Vaults** - these vaults are $ERC-4626$ compliant and composed of all types of vaults. The main selling point of these vaults stands in their ability to combine, compose, and optimize multiple yield sources into standardized vaults. They allow for full $ERC-4626$ support. Meta vaults supposedly transfer the complexity of investing, managing, and automating yield strategies from the users to Vault managers leaning into easy-to-use savings accounts.
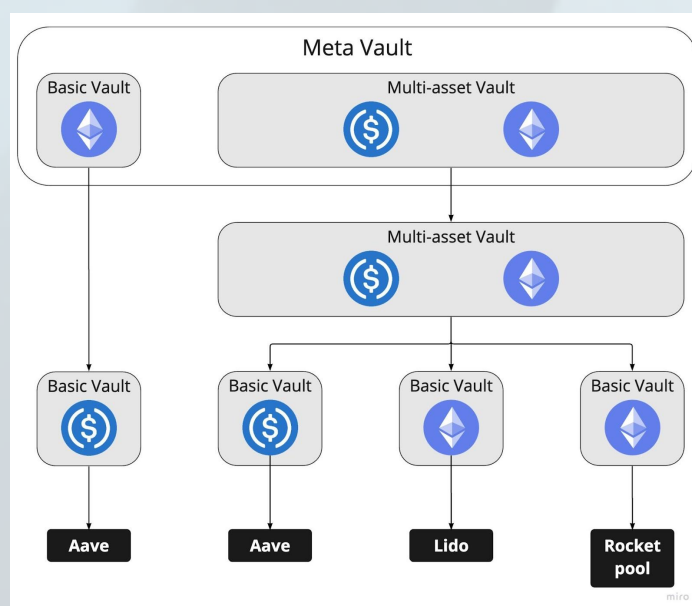
Meta Vault mechanics, 3six9 Cognitio

The two meta-assets introduced by the platform are $mUSD$ and $mBTC$.

- $mUSD$ is a dollar-pegged asset backed by a basket of selected stablecoins and includes $USDC$, $USDT$, DAI and $sUSD$.
- $mBTC$ is pegged to the price of BTC and is backed by a pool of tokenized BTC on the $Ethereum$ network, including $renBTC$, $WBTC$ and $sBTC$

These meta-assets have a vault called $mStable\ Save$. This vault generates revenue by depositing the underlying assets in Aave or Compound. Additional fees for $Save$ is generated from swaps in and out of $mUSD$.

The protocol also offers the $Feeder\ Pools$, which are liquidity pools composed of 50% $mUSD/mBTC$ and other assets that share the same peg. These pools allow for more efficiency and less slippage in trades and also reinforce the peg of pooled $mAssets$.

$mStable$'s native governance token is MTA. The token can be earned either by providing liquidity through staking or by saving $mUSD$ or $mBTC$ through the Save contracts. Token holders decide the addition/removal of $mAssets$ and $bAssets$, the redemption of minting fees and the selection of oracles.

The protocol is backed by $DeFiance\ Capital$, $3AC$, $Alameda\ Research$ and some DeFi influencers.

As of writing, there has just been a proposal on $mStable$'s governance forum to raise the fees from $10\%\ to\ 20\%$, incrementing the revenue generated by the protocol by 80%. $mStable$ has been audited by $Peckshield$, $CerTik$, and $ConsenSys\ Dilligence$.

Pros	Cons
The protocol provides stability for its own pegged assets via $Save$ contracts and $Feeder$ pools.	Saving yields $(0.5-2\%)$ are low contrasted to those of their competition.
The team is actively socially financing scholarships	The governance token is issued to boost pools

|	
for hackers at *ETH* events.	TVL through liquidity mining. The rewards dumping effect of LPs is tackled allowing them only to claim 33% of their rewards immediately. The remaining rewards are streamed linearly after 26 weeks.
The team has come up with a new vault structure based on the $ERC-4626$ format that proposes to be more efficient and composable.	*mUSD* experienced several slight depegs before stabilizing in December 2021 and its main use case is yield vampiring on the platform. We can draw this conclusion by taking a look at how the stablecoin market cap has fallen back since the platform has decreased the yield on the stablecoin pairs.
	mUSD has no integration into other major protocols.
	Unclarity on how the *Save* yields are generated.

Subcategory: auto-compounders

Beefy Finance

Website URL:	https://beefy.finance/
Chain(s):	15 + chains.
Token:	*BIFI*.
Fee structure:	9.5% on performance.
Automation process:	*Gelato* and self-hosted keepers.

Beefy is centered around auto-compounding and optimizing the yield on single asset and LP farms. The protocol is live on almost every major chain and makes safety a priority.

The main product offered by *Beefy* are their vaults, in which users are able to stake crypto tokens. The strategies tied to the Vault will automatically increase the deposited token amount by compounding arbitrary yield farm rewards back into the initially deposited asset. Funds are not locked in the *Vaults* so users are able to withdraw at any moment but the effects of compounding only kick in the medium to long term for investors.
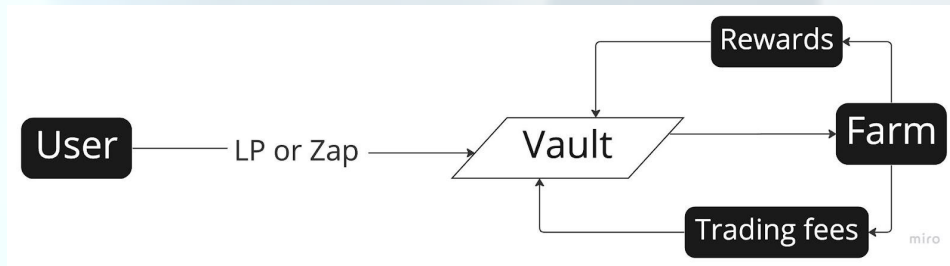
Beefy LP Vault mechanics, 3six9 Cognitio

As a receipt of deposits $mooTokens$ are distributed to investors. These are interest-bearing tokenized proof of deposit vault-unique. This type of token is both privacy-preserving and acts as an interest-bearing collateral.

The vault is subdivided into two big categories. Money Market vaults, which utilize lending platforms to generate the highest possible yield, and native token farming, which take advantage of popular farms and compound profits of the native reward token

Every platform integrated into the $Vaults$ ecosystem has gone through $Beefy\ SAFU\ Practices$. This is the set of rules established to verify that:

a) Protocol's contracts have been verified in the block explorer;
b) Non-native tokens must be issued by reputable bridges;
c) Enough liquidity is present for swapping farm token rewards;
d) Rug and migrator functions are removed or at least time locked;
e) Farm token emission rates are time locked;
f) Farm token holders with a significant portion of the circulating supply are not EOAs or multisigs;
g) All proxy implementation changes are time locked.

Every member of the community can propose a vault and $Beefy$ strategist, following manual testing on every $Vault$, and before it goes live. Each vault is hard code and the code is immutable. This means that once released, no one can modify vaults and strategies.

The platform has also integrated a $Beefy\ Safety\ Score$, a tool aimed at educating users when making a decision to enter a particular $Beefy\ Vault$.

$Beefy$ also offers $Beefy\ Boost$. This is basically a promotional layer of yield on top of the already existing vaults and is done in partnership with other protocols to promote the use of their token and pool liquidity.

$BIFI$ is the multi-utility token of the platform. Token stakers earn part of the revenue generated while both holders and stakers are entitled to vote in governance. The revenue-sharing mechanics entail users to stake the token in the $BIFI$ maxi vault, the $BIFI$ earnings pools or any $BIFI$ liquidity vault. The total supply of $BIFI$ is limited to $80k$ tokens and these are all in circulation since July 2022. As no additional tokens are minted, staking rewards are provided to stakers by buying $BIFI$ from the market.

Beefy has also deployed a series of *Beefy Escrowed* assets that give stakers access to maximize Validator Node rewards without locking the token. These tokens are backed at parity with native assets. The offering includes as of writing *beFTM, binSPIRIT, beQI* and *beJOE*.

Beefy is audited by *DeFiYield* and *Certik*. Major findings resolved.

Pros	Cons
Massive offering with complementary tools such as the safety score.	Business model continually dumps token rewards to auto-compound.
Can be used as a launch pool platform by upcoming protocols looking to attain liquidity.	
Attractive to DeFi beginners for their *SAFU best practices* and listed protocol safety scores.	

Sommelier

Website URL:	https://www.sommelier.finance/
Chain:	Cosmos SDK, Ethereum.
Token:	*SOMM*.
Fee structure:	0.25% on asset managed, performance split. Third-party fees on top.
Automation process:	By the *Sommelier* validator set.

Sommelier is built on the *Cosmos SDK* and a bi-directional Ethereum bridge. They use a *Cosmos* chain as a co-processor to *Ethereum* to process as many calculations as possible off *Ethereum*. In order to perform off-chain strategy calculations, *Somm* aggregates and batches transactions, employing off-chain data modeling techniques traditionally used in TradFi.

Their vaults are called *Cellars* and execute trades among selected sets of tokens based on instructions dictated by off-chain computation.

The only *Cellar* available as of writing is *Aave*'s, sitting at $1 \, million$ TVL and a 18.6% expected APY, with the majority of the rewards being liquidity mining incentives ($\sim 17\%$). The goal of the strategy is to select the optimal stablecoin lending position available to lend across *Aave* markets on a continuous basis, basically outperforming a static strategy of lending any single stablecoin. Besides interest rates, the strategy also observes other key variables such as rate volatility, gas fees, slippage estimations, and TVL.

The team is working on the integration of $Convex$, $Futuristic$ and $BTC - ETH \, Cellars$.

Strategy providers are third-party institutions such as VCs or community members. Before submitting the strategy to their end-users, *Sommelier* backtests each strategy's performance. The team has the

19

goal for the strategy backtests to be automated, and therefore is working on developing generic flexible tools to support backtesting and the deployment of new *Cellar* strategies.

The platform also deposits its own funds into the *Cellars* to prove the validity of each strategy and to kickstart the liquidity. To change anything about a strategy, strategy providers must submit a governance proposal resulting in an extra layer of safety for depositors.

The project is backed by *Polychain Capital*, *Alameda Research*, *Standard Crypto*, *Cygnilabs*, and *Multicoin Capital*. *Sommelier's Cellars* have been audited by *Macro Audits*.

Pros	Cons
Aims to provide cross-chain vaults to DeFi users.	Strongly relying on VC funding.
Built on *Cosmos SDK* which opens up a lot of cross-chain opportunities.	The only vault live right now is the *Aave* vault which is heavily incentivized. The platform's TVL sits at $1 *million* as of writing.

SPool

Website URL:	https://spool.fi/
Chain:	Ethereum.
Token:	*SPOOL*.
Fee structure:	11% on performance plus additional creator fee.
Automation process:	Self-hosted keepers.

Spool brands itself as an open DeFi toolbox that allows users to create and host yield-generating strategies.

The strategies are *Smart Vaults, Spools*, with a predetermined performance fee. Each vault contains a chosen risk model and a set of yield platforms to route to in order to generate yield in a risk-managed and optimized manner. These vaults are automated within the constraints of the predetermined parameters of risk appetite and "non-custodial". Once the strategy has been initiated, the vault automatically compounds the rewards to maximize yield. As of writing, there are a few stablecoin vaults live that are heavily incentivized by native tokens.

The community is incentivized to propose strategies, as the vault creators can set a performance fee capped at 20% of generated yield. Every strategy consists of two different parts. The first part is the strategy adapter, which manages deposits, withdrawals and compoundings for a given yield generator. The second part is the actual contract that interacts with third-party platforms.

Spool utilizes a $Buffer\ System$ that aggregates deposits and withdrawals. The system mechanics allow every user to deposit and withdraw to and from the buffer. If deposits and withdrawals happen at the same time, they are matched with each other. This approach makes transactions significantly cheaper for users. *Spool* also uses vault inheritance, which means any new vault routes capital to a master

21

contract and when it can be applied then transactions are batched with other vault transactions to optimize costs. Individual vaults still track exposure separately.

The protocol markets itself as a DeFi middleware, as they allow institutions to offer their clients DeFi seamlessly thanks to their *Spool SDK* and their branded white-label *Spool*. White-labeling is widely used in TradFi, with big names such as *UBS, JP Morgan and Goldman Sachs* offering this kind of solution to other banks and financial advisors.
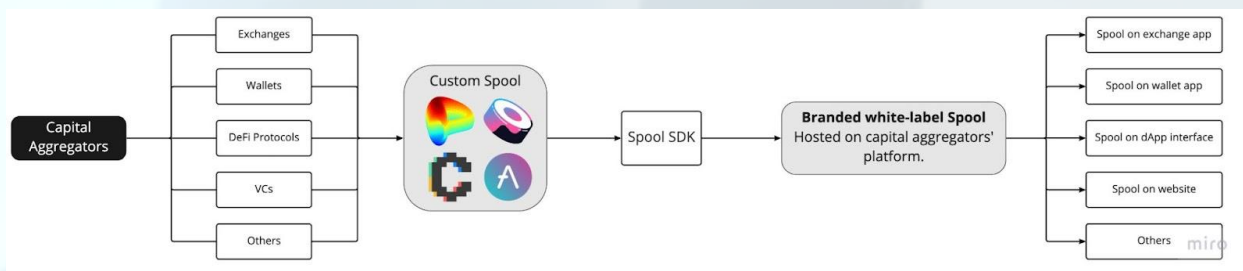
SPOOL is the governance token of the platform. Stakers can partake in governance through *voSPOOL* and accrue the entirety of the protocol generated revenue, which is streamed directly to stakers for 80% and to the DAO controlled treasury for 20%.

The project is running a $1.5 *million* bug bounty via *Immunefi* bug bounty and has been audited by *Peckchield, Zokyo and QuantStamp*. All findings were resolved.

Pros	Cons
The community vault creation can be beneficial to both their offering and community members.	The project has shown some market traction with $30 *million* in TVL but it's netting only outflows since September 2022.
Vast offering for a range of risk appetites.	

Key Takeaways

Management

A Yield Optimizer's business model is straightforward: aggregating yield from the highest-yielding venues to offer users better yields, aggregate liquidity and take performance fees on the enhanced yield. There is no complex strategy involved apart from the logic that checks the yields and automatically deposits, withdraws, and harvests. As per the blockchain works, smart contracts themselves cannot be automated. Therefore, the space relies on keepers and bots; *Externally Owned Accounts* that are incentivized to trigger smart contract function calls based on predefined conditions.

Keepers can be organized in different ways. The protocol can host their own Keeper bot that runs on a server or delegate the job to decentralized bot networks, like *Keep3r* network, or the less decentralized, *Gelato Network*. Other solutions are *Chainlink* and *OpenZeppelin*'s Defender. Setups relying upon *AWS* or other centralized servers introduce a critical centralization risk. For example, *Gemini*, one of the largest exchange players using *AWS*, was recently affected by an outage to one of their primary databases, knocking out the exchange and causing an untold loss for them and traders relying on the platform. This level of centralization can also be exploited by malicious and rogue team members, or worse - by black-hat hackers. In case of Keepers, the risk is not as significant as with private key storing or admin functions in smart contracts. Nonetheless, ideally, the keepers too would be hosted in a decentralized manner.

Alternatively, *Sommelier* is its own application-blockchain, and it can rely on their validator set to do the automation for them. This results in validators that are not only responsible for the processing of transactions, but serve a second purpose as operators of the protocol's business operations, instead of relying on Keepers or bots.

Flywheel effects

Often, specific platforms like *Curve* and *Balancer* are core building blocks of yield optimizers; we will refer to them as core yield venues. These venues have clever tokenomics in place that favor long-term alignment with the protocol by vote-locking governance tokens to increase generated yield on their platforms. A notable consequence was the *Curve Wars* which took place last year, during which several protocols were trying their best to accrue as much *Curve* and *Convex* governance tokens as possible. Among the participants were *Yearn* and *Badger*.

What yield optimizers have learned is that it is very lucrative to create their own derivative product on top of governance aggregators to incentivize and attract holders to their platforms. Examples are *yCRV*, *bveCVX*, *graviAURA*, and *Beefy Escrowed* assets. It is in the optimizers' best interest to accrue as much of these governance votes to increase the potential yields on their farms. Therefore a successful flywheel product running on top of their own platforms is of their highest interest.

23

For example, it seems that the market has deemed $mStable's\ mUSD$ model insufficient for creating value, whereas escrowing token models, like the ones from $Beefy$ are in higher demand. The key difference is that $Beefy$ builds upon specific, yield-generating assets, whereas $mStable$ aggregates "plain" stablecoins that generate yield only in lending pools. Additionally, it can be concluded that protocols building on top of governance aggregators, like Convex and Aura, can immensely benefit from the voting power that these protocols provide to boost the rewards of their pools.

Security

We find that for optimizers security must be a priority and audits are a strict requirement. Building on top of existing smart contracts adds additional points of failure in case of malfunctions and exploits. The mentioned projects do keep this in mind and implement security as a top priority. $Beefy$ displays a safety score for every implementation they offer and this helps users have a broader idea of the products they are using.

Conclusion

We hope that this paper served as a short overview of successful DeFi yield optimizers. The intent is that it can be an insight into what has worked and as an inspiration for future yield-enhancing products.

Optimizers are a great tool for investors and traders that do not want to actively monitor and manage positions. Protocols like $Yearn$ and $Badger$ have specialized in creating infrastructure to automatically generate the highest possible yield across safe and reliable yield venues. They often build on top of other yield generating positions from $Aave, Curve, or\ Balancer$. These platforms have been around for years and have proven to be reliable, secure and profitable and therefore have created a trustworthy piece of infrastructure. By utilizing these platforms, optimizers have become decentralized, automated money-managers which are accessible to anyone regardless of jurisdiction or portfolio size. This is in line with the ethos of crypto and DeFi, but also a demonstration of the power of smart contracts.

Still, this space is still evolving. Every now and then, more DeFi products are invented. Recent examples include the success of many perpetual DEXs, and these open up the opportunity for new

strategy vaults. In that regard, DeFi is an extremely exciting space to be involved in and an end-game is still far out. There are so many moving parts that can be improved on from optimized automation and enhanced profitability, to safer oracle systems and the inclusion of more advanced strategies.